Subcycle Waveform Modeling of Traffic Intersections Using Recurrent Attention Networks

Yashaswi Karnati, Rahul Sengupta, Anand Rangarajan, *Member, IEEE*, and Sanjay Ranka, *Fellow, IEEE*

***Abstract*— Traffic flow dynamics in the vicinity of urban arterial intersections is a complex and nonlinear phenomenon, influenced by factors such as signal timing plan, road geometry, driver behaviors, etc. Predicting such flow dynamics is an important task for urban traffic signal control and planning. Current methods use microscopic simulation for studying the impact of a large number of signal timing plans at each of the intersections. A major drawback of microscopic simulation is that they are they are based on source destination traffic generation models and cannot incorporate the high resolution loop detector data such as that are provided by automated traffic signal performance measures (ATSPM) based systems. The arrival (or departure) information of each vehicle on a detector can be thought of as a time series waveform. Given the high granularity of ATSPM data, this waveform can be used to several interesting analyses. The waveforms can be used to derive information on platoon dispersion as vehicles progress across the corridor. Also, these waveforms can be modelled to understand how the vehicles progress across the corridor for a variety of signal timing plans. In this paper, we show that deep neural networks based machine learning systems can be used to effectively leverage the waveforms collected at multiple sensors (stopbar and advanced) on the intersection to model the traffic dynamics both at an intersection and across intersections. We show that modelling of these waveforms can be useful to understand traffic flow dynamics under different signal timing plans and can be potentially integrated into signal timing optimization software. Further, these methods are three to four orders of magnitudes faster than using microscopic simulations.**

***Index Terms*— Traffic, intersection, waveform, neural networks, deep learning, feed-forward neural networks, recurrent neural networks, teacher forcing, signal timing optimization.**

I. Introduction

MITIGATING traffic congestion and improving safety are the important cornerstones of transportation for smart cities. Despite significant advances in vehicle technology, traffic engineering practices, and analytics-based solutions, traffic congestion is still a major problem. Traffic congestion resulted in nearly $305 billion in congestion costs and caused Americans to lose 97 hours per person in congestion.

Manuscript received January 29, 2021; revised August 26, 2021 and October 5, 2021; accepted October 15, 2021. Date of publication November 1, 2021; date of current version March 9, 2022. This work was supported in part by the National Science Foundation, Division of Computer and Network Systems under Grant 1922782. The Associate Editor for this article was T.-H. Kim. *(Corresponding author: Yashaswi Karnati.)*

The authors are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yashaswikarnati@ufl.edu; rahulseng@ufl.edu; anandr@ufl.edu; sranka@ufl.edu).

Digital Object Identifier 10.1109/TITS.2021.3121250

Reducing congestion requires coordinating the signal timing plans of each intersection on a corridor or network so that most of the vehicles do not have to wait at traffic intersections (effectively by changing the cycle length, splits and offsets). This is a complex problem as changing the signal timing on one intersection affects the traffic on the adjacent intersections. Thus, it requires choosing the appropriate combination of signal timing plans for each intersection on a corridor or a network. Microscopic simulations are useful in understanding the correlated impact of signal timing plan at multiple intersections on the overall performance of traffic movement. In particular, they are able to model the output traffic patterns (along all outbound lanes) of an intersection based on the traffic patterns of the input lanes. Additionally, they are able to capture the dispersion of platoons when the vehicles move from one intersection to another. Both of these are important requirements for corridor or network optimization.

A major drawback of microscopic simulation is that they are they are based on source destination traffic generation models and cannot incorporate the high resolution loop detector data such as that are provided by automated traffic signal performance measures (ATSPM) [9] based systems. Unlike origin-destination models these systems can provide traffic arrivals and departures at stop bar detectors (and in many cases advanced detectors) at a decisecond level (and are effectively much more precise than the origin-destination models in capturing the traffic variations throughout the day). Although, some simulators allow this for a single intersection, they cannot achieve this for corridor and network simulation where neighboring intersections have input-output relationships. Additionally, microscopic simulation, by their nature, are sequential in a nature and the time requirements are proportional to the number of vehicles in the system and are relatively slow.

The arrival (or departure) information of each vehicle on a detector can be thought of as a time series waveform (see Figure 1). Given the high granularity of ATSPM data, this waveform can be used to derive information about platoons (multiple vehicles passing without significant distance) or gaps (no vehicles passing through for a duration). In this paper, we show that deep neural networks based machine learning systems can be used to effectively leverage the waveforms collected at multiple sensors (stopbar and advanced) on the intersection to model the traffic dynamics both at an intersection and across intersections.

