Early Detection of Driving Maneuvers for Proactive Congestion Prevention

Debasree Das[1], Shameek Bhattacharjee[2], Sandip Chakraborty[1], Bivas Mitra[1], Sajal K. Das[3]
[1]Indian Institute of Technology Kharagpur, Kharagpur 721302, INDIA
[2]Western Michigan University, MI 49008, USA
[3]Missouri University of Science and Technology, MO 65409, USA
Email: debasreedas1994@gmail.com, shameek.bhattacharjee@wmich.edu, sandipc@cse.iitkgp.ac.in, bivas@cse.iitkgp.ac.in, sdas@mst.edu

***Abstract*—Road traffic congestion affects not only the commute delay but also a city's overall social, economic, and environmental growth. Existing approaches for road congestion mitigation primarily adopt a reactive approach by detecting congestion after it occurs and recommending alternate routes to the vehicles, which fails to prevent congestion cascading. In contrast, we propose a pervasive platform called *ProCon* that proactively infers the driving micro-behaviors that can contribute to congestion formation and assist the drivers in avoiding such maneuvers in real time during the navigation. Thorough evaluations over multiple real-life and simulated datasets indicate that *ProCon* can reduce congestion for more than 60% of the scenarios on average while significantly reducing the travel time of the vehicles.**

***Index Terms*—Road congestion, Proactive mitigation, Driving behavior, Pervasive recommendation**

I. Introduction

Road congestion impacts navigation as well as the overall socio-economic-environmental growth of a region. According to the congestion reports published by TomTom[1], the top-3 traffic-congested megacities (Lima in Peru, Bengaluru and Mumbai in India) spent an additional 24 minutes on average in traveling 10 km in 2022. Existing approaches of road navigation primarily use a *reactive* approach, i.e., predict congestion based on the vehicle density and associated events, like the average speed of the cars, nature of the road, demographic factors, etc. Although such approaches may work intermittently and help for long-term policy planning with reduced recurring traffic congestion, they do not explicitly consider congestion prevention as an objective during pervasive navigation.

One weakness of reactive congestion measures is that they overlook the ripple effects of individual driving actions that impact the traffic even at a city scale. While traffic congestion can be recurring and local (like congestion near a shopping/event complex daily during busy hours), it triggers a cascading traffic pattern that influences several non-recurring congestion hotspots [1] across the city depending on the *perceptual bias* [2] of the driver (habitual perception based on standard driving practices) towards the regional driving micro-cultures. For example, a driver might look for an alternate exit while observing a lane block ahead. However, suppose multiple drivers follow the same alternate exit based on

[1]https://www.tomtom.com/traffic-index/ranking/?population=MEGA(Accessed: May 9, 2024)

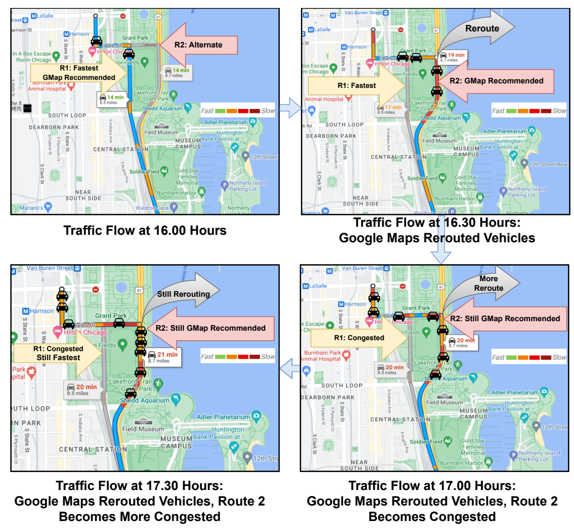


Fig. 1: Congestion cascading due to map-navigated rerouting (Data from Chicago, USA on September 6, 2023)

their local perception. In that case, the alternate route might get congested, and the impact of congestion thus gradually propagates to other parts of the city (see Fig. 1). Coupled with such perceptual bias, several anomalous driving behaviors originating from specific driving micro-cultures, such as side-slipping to avoid snows or potholes on the road, frequent overtaking on the lanes, etc., influence the traffic cascading pattern that leads to eventual road congestion [3]. Considering the situation discussed above, this paper tackles the following research problem. *Considering the runtime driving micro-behavior, can a navigation app recommend maneuvers that proactively prevent traffic congestion cascading while not significantly impacting travel time or distance?*

Challenges. Although existing works [4]–[7] have developed models to correlate historical data and run-time information to predict future traffic density, there are challenges in proactively adjusting real-time navigation to prevent congestion.

1 *Traffic density alone can not predict congestion well ahead before it starts occurring.* Traffic density enables us to understand congestion only after it starts to form and does not allow preventive measures. In contrast, tracking changes in the driving micro-behavior along with context information

enables an a priori understanding of the traffic cascading that may lead to eventual congestion.

2 *Traffic congestion does not linearly depend on the driving micro-behavior.* Anomalous maneuvers by a few vehicles do not always lead to traffic congestion. Instead, there is a complex interplay among the maneuvers taken by different vehicles *coupled with* the road condition, lane type, and latent factors to determine whether the scenario may lead to traffic congestion soon. Further, using such information in real-time to predict congestion before it actually happens is a challenge.

3 *Proactive congestion mitigation demands highly dynamic route planning.* Unlike reactive approaches where the vehicles are rerouted after observing traffic congestion on one path, navigation planning to proactively mitigate the congestion cascading needs frequent runtime analysis of the contextual information that depends on several highly dynamic latent factors (e.g., weather, road, lighting conditions, etc.).

Our contributions. To address the above challenges, this paper proposes *ProCon*, a multimodal domain adaptive model that pervasively learns the contextual impact of different driving maneuvers on the possibility of local congestion formation and proactively alerts the driver to avoid the maneuvers having a high probability of forming congestion. *ProCon* runs locally on a smartphone as a standalone wrapper on top of the navigation app and uses the IMU and GPS data coupled with the associated map information. In cotrast to the existing works, our contributions in this paper are as follows.

1 *Characterize driving behavior leading to congestion.* We conduct a small-scale study using digital map data (e.g., Google Maps, TomTom) and naturalistic driving trajectory dataset inD [8] and the largest driving dataset Berkeley Deep Drive (BDD) [9] to understand the aspects of driving behavior and spatio-temporal road properties that lead to varying levels of congestion (e.g., low, moderate, high).

2 *Self-Explainable on-device model for proactive congestion prediction.* The crux of *ProCon* leverages (a) vehicle trajectory and interactions with peer vehicles and (b) spatio-temporal road properties to detect its correlation with driving behavior. Following this, we design a model that understands the temporal dependency among different types of driving behavior with dynamic road properties. The model-identified explanations responsible for congestion help us predict congestion ahead of time and accordingly recommend driving micro-behaviors that help in macro-scale traffic management (like dynamic rerouting to prevent congestion cascading).

3 *Evaluation with real and simulated datasets.* We use two naturalistic driving datasets [8], [10] along with runtime simulation through Anylogic 8.8.4 simulator [11] to evaluate *ProCon* comprising more than 10 various road intersections with 70 driving hours. Thorough experiments show that *ProCon* can prevent 50–70% of the cascading congestion scenarios when the congestion possibility is predicted 30 minutes ahead of time. Further, it results in a 10–40% reduction in the travel time with around 0–6% increase in the travel distance.

II. A Small-Scale Study with Real Datasets

Existing studies have shown that recurring congestion is predictable based on the demographic and spatio-temporal geo-patterns of the traffic movements [12]; however, it influences several non-recurring congestion-cascades across the city depending on the run-time traffic behavior and the local driving micro-cultures [13]. To understand whether such congestion cascading can be prevented by intelligently controlling the driving behavior during run-time, we conduct a small-scale pilot study to answer the following research questions. a *Does a particular vehicle maneuver always trigger the congestion cascading or do different maneuvers trigger congestion cascading at different times and places?* b *Does rerouting with "Lane Change" maneuver trigger congestion cascading in neighborhood roads?* c *Do all the vehicles exhibit a similar behavior during congestion cascading?*

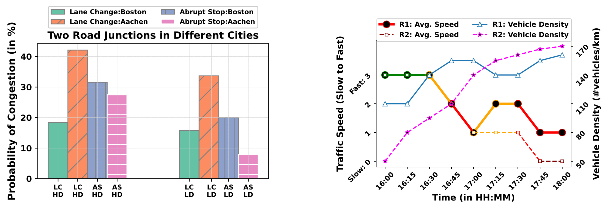
We conduct the study using TomTom and Google Maps data, and two publicly available driving datasets, namely Berkeley Deep Drive (BDD) [9] and inD [8]. The BDD is the largest driving dataset collected by 10k drivers voluntarily in the USA and Israel, covering 18 cities. The multimodal data covers IMU, GPS and video recording as the driver view. The German dataset inD covers 4 road junctions with IMU and GPS data from each vehicle along with the drone-recorded videos. We utilize two cities from the two datasets, Aachen, Germany & Boston, USA, for the following study.

A. Impact of Vehicle Maneuvers on Congestion Formation

We start by analyzing how different vehicle maneuvers influence congestion formation over two cities with different driving micro-cultures – Aachen, Germany, and Boston, USA. We focus on two different maneuvers – "*Lane Change*" (LC) and "*Abrupt Stop*" (AS), extracted using existing studies [14], under two different scenarios – *low vehicle density* (LD, $<$ 80 vehicles/km) and *high vehicle density* (HD, $>$ 200 vehicles/km). Notably, Aachen observes more frequent lane changes, while Boston observes frequent abrupt stops and acceleration/deceleration on the road. Fig. 2(a) indicates that Aachen observes a higher probability of congestion during high and low vehicle densities in the presence of frequent lane changes. On the contrary, we observe that abrupt stops influence congestion in Boston with a higher probability for both low as well as high vehicle densities. Such observations indicate that different vehicle maneuvers influence congestion differently, which depends on the driving micro-cultures of a locality (like lane changes are more common in Aachen, while abrupt stops are common in Boston) and their spatio-temporal impacts on the overall traffic movements.

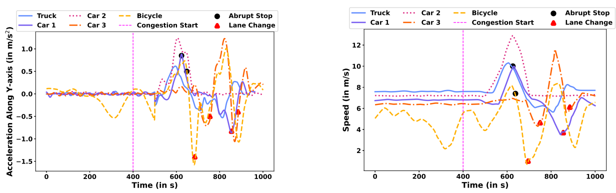
B. Congestion Propagation due to Frequent Rerouting

To observe how frequent rerouting coupled with the *Lane Change* maneuver influences congestion cascading, we consider the scenario shown in Fig. 1. As we have seen earlier, the congestion cascades from R1 to R2 and then persists in R2 even when it gets mitigated from R1 (after 17:00 hours). To explore this further, we observe that while R1 is a straight



(a) (b)

Fig. 2: (a) Probability of congestion under different maneuvers, (b) Congestion propagation due to frequent rerouting.



(a) Acceleration pattern (b) Speed pattern

Fig. 3: Impact of congestion on driving behavior.

drive, R2 has multiple lanes. Fig. 2(b) depicts the two routes' congestion information and the vehicle density on a temporal scale with their traffic speed as reported by the Google Maps API. The figure indicates that the vehicle density increased non-linearly in R2. As R2 already has more provision for changing lanes, vehicles start to take alternate lanes to avoid traffic. However, the lane change behavior further influenced the average speed of the cars on R2. The figure indicates that although R2 is a multi-lane highway and the vehicle density on that route was close to R1 at around 17:45 hours, it observed more congestion and lower vehicle speed than R1. The figure indicates that to mitigate congestion in the current route (R1), diverting traffic to other routes (R2) uncontrollably may propagate congestion more rapidly to the neighborhood roads. This also establishes a concrete base that further performing some maneuvers (say, lane change) to avoid traffic can worsen the congestion over time.