In particular, using these waveforms at stopbar and advanced detectors:

1) We develop models to both impute the traffic waveform from each inbound direction to an intersection as well as the traffic waveform to each outbound direction from the intersection conditional on the signal timing plan. The input waveforms in all the directions can be used to understand if the current signal timing is near optimal. The models can be used to model the vehicle progression to downstream intersections and estimate performance measures for different signal timing plans. However, since they use data that is directly measured based on the traffic sensors in the network, they are a much more accurate indicator of traffic movement than imputed origin destination pairs.
2) We develop models that can predict the dispersion along a road segment (exit from one intersection to entrance of the neighboring intersections) more accurately than the Robertson model [15] that is traditionally used (cf. Section VII for more details on other modeks). This is because our models, like microscopic simulation models, can effectively capture non-uniform velocities of vehicles as well variation in driver behaviors.
3) We develop models that can predict the impact of signal timing of the downstream intersection on platoons as they arrive close to a downstream intersection. The output waveform from a given intersection along with the output waveform on the downstream also be used to understand the leakage or addition of traffic during a short time period.

Thus, our models can capture both the local (i.e., near an intersection) traffic flow dynamics as well as coupled traffic flow dynamics (i.e., between two consecutive intersections) and are significant extensions of the prior work on platoon dispersion models (cf. Section VII). We develop and provide multi-scale error measures to demonstrate that our predictions are accurate and comparable to microscopic simulation.

Our models use novel deep learning based architecture with attention layers [4] and teacher forcing [21] that is specifically designed to model and predict the behavior of input and output behavior at an intersection using advance and stop bar high resolution loop detector data and signal timing information. The use of our GPU implementation of deep neural networks can generate accurate predictions at three to four orders of magnitude faster than using microscopic simulations for this purpose.

Thus our methods are novel and leverage both the recent advent of automated traffic signal performance measures (ATSPM) [9] systems provide this information and signal timing information at a decisecond level and the availability of cheap GPU-based computing and deep neural network algorithms to model traffic behavior. Based on our detailed literature survey, we have not found any related work in predicting outflow waveforms or imputing input waveforms at a subcycle level using neural networks or related techniques. Most of the previous work is in volume prediction or predicting flows at downstream intersections. We outline this in Section VII.

Adaptive traffic signal control software's used in practice (TRANSYT-7F, SCOOT etc.,) leverages Robertson platoon dispersion models to predict vehicle arrival rate at downstream intersection and use that to calculate optimal signal timing parameters like cycle length, offset, green splits etc. The effectiveness of these tools depends on platoon dispersion models, how well they predict the progression of vehicles downstream which our models can provide better accuracy. Thus, our methods can be integrated into other signal timing optimization frameworks, such as those based on linear models and reinforcement learning for corridor and network coordination optimization.

The rest of the paper is outlined as follows. Section VII presents the related work of different techniques used for traffic state estimation. In Section II, we briefly introduce several common terms used in deep learning. In Section III, we present different models that we develop to relate various observable and unobservable quantities. To test our approaches using realistic waveforms, we input real-world recorded traffic flow data to a microscopic simulator and leverage parallel computing to generate a large ($\tilde{4}$0 million cycles) dataset at 5-second resolution. Section IV describes how we preprocess raw data and generate synthetic datasets from real-world controller logs data using SUMO. In Section V, we describe the architecture of the dual attention encoder-decoder model. Experimental results are provided in Section VI, and conclusions in Section VIII.

II. Preliminaries

A. Deep Learning

Artificial neural networks are a class of machine learning model inspired by biological neural networks. They consist of systems of interconnected units, with each unit (called "neurons") taking multiple inputs and emitting a single output, based on an activation function. Neurons are connected to each other in layers and are trained on a dataset via the back-propagation rule.

With the advent of powerful computing technologies such as graphics processing units (GPUs) and tensor processing units (TPUs) and economical large cloud storage technologies, large multilayer neural networks (with thousands to millions of tunable parameters) are being trained on equally large datasets running into 100s of gigabytes. This paradigm is referred to as "deep learning." Deep learning can be understood as a representation-learning method that consists of neural networks that learn a hierarchy of representations, ranging from simple features to more abstract ones.

B. Gated Recurrent Units

A recurrent neural network (RNN) [5] is a class of artificial neural network, which is especially well-suited for modeling temporal sequences. These networks process input sequences within the context of their internal hidden state ("memory") in order to arrive at the output. The internal hidden state is an abstract representation of previously seen inputs. Thus, they are capable of dynamic contextual behavior.