C. Change in Driving Behavior During Congestion

Here, we examine a non-congestion-to-congestion scenario using the data from Frankenberg, Germany, (on August 20, 2018, weekday) during the early evening hours. To check whether all types of vehicles behave similarly in terms of taken maneuvers, we pick 5 vehicles (3 cars, 1 truck, and 1 bicycle), which were peer vehicles in one directional traffic, and their trajectories are plotted in Fig. 3(a,b) using acceleration along Y-axis and speed values. We consider two maneuvers as used earlier – *abrupt stop* (black circle) and *lane change* (red triangle). Initially, the road had no congestion, and all the vehicles were moving with almost constant speeds (close to zero acceleration) as depicted in Fig. 3(b). However, at around 400 seconds, we observe some mild congestion on the road, and the vehicles start maneuvering abruptly. From Fig. 3(a), we observe that while car 2 decelerated smoothly, car 1 stopped abruptly (at 650 seconds) and further changed the lane (at 870 seconds). Notably, car 3 also changed the lane almost simultaneously (at 880 seconds), possibly replicating the behavior of car 1. However, the truck moved slowly and did not show any abrupt behavior. At the same time, the bicycle tried to avoid traffic by frequent lane changes followed by an abrupt stop with sharp deceleration. We observe a gradually increased congestion on that road between 600 seconds to 900 seconds, primarily due to the abrupt maneuvers taken by some vehicles, which got further contaminated to the following vehicles. Therefore, such diversity in vehicle behavior may trigger abrupt maneuvers, worsening the congestion scenario. These observations indicate that it is crucial to proactively inform or alert the drivers not to take abrupt maneuvers that can lead to traffic congestion.

III. Problem Definition and System Overview

Consider a set of vehicles $\mathcal{A}$ traveling on a road segment e having a set of spatio-temporal road properties $w \in \mathcal{W}$. The vehicles perform a set of maneuvers $\mathcal{M}_{t'}$ at time window t'. Further, we consider that maneuvers of one vehicle $a \in \mathcal{A}$ may impact maneuvers of its peer vehicle $a' \in \mathcal{A}$, denoted as *impact propagation* ($\mathcal{I}_{a,a'}$). The objective of this paper is to measure the influence of $\mathcal{I}_{a,a'}$ for each pair of peer vehicles $\{a, a'\} \in \mathcal{A}$ on congestion ($\mathcal{C}_t$) formation at a later time window t ($t' \leq t$). We consider four types of congestion scenarios during time-window t, $\mathcal{C}_t \in [0, 3]$ signifying {no, low, moderate, high} congestion, respectively. We aim to solve the congestion prediction $\mathcal{C}_t$ as a function $\mathbb{F} : \mathcal{R} \rightarrow (\mathcal{M}_{t'}, w(e), \mathcal{I}_{a,a'})$, where $\mathcal{R}$ represents the derived context in the form of a sequence of maneuvers $\{m_1, m_2, \cdots, m_n\} \mid m_i \in \mathcal{M}$ and road properties $\{w_1, w_2, \cdots, w_n\} \mid w_i \in \mathcal{W}$.

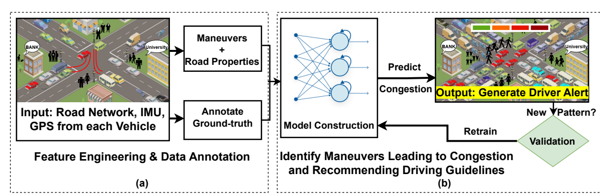


Fig. 4: *ProCon* system architecture.

A. System Overview

We propose a system architecture comprising two modules: *Feature Engineering and Data Annotation* (see Fig. 4(a)) and *Identify Maneuvers Leading Congestion* (see Fig. 4(b)).

1) Feature Engineering: ProCon relies on each vehicle's timestamped IMU and GPS data to infer per-vehicle maneuvers and map-extracted data to infer the road properties. We use existing literature [15] to extract 4 different maneuvers: "*Stop*", "*Turn*", "*Lane Change*", and "*Jerkiness*", from each vehicle using IMU/GPS data. We further extract vehicle dynamics, such as "*Relative Distance*" and "*Relative Speed*" between two consecutive vehicles using their position (GPS) information, which helps us to encode impact propagation $\mathcal{I}_{a,a'}$ among the nearby vehicles during model development.

To extract road features w, we design a map-extracting framework to infer 9 spatio-temporal features using Google Maps, TomTom, and OpenStreet Maps API, utilizing GPS data. The features such as “Road Segment Distance”, “Traffic Density”, “Traffic Speed”, “Speed Limit”, “Presence of Multimodal Transportation”, “Weather Type”, “Type of Road”, “#Point of Interest (Police Station, Hospital, Bank, ATM, Supermarket, Post Office, Institute)”, “#Connected Neighborhood Road Segment”, are extracted in a recurring manner for consecutive time-stamps (t) where t is a 20-second time window. Note that, whenever a vehicle performs a maneuver, its probability is calculated based on previously performed maneuver. Basically, we utilize the Markov property to compute the maneuver chain $\mathcal{N}$ denoting transitions of maneuvers performed by a set of vehicles. Next, we utilize these road properties to comprise a $[1 \times 9]$-dimensional feature vector w to fuse with the computed maneuver chain $\mathcal{N}$. Some of the road properties (like “*#Point of Interest*” & “*#Connected Neighborhood Road Segments*”) are encoded according to their actual values, whereas the rest are encoded as a discrete value $\in [0, 3]$ where 0 represents normal situations and a higher value represents poor situations.

2) Annotating Ground-Truth: We use the video data captured using a camera-equipped drone to annotate the ground-truth congestion level and vehicle maneuvers. Notably, this annotated data is required only to develop the model and not required during the run-time. We annotate the congestion level as a human-annotated label, $\mathcal{C}_t^g$, with the help of 3 volunteers by showing the video recordings. Each recording is annotated for 60 seconds window, and a label $\in [0, 3]$ is assigned denoting {“*No*”, “*Low*”, “*Moderate*”, “*High*”} congestion, respectively. As the congestion level observed over the video may vary with human perception, we fetch the traffic speed for the corresponding trips as {“*Fast*”, “*Moderate*”, “*Slow*”, “*Stop and Go*”} mapping to $[0, 3]$ from HERE Maps[2] to cross-verify the sanity of the human-annotated ground-truth.

Now, to distinguish the events that occurred ahead of different congestion levels, we instruct the volunteers to observe closely and figure out the maneuvers (say, over-speeding, turns, etc.) along with road demography and note them at every 60 seconds time window. We do not restrict them by providing any a priori information about maneuvers or road properties set to avoid bias and rely only on visual observation. These ground-truth observations are kept in $\mathcal{L}^g$, maintaining their sequencing order for model construction and validation.

3) Identifying Maneuvers Leading to Congestion: We divide each trip into continuous non-overlapping time windows of size δ where a sequence of driving behavior with its congestion label helps to construct a model with input features as a time-series data, as shown in Fig. 4(b). Following this, *ProCon* proceeds to analyze the influence of certain maneuvers that have a higher probability of congestion formation (higher $\mathcal{C}_t^p$ levels) using a self-explanatory method, called the *historical model*. During run-time execution, *ProCon* intelligently fuse information independently obtained from each vehicle (say, ego vehicle) along with the live available information on a standard map-based navigation service to predict the possibility of congestion if the ego vehicle takes a specific maneuver, by harvesting the historical model. We also incorporate model retraining after carefully validating the unforeseen data for adapting the model for changing traffic scenarios.

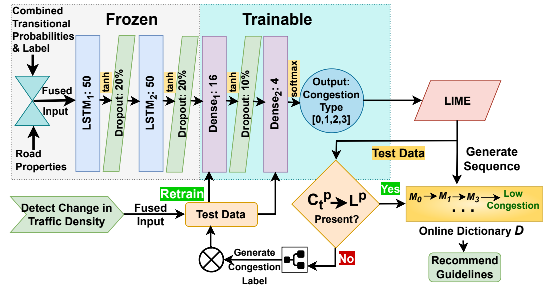


Fig. 5: Working principle of *ProCon*.

IV. Model Development

ProCon leverages extracted features, viz., maneuver chain $\mathcal{N}$ and road features $w \in \mathcal{W}$. However, as these features are in time-series format with variable lengths, capturing their temporal dependencies in the feature space is essential. Thus, we need a sequential model like a recurrent neural network (RNN) to capture the sequence of maneuvers leading to different congestion levels. We use an LSTM model with retraining to overcome performance drops on non-IID data. Additionally, we make the model self-explanatory to overcome the non-interpretability of black-box models.

A. Training Phase

As the maneuvers involved in $\mathcal{N}$ can either be “*safe*” or “*risky*” (denoted as anomaly x), depending on how well it is performed, it can contribute differently to congestion. Additionally, confidence in computing such an anomaly of the maneuver will help us to build a robust model. As maneuvers are the core component in constructing our model, leveraging their computation quality (i.e., *confidence*) is necessary to avoid false-positive derivation. Thus, we represent anomaly type as a discrete variable $x \in \{-1, 1\}$ of maneuver m and confidence of anomaly as a continuous variable $y \in \{0 - 1\}$. Next, we adapt Dempster-Shafer (D-S) [16] theory to associate both the confidence y and probability of performing m-th maneuver to compute subjective probability. Following this, the subjective probability is multiplied with x to associate level of anomaly in the maneuver chain $\mathcal{N}$. Then, we interleave derived maneuver chain $\mathcal{N}$ with its corresponding road feature vector w denoting road properties at each time-stamp t. This interleaving process transforms a single scalar value from one time-stamp into a $l_1 \times (l_2 + 1)$-dimensional vector; the first dimension l_1 represents the length of the Markov chain while the l_2 dimension represents the associated road properties, and the extra $+1$ dimension added represents the time-stamp. We make l_1 as variable length as within a time window number of transitions between maneuvers can not be of fixed length.

1) Architecture considerations: Consequently, the resulting input to our LSTM model for each sequence is a matrix of size $< (l_1 \times (l_2 + 1)), (1 \times 4) >$, where 1×4 is the one-hot

[2]https://www.here.com/ (Accessed: May 9, 2024)

vector of the ground-truth congestion level $\mathcal{C}_t^g$ as shown in Fig. 5. Our model comprises one LSTM layer of 50 units as the input layer, followed by two hidden layers as one LSTM (50 units) and Dense layers (width as 16), as shown in Fig. 5.

Between the input and the two hidden layers, three dropout layers of 20% (first two) and 10% are added, respectively, to prevent over-fitting, as it will exclude a few random inputs before feeding it to the next layer. The output from the last dropout layer is finally fed to a dense layer of the same dimension, i.e., a width of 4 as of the number of output labels $[0-3]$ with the softmax activation function. Softmax generates a label-wise probability vector as a multinomial one, and the label with the highest probability is chosen as the predicted label $\mathcal{C}_t^p$. We use "*tanh*" activation function for the other layers, as it works well with continuous input range values.

2) Loss function and optimization: As an optimizer, we use *Adam* as it converges rapidly and faster to compute for large datasets. For minimizing the loss between $\mathcal{C}_t^g$ & $\mathcal{C}_t^p$, we use *Categorical Cross Entropy* (CCE) as it works best with multinomial representation of labels.