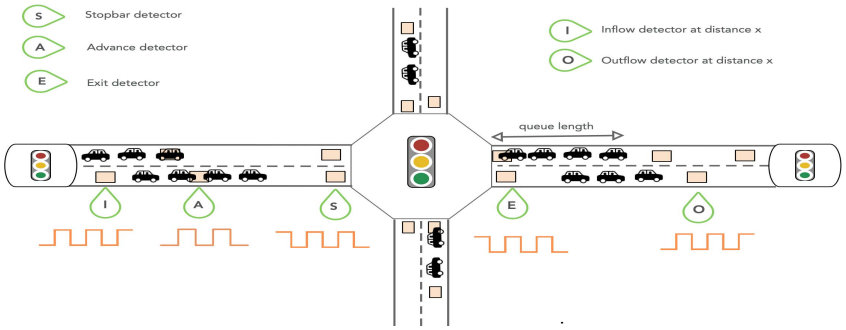

Fig. 1. Generic single-lane intersection. In real world, the vehicle waveforms is generally observed at stop bar, advance detectors. We also introduce a set of virtual detectors - exit, inflow, outflow detectors. These are the locations where the waveforms are imputed so that we can model the traffic flow dynamics between any pair of intersections.

Our task at hand involves processing several temporal sensor streams (of detector and signal state readings), and thus, GRU (a version of RNN) [6] is an appropriate model to use.

Teacher forcing [21] is a common training technique for training RNNs (and thus GRUs). In teacher forcing, the actual answer of the previous time step is provided to the RNN while it predicts the current output. An error in the previous output could cause a large error in the current output, which in turn would accumulate over time steps. Teacher forcing remedies this by penalizing the network for the wrong answer at that time step, but doesn't allow the network to commit a series of errors based on the initial error. This technique has been shown to lead to faster convergence. Given our large dataset and models, we employ this technique to speed up model training.

C. Attention Mechanism

Attention mechanism [4] is a deep learning technique for effectively dealing with long-range dependencies in neural models. The broad idea is to create linkages between the current context vector (which, in the case of GRUs, would include the last hidden state) and the entire source input (or its abstract representation). Thus, the context field of the model is enhanced and is no longer prone to forget events in the distant past. Attention mechanism is especially useful for our task because the discharge profile at one approach is correlated in time to what happened in previous cycles.

III. Proposed Models

Vehicle loop detectors that have traditionally been deployed at intersections to detect the passage of vehicles can measure the absence or presence of vehicles passing above them. The arrival (or departure) information of each vehicle on a detector can be thought of as a time series waveform (see Figure 1). This waveform can be used to provide information about platoons (multiple vehicles passing without significant distance) or gaps (no vehicles passing through for a duration. These waveforms are only available at advance detectors and stop bar detectors, as there are typically no detectors available at inputs and outputs (some U.S. states have these detectors, but most do not).

To model the progression of vehicles between intersections, in terms of input and output waveforms, we introduce a set of virtual detectors placed at an intersection (Figure 1) - Exit detector, Inflow detector, Outflow detector. The underlying idea is that we train the neural network models based on simulated data to be able to impute waveforms at these virtual detectors using data observed at stop bar and advance detectors. Imputing waveforms at these virtual detectors helps us model the traffic flow dynamics between a pair of intersections independent of distance between them and also understand the progression of vehicles for a variety of signal timing plans (the waveforms observed at stop bar and advance detectors is correlated with signal timing). Our focus is on making predictions of these waveforms at the granularity of approximately 5-10 seconds (generally, this corresponds to a range of 0-3 vehicles). Our experimental results show that we can achieve this with a high level of accuracy.

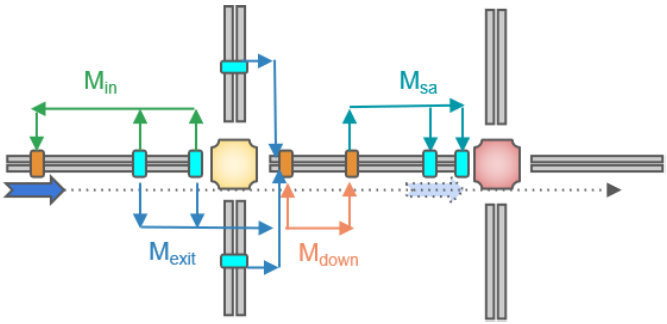

Fig. 2. Diagram showing input, output relationship's for different models as proposed in Section III. We decompose the modelling to capture both the local (i.e., near an intersection) traffic flow dynamics as well as coupled traffic flow dynamics (i.e., between two consecutive intersections). M_{exit} - To model the waveform at exit detectors. M_{in} - To reconstruct inflow waveform. M_{down} - To predict progression of exit waveform towards downstream intersection. M_{sa} - To impute waveform at stopbar, advance detectors.