3) Handling class imbalance: Our training data has a class imbalance known to drop the model performance. Typical label balancing techniques like SMOTE oversample training instances blindly without concisely selecting neighborhood examples, which can also introduce noise, hence unsuitable for our approach. As a solution, we use the "*Hard Negative Mining*" technique to boost the performance of our model. We refer to the training instances with low frequencies for specific labels and more entropy in the feature space as "*Hard to Classify*" examples. We carefully choose them during model training by inspecting the loss function as well as their distribution in training data and increase their weight by $b\%$ according to the distance between the ground-truth and the predicted congestion labels as $b = |\mathcal{C}_t^g - \mathcal{C}_t^p| \times 10 | b \in [10\% - 30\%]$. Hence, the model weights are readjusted accordingly to weighted training samples to learn the best parameters.

Now, at the end of the training, the model can predict $\mathcal{C}_t^p$; we are interested in identifying the sequences of maneuvers and road properties responsible for such prediction. Hence, we extend the model by making it interpretable as follows.

4) Generating interpretable sequences $\mathcal{L}^p$: ProCon ascertains the influence of specific maneuvers from the Markov chain and road properties on the predictions from the LSTM model. For this, we employ LIME (*Local Interpretable Model-agnostic Explanations*) method [17], which is a model-agnostic technique to faithfully explain any classifier's predictions by approximating them locally with an interpretable model. For each test sequence, LIME highlights the features (comprising the maneuvers and road properties) that played a pivotal role in the model's prediction. Thus, the significant transitions are identified, and following this, the corresponding maneuvers and road properties are decoded at given timestamps, forming a sequence $\mathcal{L}^p$. These sequences are stored in an online dictionary $\mathcal{D}$ (see Fig. 5) where congestion type $\mathcal{C}_t^p$ is the key and a set of sequences comprising maneuvers and road properties are populated as a value against the key. This way, major possible sequences responsible for a typical type of congestion will be captured, which can be further used in runtime for proactive prediction. For example, if the predicted congestion type is 2 denoting "*Moderate*" congestion, and a set of sequences are derived as {"*Sudden U-turn → Abrupt Deceleration by Peer Vehicles* given *Evening Time, Presence of Trucks*", "*Abrupt Stop → Lane Change → Relative Distance Variation* given *Morning Time, Presence of Pedestrians, High Traffic Density*",⋯}, it represents the possible causes behind the "*Moderate Congestion*" key. We use $\mathcal{D}$ during the runtime.

B. Opportunistic (re)Training

As traffic congestion has regional influence, the feature distribution might vary for an unseen region. Therefore, new sequences of maneuvers and road properties can always arrive at runtime. To make our model generalized enough to work and adapt itself at runtime, we incorporate *transfer learning* architecture [18]. Hence, we freeze all the layers except the last three to save on computational resources. The frozen layers will be used as a bootstrapped model in a new region, whereas, the last three will adjust their weight parameters to learn the differences in data distribution so that the loss function CCE incurs minimal loss.

While running the model from an edge device deployed on a vehicle, we continuously sense the GPS data to infer a significant change (increase) ϵ in the traffic density in Δt time interval. At that point, the collected IMU and GPS data are fed into our system to infer the congestion level $\mathcal{C}_t^p$, and its corresponding sequence of maneuvers and road properties are generated subsequently. This ensures our system can predict congestion proactively while saving computational resources. To identify new data points that the model needs to learn, we need both the ground-truth congestion labels $\mathcal{C}_t^g$ and observed data along with identification of erroneous prediction from the model. We leverage the fact that most sequences are derived during the training phase. Hence, for the sample test data, if we get a new sequence and the predicted $\mathcal{C}_t^p$ also does not exist in $\mathcal{D}$, we consider the sample as a retraining candidate.

However, we cannot use human annotators to collect $\mathcal{C}_t^g$ during the runtime as it can be biased or even malefic. Therefore, we use a decision-tree model trained on several road properties {*traffic density*, *vehicle speed*, *time of the day*, *speed limit*} to annotate $\mathcal{C}_t^g$ at runtime. This validation property is legit as it lies on popular digital maps inference techniques[3]. Following this, the model gets retrained and accordingly adapts itself.

C. Runtime Inference

Finally, given the road properties, our method periodically predicts the congestion level $\mathcal{C}_t^p$ at time $t_{\text{now}} + t$, where t_{now} is the current time. Suppose the predicted congestion level is higher than the current one (as returned by the Map API). In that case, *ProCon* checks $\mathcal{D}$ to figure out the chains of maneuvers $\mathcal{L}^p$, which can be responsible for an increasing congestion level. *ProCon* recommends that drivers avoid such maneuvers

[3]https://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html (Accessed: May 9, 2024)

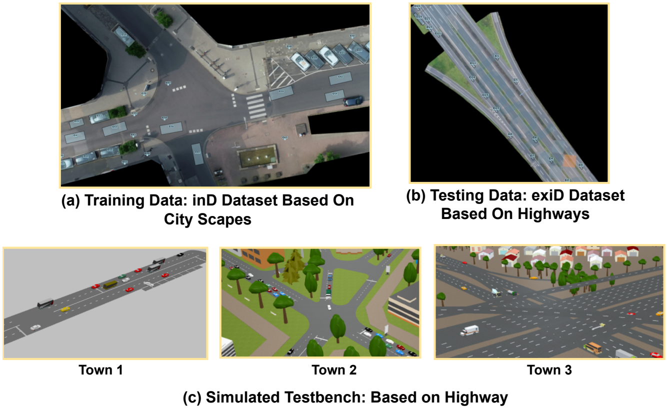


Fig. 6: Experimental setup with different datasets.

until the next prediction window. We analyze the impact of t during the evaluation. Notably, although avoiding maneuvers is recommended at a micro-level during the runtime, they capture a wide range of fine-grained recommendations and help to manage macro-scale traffic. For example, if turning towards an exit may reroute the vehicle, resulting in a high congestion probability at that route, *ProCon* recommends not to take the turn at that exit, thus even balancing traffic across different routes at a macro-scale.

V. Experimental Evaluation

In this section, we evaluate how well *ProCon* can proactively identify the variation in maneuvers and road properties to alert drivers about possible congestion formation. Notably, evaluating *ProCon* from real-time deployments is not feasible; hence, we use a simulator "AnyLogic" [11] to emulate driving behavior with a naturalistic driving environment along with two other real-world datasets as follows (see Figs. 6(a), (b)).

A. Experimental Setup

We train *ProCon* on a computing system with Intel(R) Core(TM) $i7-1255$U processor with 1.70 GHz speed, 16 GB primary memory, and 500 GB disk space. We use Python 3.8.3 with Tensorflow $2.13.0v$ running in the background to implement our model architecture. We use 1500 training samples from the inD dataset running over 200 epochs consuming 30% CPU, 3 GB of GPU, and 13.64% primary memory, running over ≈ 400 seconds. For the retraining architecture, we use exiD [10], which is a highway dataset extracted from German Autobahns, comprising 16 hours of driving data with 3 types of traffic participants as "*Car*", "*Truck*" & "*Van*", respectively. As proposed in §III-A3, the maneuver's distribution from the annotated sequences $\mathcal{L}^g$, responsible for congestion, differs significantly for inD and exiD datasets, respectively. For top-3 maneuvers "*Stop*", "*Lane Change*", and "*Relative Speed*", the distribution for inD and exiD datasets vary as {(32%, 16.4%), (12.3%, 17.2%), (8.2%, 22.8%)}, respectively. Certain differences in maneuvers for the two datasets, such as *stop pattern, relative distance* in city-street vs highway or others, further highlight the importance of the retraining architecture. We consider δ = 60 seconds, ϵ = 20%, and Δt = 10 minutes as the hyperparameters of *ProCon*. Our model takes 0.5 seconds with 10% CPU, 1 GB of GPU, and 12% primary memory consumption to run on each trip with retraining if required, thus making the system suitable to deploy at run-time.

For the simulation setup to test *ProCon* at runtime, we use Anylogic 8.8.4 simulator running over 20.0.2 JDK environment to simulate 3 urban traffic scenarios based on traffic intersections, comprising "*City Street*", "*Highway*", "*Parking Lot*", and a variety of traffic participants (see Fig. 6(c)). Traffic congestion due to cascading majorly originates from traffic intersections, in terms of longer waiting time, increased traffic density, etc., and propagates through connecting roads, eventually paralyzing the neighborhood road network [1]. As traffic intersections are a significant source of traffic phenomena such as congestion, we focus on emulating different intersection structures to achieve diversity and increased complexity. The urban cities with varying demography are emulated as follows - (i) A simple highway junction with 2 forward and 2 backward lanes with one-way traffic and traffic participants are cars, trucks, and vans, respectively (Town 1), (ii) a 5-way city road intersection with 2 lanes on each road and one-way traffic with traffic participants are pedestrians, cars, buses, respectively (Town 2) & (iii) a complex city with two 4-way road junctions with utmost 6 lanes and 2-way traffic, 2 flyovers and a variety of traffic participants as buses, lorries, trucks, cars, vans (Town 3). As these types of road intersections are mostly observed throughout different nations, the broad characteristics during proactive congestion predictions will be captured.

B. Baseline Implementation

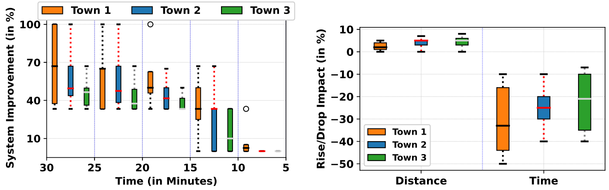
(1) We use a rule-based supervised Random Forest (RF) model with 50 decision trees, which branches out to an unlimited depth for training. Although RF model predictions can be explained using its feature importance map, we use LIME only to generate the sequences responsible for congestion. Using this baseline, we can compare both congestion levels and sequences of maneuvers.

(2) We implement DTW [19] that identifies anomalous trajectory of vehicles by adapting the model for congestion scenarios using inD [8] using hierarchical clustering. It captures individual vehicles' spatial & temporal dynamics and labels trajectories anomalous or normal. We compare how well *ProCon* can infer maneuvers responsible for congestion in contrast to the anomalous maneuvers inferred by DTW.

(3) We adapt an existing work *SPAT* [7] by incorporating $\mathcal{W}$ from current and all the neighborhood road segments to understand the spatial correlation between road segments on congestion formation and propagation from neighborhood roads. We leverage our model only with variable length input dimension, as $\mathcal{W}$ from each road is fed into a time-series format with the corresponding road's congestion level.

C. System Improvement

We start by inspecting the improvement in terms of congestion reduction if the vehicles are alerted to avoid maneuvers



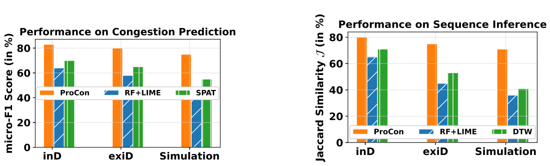
(a) Congestion reduction. (b) Impact on time & distance.

Fig. 7: Overall system performance.

based on congestion level prediction. We compute the improvement as $\frac{\mathcal{C}_t^p w - \mathcal{C}_t^p w/o}{\mathcal{C}_t^p w/o}$, where $\mathcal{C}_t^p w$ & $\mathcal{C}_t^p w/o$ are predicted congestion with and without the driving alerts, respectively, and t is varied between $[30-5]$ minutes indicating how early we want to predict the congestion levels. For each town in the simulation, we run the experiments for $60-90$ minutes with a minimum of 10 trips and report how system improvement varies with respect to t. As depicted in Fig. 7(a), Town 1 gets the highest improvement with a mean 67% if vehicles are alerted 30 minutes ahead. For Town 2 & 3, we get 56% & 45% mean improvement in avoiding congestion. However, the improvement is not worthwhile as the alert is generated later, i.e., 15 minutes ahead, even in Town 1 with a simple vehicle movement pattern. Vehicles get enough time to rectify their maneuvers for the former cases ($[30-20]$ minutes); however, a delayed prediction fails to avoid congestion cascading, thus exhibiting less improvements in avoiding congestion.