In this work we propose non-parametric neural networks to model the progression of vehicles between signalized intersections. We decompose the modelling of platoon dispersion between intersections using 4 different models (Figure 2) based on where the observations and collected and where predictions are made. Description of each these models is presented in the following subsections. Figure 2 shows input-output relationships for different models. Given waveforms at stop bar and advance detectors and signal timing information, the models should be able to predict waveforms at the exit

of the intersection and the waveform at a certain distance downstream (outflow) and to reconstruct inflow at a certain distance upstream. The description for each of the models follows.

A. Stop-Bar-to-Exit Waveform Prediction Model (M_{exit})

This model predicts the waveform that exits the intersection, given (1) the waveforms at stop bar detectors (S) and advance detectors (A) along all the directions, (2) the signal timing plan of the intersection and (3) turning movement counts. This model captures the local effects at the intersection. In practice, intersections do not always have an exit detector, but nevertheless, we instantiate one in our simulator in order to gain an in-depth understanding of flow dynamics local to an intersection.

B. Inflow Waveform Reconstruction Model (M_{in})

This model reconstructs the unperturbed inflow waveform at an intersection, given the observed waveforms at the intersection's stop bar and advance detectors and the signal timing plan. The inflow waveform can be thought of as the incoming waveform that has exited the upstream intersection and is still sufficiently far away from the intersection of interest and thus is not yet affected by the queues and signal timing plan of the intersection of interest.

An important application for reconstructing the inflow is to potentially use it for signal timing optimization. The observed stop bar and advance detector waveforms are heavily correlated with the observed signal timing plan and heavily affected by the queue behavior and thus cannot be directly used for signal timing optimization.

C. Exit-to-Downstream Waveform Prediction Model (M_{down})

This model predicts the modification of the exited waveform as it travels downstream to the next intersection. Given an exit waveform, the model aims to predict the waveform at a certain distance downstream from the intersection (not affected by downstream signal state). This waveform is treated as the inflow waveform to the downstream intersection.

Instead of predicting the waveform at downstream advance and stop bar detectors, we employ a two-step strategy: use the exit waveform to predict the downstream inflow waveform and use the predicted inflow waveform to predict waveforms at downstream advance and stop bar detectors. In this way, we can model the interactions between any pair of intersections independent of the distance between them.

D. Stop-Bar-Advance Waveform Model (M_{sa})

This model predicts the waveforms at stop bar and advance detectors, given the unperturbed inflow waveform and the signal timing plan. We can also use this model to verify that the inflow reconstruction done by M_{in} is of sufficient quality to be used to replicate the same observed stop bar and advance waveforms.

The waveforms are represented as 1-D vectors, each with T components. Here T refers to the length of time a particular sensor's data is being considered, with each component being aggregated at a 5-second level. In our work, T = 150, i.e., each data vector corresponds to 750 seconds of data (roughly 6-7 cycles), aggregated at the 5-second level. The time gap between two vehicles (headway) near an intersection is usually 2 seconds, which leads to an average of 2-3 vehicles per 5-second interval. We find this level of aggregation sufficiently expressive to capture platoon dynamics and at the same time not overly compute-intensive to train our models. Signal timing information is encoded using eight vectors, each with T components, all either 1s or 0s, 1 indicating that a particular direction is green at that time interval (a typical intersection has eight phases or directions of vehicular movement).

TABLE I
TABLE DESCRIBING NOTATIONS OF DIFFERENT VARIABLES USED IN THE MODELLING

Name	Description	Agg. level	Dimension	Type
S	Waveform at stopbar detector	5 sec	1x150	Integer (0-5)
A	Waveform at advance detector	5 sec	1x150	Integer (0-5)
SIG	Signal timing information	5 sec	8x150	Binary (0,1)
TMC	Turning movement counts ratio	750 sec	1x12	Integer
INF_d	Inflow waveform at distance d upstream the intersection	5 sec	1x150	Integer (0-5)
EXIT	Waveform at virtual exit detector	5 sec	1x150	Integer (0-5)
OUT_d	Outflow waveform at distance d downstream the intersection	5 sec	1x150	Integer (0-5)

TABLE II
TABLE DESCRIBING INPUT AND OUTPUT VARIABLES FOR DIFFERENT MODELS AS SHOWN IN FIGURE 2

Name	Description	Inputs	Outputs
M_{exit}	To predict waveform at exit detector	S, A, SIG, TMC	OUT_d
M_{in}	To reconstruct inflow waveform	S, A, SIG	INF_d
M_{down}	To predict progression of exit waveform towards downstream intersection	OUT_0, d	OUT_d
M_{sa}	To impute waveform at stopbar, advance detectors	INF_d, SIG	S, A