D. Impact of *ProCon* on Vehicles' Travel Time and Distance

While vehicles adhere to the recommended maneuvers to avoid congestion as prescribed by *ProCon*, it is vital to analyze if they need to travel more distance or time to reach the destination. Therefore, we measure such impact as rise or drop in the travel time and distance for each town's collective set of vehicles. We compare two scenarios, i.e., when vehicles travel based on Google map recommendation and when *ProCon* recommends suitable maneuvers to vehicles upon sensing congestion and report such impact in Fig. 7(b) as the percentage of rise/drop in travel distance/time. We observe vehicles have to travel a minimal distance (a mean of 3%) while they follow *ProCon*'s recommendation. However, we observe that the travel time reduces significantly when *ProCon*'s recommendations are used. Notably, the proactive approach not only reduces congestion on the road but also helps the vehicles to dynamically choose a route at runtime that reduces their overall travel times.



(a) Congestion prediction. (b) Congestion reasoning.

Fig. 8: *ProCon* performance in congestion analysis.

E. Performance of Congestion Analysis

ProCon's congestion analysis is evaluated from two aspects - (a) congestion prediction, (b) congestion reasoning inference. **Congestion Prediction:** We achieve a micro-F1 score [20] of 86% for 5-fold cross-validation and 83% for the test set for congestion prediction for inD dataset with a 70:15:15 train:validation:test split, as shown in Fig. 8(a). However, while training the data with inD and testing over exiD dataset and simulation, we achieve 80% and 75% micro-F1 score, respectively. Further, we observe that *ProCon* performs at least 25% better than the two other baselines for congestion prediction, over all the three datasets. *RF+LIME* could not adapt itself with the shift in data distribution while treating each data sample as an independent instance. In contrary, *SPAT* uses a rich set of road and traffic properties; so, it could predict the congestion instances that are due to narrow roads or high volume of traffics. However, it failed to predict the congestion cascading due to the influence of the driving behavior.

Congestion Reasoning Inference: For measuring the accuracy as how correctly the interpretable sequences are generated, we use Jaccard Similarity [21] measure between the generated $\mathcal{L}^p$ and annotated $\mathcal{L}^g$ sequences as $\mathcal{J} = \frac{\mathcal{L}^g \cap \mathcal{L}^p}{\mathcal{L}^g \cup \mathcal{L}^p}$. Fig. 8(b) shows the results for all the three datasets and also compare them against two baselines (§V-B). We obtain $\mathcal{J}$ as 80%, 75% & 71% for inD & exiD datasets and simulation, respectively. But, for the two baselines, especially for *RF+LIME*, $\mathcal{J}$ is worse due to temporal independence. In contrast, DTW performs quite well for the inD dataset (71%) as it considers the temporal ordering of vehicle trajectory information but fails for other datasets due to non-adapting model architecture. Also, the high values of $\mathcal{J}$ for all the datasets denote a decent overlapping between human-annotated and model-generated inference for predicting congestion, thus making *ProCon* robust enough.

F. Ablation Study

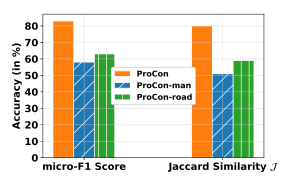


Fig. 9: Ablation study.

Next, we inspect how *ProCon* performs while excluding two feature categories, i.e., maneuver chain ($\mathcal{N}$) and weight of the road ($\mathcal{W}$), one at a time. We call these two variations as ***ProCon*−man** where we exclude $\mathcal{N}$ and ***ProCon*−road** where we exclude $\mathcal{W}$. For both the variants, we report micro-F1 score and Jaccard similarity for the test set of the inD dataset in Fig. 9. Although ablating both the features one at a time drops both congestion prediction and its reasoning inference quite poorly, still the drop is severe for *ProCon*−man. The analysis indicates that both the feature categories have their importance on the model performance.

VI. Related Work

Recent works on estimating traffic congestion analysis [4], [5] focus on crowd propagation patterns from diverse traffic network properties, e.g., spatial relation between roads. Multiple sources responsible for congestion are studied in [6], showing influences of different zone types, sudden accidents, etc. Utilizing traffic volume, road connectivity, etc., [22] extrapolate upcoming traffic congestion by anticipating a sudden rise in traffic density. Additionally, [23] has developed a model to predict the congestion propagation time. However, all such approaches require a diverse set of information, sometimes in a centralized manner and modification of road architecture, optimizing traffic rules/signs, and changing traffic policies.

In contrast, different modalities such as spatial correlation among road segments [7], GPS probes and crowd-sourced tweet information [24], vehicle density, road capacity, vehicle speeds, etc. are fused for predicting causes of congestion. However, this happens reactively in retrospect and works well for recurring congestion scenarios. However, attributing driving behaviors to non-recurrent congestion [25] is very challenging because they tend to have random patterns [26] as shown by SOTA methods. Already existing works [27] show a strong relation between dynamic traffic behavior and congestion. Therefore, understanding the pattern of individual driving behaviors in the context of the surrounding road network and its effect on neighboring vehicles will establish a causal relationship between maneuvers and eventual congestion. Learning this relationship will allow us to prevent maneuvers leading to non-recurring congestion. *ProCon* develops a system in this direction by mitigating the limitations of the existing approaches leveraging such causal relationships.

VII. Conclusion

This study developed a multimodal domain adaptive model called *ProCon* that proactively predicts the possibility of congestion using the contextual impact of different driving maneuvers and surrounding road information. Subsequently, drivers are alerted to avoid such maneuvers to prevent congestion formation during navigation. Alerting the drivers well ahead of time and adhering to our recommendations shows the credibility and efficiency of *ProCon* by reducing congestion for most traffic scenarios. Although dependency on driving maneuvers and road properties are well-captured, the effect of some unanticipated events, e.g., sudden natural or man-made disasters (say, thunderstorms or mob violence), are not cosidered in the current design of the model. Analyzing such information needs additional modalities, such as fusing social network information (say, tweets); we look forward to them as potential future directions.

Acknowledgement: This work was done while D. Das was visiting the Missouri University of Science and Technology during May-October 2023 as a Visiting Scholar.

References

[1] Y. Liang, Z. Jiang, and Y. Zheng, "Inferring traffic cascading patterns," in *25th ACM SIGSPATIAL GIS*, 2017, pp. 1–10.

[2] Y. Ma, X. Gu, Y. Yu, A. J. Khattakc, S. Chen, and K. Tang, "Identification of contributing factors for driver's perceptual bias of aggressive driving in china," *Sustainability*, vol. 13, no. 2, p. 766, 2021.

[3] L. Fei, H. Zhu, and X. Han, "Analysis of traffic congestion induced by the work zone," *Physica A: Statistical Mechanics and its Applications*, vol. 450, pp. 497–505, 2016.

[4] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino, "CrowdMeter: Congestion level estimation in railway stations using smartphones," in *IEEE PerCom*, 2018, pp. 1–12.

[5] X. Xie, Y. Yang, Z. Fang, G. Wang, F. Zhang, F. Zhang, Y. Liu, and D. Zhang, "coSense: Collaborative urban-scale vehicle sensing based on heterogeneous fleets," *ACM IMWUT*, vol. 2, no. 4, pp. 1–25, 2018.

[6] M.-A. Gorji, M. Akbarzadeh, and S.-N. Shetab-Boushehri, "Evaluation and improvement of the urban transportation networks resilience in short-term non-recurring traffic congestion: a novel graph connectivity-based criteria," *Transportation Engineering*, vol. 10, p. 100152, 2022.

[7] X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao, "Traffic congestion prediction by spatiotemporal propagation patterns," in *20th IEEE MDM*. IEEE, 2019, pp. 298–303.

[8] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *IEEE IV*, 2019, pp. 1929–1934.

[9] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *IEEE CVPR*, 2020, pp. 2636–2645.

[10] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exiD dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *IEEE IV*, 2022, pp. 958–964.

[11] A. Borshchev, "Anylogic 7: New release presentation," in *Proceedings of the 2013 WSC: simulation: making decisions in a complex world*, 2013, pp. 4106–4106.

[12] Z. Pan, A. Sharma, J. Y.-C. Hu, Z. Liu, A. Li, H. Liu, M. Huang, and T. Geng, "Ising-traffic: Using ising machine learning to predict traffic congestion under uncertainty," in *37th AAAI*, 2023, pp. 9354–9363.

[13] A. T. Kwee, M.-F. Chiang, P. K. Prasetyo, and E.-P. Lim, "Traffic-cascade: Mining and visualizing lifecycles of traffic congestion events using public bus trajectories," in *27th ACM CIKM*, 2018, pp. 1955–1958.

[14] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, and M. Li, "Fine-grained abnormal driving behaviors detection and identification with smartphones," *IEEE Trans. on Mobile Computing*, vol. 16, no. 8, pp. 2198–2212, 2016.

[15] D. Das, S. Pargal, S. Chakraborty, and B. Mitra, "Why slammed the brakes on? auto-annotating driving behaviors from adaptive causal modeling," in *IEEE PerCom Workshops*, 2022, pp. 587–592.

[16] B. Yu and M. P. Singh, "An evidential model of distributed reputation management," in *ACM AAMAS*, 2002, pp. 294–301.

[17] M. T. Ribeiro *et al.*, ""why should I trust you?" explaining the predictions of any classifier," in *22nd ACM KDD*, 2016, pp. 1135–1144.

[18] W. Shao *et al.*, "FADACS: A few-shot adversarial domain adaptation architecture for context-aware parking availability sensing," in *IEEE PerCom*, 2021, pp. 1–10.

[19] J. Zhao and M. Sartipi, "Automatic identification of anomalous driving events from trajectory data," in *25th IEEE ITSC*, 2022, pp. 851–856.

[20] Z. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize f1 measure," in *PKDD*, 2014, pp. 225–239.

[21] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 44, pp. 223–270, 1908.

[22] A. M. Nagy and V. Simon, "A novel congestion propagation modeling algorithm for smart cities," *Pervasive and Mobile Computing*, vol. 73, p. 101387, 2021.

[23] W. Zhong, Q. Suo, A. Gupta, X. Jia, C. Qiao, and L. Su, "MetaTP: Traffic prediction with unevenly-distributed road sensing data via fast adaptation," *ACM IMWUT*, vol. 5, no. 3, pp. 1–28, 2021.

[24] S. Wang, L. He, L. Stenneth, S. Y. Philip, Z. Li, and Z. Huang, "Estimating urban traffic congestions with multi-sourced data," in *17th IEEE MDM*, vol. 1. IEEE, 2016, pp. 82–91.

[25] S. Wang, S. Djahel, Z. Zhang, and J. McManis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2888–2899, 2016.

[26] Y. Li, H. Hou, Z. Dong, Y. Zang, Y. Zhang, and Y. Song, "Spatiotemporal analysis of static and dynamic traffic elements from road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3297–3311, 2022.

[27] L. Qi, M. Zhou, and W. Luan, "Impact of driving behavior on traffic delay at a congested signalized intersection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1882–1893, 2016.