TABLE III

TABLE SHOWING SAMPLE OF RAW EVENT LOGS FROM SIGNAL CONTROLLERS. MOST MODERN CONTROLLERS GENERATE THESE DATA AT A FREQUENCY OF 10 HZ

SignalID	Timestamp	EventCode	EventParam
1490	2018-08-01 00:00:00.000100	82	3
1490	2018-08-01 00:00:00.000300	82	8
1490	2018-08-01 00:00:00.000300	0	2
1490	2018-08-01 00:00:00.000300	0	6
1490	2018-08-01 00:00:00.000300	46	1
1490	2018-08-01 00:00:00.000300	46	2
1490	2018-08-01 00:00:00.000300	46	3

As mentioned earlier, we use simulated data for training neural network models to predict these waveforms at a sub-cycle level. The main reason for using simulated data is that many unobserved quantities like queue lengths or waveforms at exit detectors or outflow detectors can be captured.

IV. DATA GENERATION

Induction loop detectors collect high resolution data that provide information such as whether a vehicle passed over them or not, intersection behavior, and timing pattern.

Table III shows a sample of high resolution data. The data consist of the following attributes:

1) SignalID: Intersection identifier
2) Timestamp: Time at which the event was logged (decisecond resolution)
3) EventCode: What event at the signal was captured
4) EventParam: Value of the event or attribute at that time-stamp.

These data also come with metadata which describe the different event codes and event parameters; for example, event code 81 indicates a vehicle departure, and event code 2 indicates start of green phase. An event parameter identifies the particular detector channel or phase in which the event was captured. The dataset we used consists of controller log data from 329 signalized intersections in Seminole County, Greater Orlando Metropolitan Area.

In order to model an intersection, we need a large dataset to train our models. While several real-world datasets exist [19], they are of limited utility for our task because they do not model intersections under diverse traffic and signal timing conditions. In the real-world, it is highly unlikely a traffic authority would implement undesirable signal timing schemes for gathering data because that would have adverse real-world consequences. On the other hand, microscopic simulators offer us the flexibility of implementing undesirable signal timing plans. They also implement reasonable approximations of real-world vehicle behaviors. However, building simulation scenarios (maps, vehicle flows etc.) and running them is often a time-consuming process. We use Simulator of Urban Mobility (SUMO), an open source microscopic traffic simulator [12]. Instead of using SUMO's built-in modules for programming flows either using random flows or from origin-destination (OD) matrices, we use waveforms from the real world for running the simulations.

TABLE IV

TABLE SHOWING MINIMUM AND MAXIMUM GREEN TIMES FOR ACTUATED SIGNAL TIMING PLAN PHASES USED IN OUR SIMULATIONS

Traffic Movement	Minimum Green Time (seconds)	Maximum Green Time (seconds)
Corridor Through and Right	20	70
Side Through and Right	10	30
Corridor Left	10	30
Side Left	10	30

We use a three-stage approach for generating simulation data:

1) Generate a realistic intersection configuration in SUMO
2) Derive traffic waveforms from real data
3) Run parallel simulations with adaptive control using waveforms from 2 and intersection configuration in 1.

These are described in the following subsections.

A. Intersection Configuration

Our simulation consists of a one-intersection scenario with four approaches based on standard NEMA (National Electrical Manufacturing Association) phasing [1]. It consists of four through and right movements and four left-turn movements, one of each for the four approaches. Most urban arterials have an exclusive left-turn buffer at each approach to cater to left-turning traffic. This prevents the left-turning traffic from blocking the through and right traffic until the buffer is filled.

Each approach is initially a single lane which fans out into a through-lane and an exclusive left-turn buffer. The left-turn buffer extends 60 meters and can hold 6-7 vehicles. There are two stop bar detectors per approach, one for through and right traffic and one for left-turn lanes. There is one advance detector 90 meters from the intersection, just beyond the end of the left-turn buffer. Multiple lanes for each movement group can be handled by (a) aggregating detector counts per movement group and (b) training multiple models, one for each intersection geometry of interest. In this study, we only focus on the most general and minimal configuration.

In addition, we also place additional detectors at 500 meters upstream from the intersection to capture the unperturbed incoming inflow waveform. We also place additional exit detectors to capture the exit waveform as it exits the intersection, along outbound approaches.

In order to gather downstream data, we place gating traffic signals that mimic a downstream intersection 800 meters from the main intersection. They are simply one-phase signals without any side streets. There are four such gating signals, one along each of the four outbound directions.

The signal timing plan for the intersection is an actuated signal timing plan with minimum and maximum times, as shown in Table IV. The maximum is chosen with consideration of acceptable pedestrian wait times. The gating signals have a single phase with red time of 60 seconds and a green time of 55 seconds, interleaved with a yellow time of 5 seconds, with variable offsets with regard to the main intersection.

TABLE V

TABLE SHOWING FLOW VOLUME BOUNDS. GIVEN THAT WE INTEND TO MODEL ARTERIAL STREETS, WE ENSURE THE BULK OF THE TRAFFIC FLOW IS ALONG THE CORRIDOR. WE RANDOMLY CHOOSE TRAFFIC FLOW VOLUMES THAT ARE SPLIT BETWEEN THE MINIMUM AND MAXIMUM BOUNDS

Traffic Movement	Lower Main Flow Volume Partition Bound (%)	Upper Main Flow Volume Partition Bound (%)
Corridor Through and Right	40.00	72.72
Side Through and Right	20.00	9.09
Corridor Left	20.00	9.09
Side Left	20.00	9.09

Given that we intend to model arterial streets, we ensure the bulk of the traffic flow is along the corridor. We randomly choose traffic flow volumes that are split between the minimum and maximum bounds as shown in Table V.

Thus, the main flow along the corridor through and right direction will be between two and eight times the flow along the other streets. These ratios are based on the observed traffic flows in the recorded Orlando dataset.

B. Input Traffic Generation

We use advance detector logs from the Orlando dataset to generate vehicle flows at a 1-second resolution. We randomly sample flow patterns observed at these two detectors for the straight and side streets, and ensure they fall between the above-mentioned volume flow constraints. We program these arrival patterns in the SUMO [12] microscopic simulator. These patterns are further shaped by the gating signals at the start of the four incoming approaches. This ensures variable platooning of volumes and inflow and outflow distributions based on real-world data.

C. Parallel Dataset Generation

The data generation process makes use of a multiprocessing environment. At any instant, several simulations will be running in parallel as each thread runs a simulation, processes the logs, and dumps the dataset into the file system.

Each simulation generates logs which have information of every time step of the simulation. These logs are processed, and the following information is stored:

- Waveforms at all the detectors for all the approaches
- Signal timing information
- Queue length for all the approaches
- Turn movement counts for all the possible movements

Within a simulation, after an initial simulation warm-up of 600 seconds, logs are extracted in windows of 1,000 seconds. These usually contain 8-9 complete cycles on average. Thus, each data exemplar consists of a set of waveforms of different signals and detectors, queue lengths, and turn movement counts for a window of 1,000 seconds ($T = 200$), aggregated at 5-second resolution.

A large dataset of 5 million such exemplars is thus generated, accounting for 40 million traffic cycles of simulation. The dataset is then split into training sets and testing sets in the ratio of 70:30.

The creation of such a vast dataset involved considerable engineering effort. The entire pipeline was implemented in the Python programming language. A multiprocessing library was used to run up to 60 parallel instances of SUMO and preprocess output XML logs in batches. Numpy and Dask were then used to create vectors for training and testing. These vectors were stored in HDF5 format using the H5PY library.

Implementation, training, and evaluation of the deep learning models was done using the PyTorch [14] library. The University of Florida's HiPerGator supercomputing resources were used to train and test multiple models in parallel.

V. PROPOSED NEURAL NETWORK ARCHITECTURES

In this section, we describe the architecture of the Dual Attention Encoder-Decoder model. Figure 4 shows the proposed architecture. The model has four components: encoder, decoder, temporal attention module, and phase attention module. The encoder takes the waveforms at all the stop bar and advance detectors and generates a hidden representation. The decoder outputs the value of the output waveform at each time step. The attention modules help the network to concentrate on relevant temporal and spatial information. The architecture is described below.

The encoder is a GRU layer with 50 hidden units; input to it is of the size (*batch_size x no._of_input_variables x no._of_time_intervals*). In the forward pass, the last hidden state of the encoder(*batch_size x 50*) along with all hidden states (*batch_size x 50 x no_time_intervals*) is returned. The last hidden state of the encoder is used to initialize the initial hidden state of decoder. If only the last hidden state is passed through the decoder, it has the burden of representing the waveform across all the time points.

Attention modules allow the model to focus on specific parts of the encoder outputs based on the decoder's outputs. The temporal attention module is a feed-forward layer using the current decoder output and hidden state as inputs, and the output is a vector, (*batch_size x no._of_time_intervals*), which comprises attention scores representing the importance of each hidden state of the encoder for the current prediction. The phase attention module is another feed-forward network that uses the signal timing vector and decoder hidden state as inputs to generate attention scores (*batch_size x 50*). The phase attention score represents the importance given to each of the hidden units. These attention scores are multiplied by encoder outputs to create a weighted combination and then passed through the decoder.

The decoder is also a GRU layer, taking the weighted combination from both the attention modules as input (*batch_size x no._of_time_intervals+50*). The decoder outputs a prediction for the next time point and updates its hidden state. This predicted value is used by the attention module to generate attention scores for the next time step.

This architecture was inspired by recent advances in sequence-to-sequence models, namely transformer networks. Transformer networks also broadly use the encoder-decoder paradigm with attention. However, they trade RNN-based

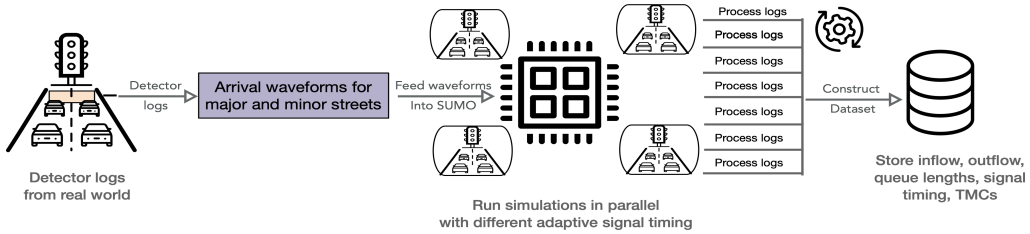

Fig. 3. Figure showing overall simulation framework for data generation. We use recorded controller log data from real world intersections and use that to run multiple simulation in parallel with different signal timing plans. The simulation logs are processed and waveforms at stop bar, advance, exit, inflow, outflow detectors along with signal timing information is stored for creating the dataset.

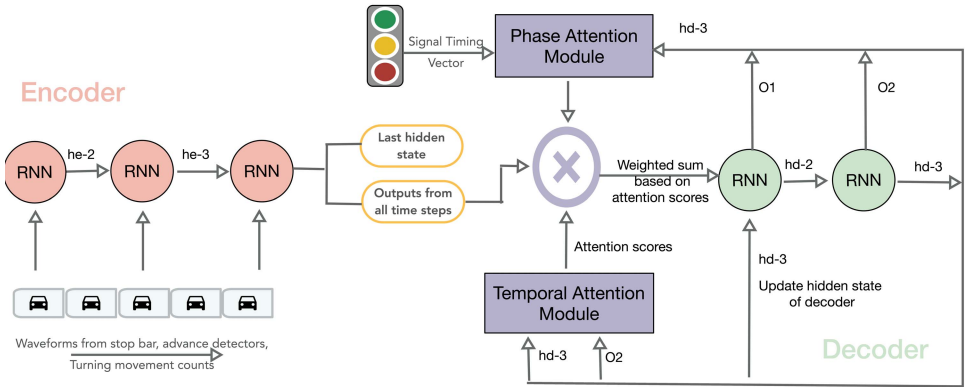

Fig. 4. Dual Attention Encoder Decoder model (DA-ED). The Encoder RNN unrolls over the input waveforms to generate a hidden representation. The decoder is also an RNN that outputs the value of the output waveform at each time step, the input being encoder's output for the current timestep multiplied by the attention scores. Attention modules allow the model to focus on specific parts of the encoder outputs for prediction at current timestep. The temporal attention module is a feed-forward layer using the decoder's output at current timestep and hidden state as inputs. The phase attention module is another feed-forward network that uses the signal timing vector and decoder hidden state as inputs to generate attention scores which represents the importance given to each of the hidden units.

(GRU-based) encoders and decoders for convolutional neural networks (feed-forward networks) to enable easy parallelization. Transformer networks are uniquely suited to natural language processing tasks, modeling long-range sparse dependencies such as correct subject-pronoun matching. For example, consider the sentences "The cat is sleeping on the mat. It has eaten its food." The word "it" in the second sentence depends squarely on "cat" in the first sentence, but not on "mat," which directly precedes it. However, in traffic waveform estimation, it is not the case that an event (such as a sudden inflow spike) occurring several cycles ago will suddenly affect the present cycle without having affected the intervening cycles. The congestion caused by the spike will dissipate over several successive cycles, and its effect will be contained in those successive cycles. Given that traffic state change is a gradual process, RNN architectures are well-suited for such situations, as they rely on the preceding hidden state for predicting the current state.

We compare the performance of the proposed architecture with feed-forward networks in section VI. The feed-forward network we use is a standard fully connected network with four hidden layers with 72, 56, 56, and 72 hidden units. We show that the proposed architecture has a much better prediction accuracy with 70% fewer parameters.

VI. Experimental Results

In this section, we present the experimental results for different models proposed in section III and compare the performance of the proposed architecture with standard feed-forward networks.

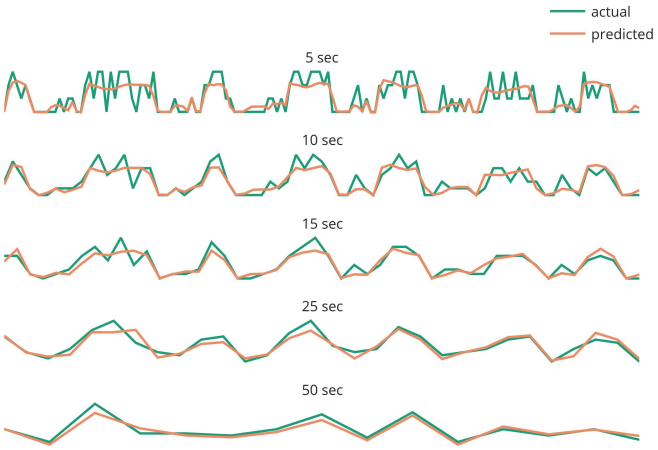

Fig. 5. Plot showing actual vs. predicted exit waveform at different resolutions. The key observation here is that even though the actual vs. predicted waveforms may not exactly match at a 5-s-bucket resolution, if we aggregate them to higher resolution (10, 15, 25, 50 s, etc.), the actual and predicted waveforms almost match.

Even though the models are trained with mean square loss as the loss metric, we analyze the error in terms of *veh per bucket*. As our bucket is 5 s, the error denotes the absolute value of difference between actual and predicted number of vehicles in the 5-s bucket. Figure 5 shows the actual vs. predicted waveform for M_{exit} at different resolutions.

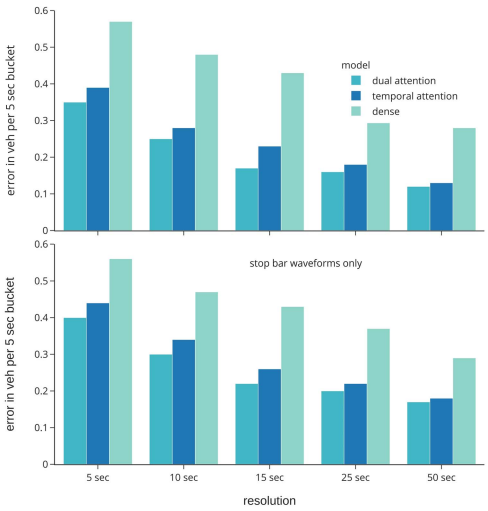

Fig. 6. Plot showing errors for predicting exit waveform using different models (M_{exit}). This plot shows that dual attention encoder decoder model has the best prediction accuracy.

TABLE VI

TABLE COMPARING ERRORS FOR M_{down} MODEL AT DIFFERENT RESOLUTIONS. THE RESULTS SUGGEST THAT USING ROBERTSON MODEL FOLLOWED BY DUAL ATTENTION ENCODER DECODER (DA-ED) MODEL HAS BETTER PREDICTION ACCURACY. DNN: DEEP NEURAL NETWORK

		Error in veh. per bucket				
Model	Inputs	5 sec	10 sec	15 sec	25 sec	50 sec
Robertson	E, d	0.62	0.52	0.46	0.37	0.23
DNN	$S(E, d), d$	0.55	0.46	0.40	0.34	0.25
DNN	$R(E, d), d$	0.55	0.46	0.40	0.34	0.25
DA-ED	$S(E, d), d$	0.40	0.28	0.21	0.15	0.09
DA-ED	$R(E, d), d$	0.38	0.25	0.19	0.13	0.08

The key observation here is that even though the actual vs. predicted waveforms may not exactly match at a 5-s-bucket resolution, if we aggregate them to higher resolution (10, 15, 25, 50 s, etc.), the actual and predicted waveforms almost match. Suppose the actual values in the next two buckets are 3 and 3, but the model predicts them to be 2 and 4; the error is almost 33% per bucket, but at 10-s resolution, the error is 0%. The error at different resolutions indicates that the overall momentum of the system is conserved (the total volume of the predicted waveform is equal to total actual volume). So we also report the error for different resolutions of the waveform even though our prediction is for 5-s resolution.

Figure 6 shows the errors at different resolutions for M_{exit} using different architectures: feed forward, temporal attention, and dual attention. As mentioned earlier, the error is reported in terms of the number of vehicles per bucket. It can be clearly seen that the dual attention encoder decoder (DA-ED) architecture outperforms the baseline feed forward network. Also, we can see that using only waveforms at stop bar detectors and signal timing also gives similar prediction accuracy.

Table VI shows the errors at different resolutions for M_{down} model. This model predicts the modification of the exit waveform as it travels downstream. The table shows the error for predicting the downstream waveform using different models.

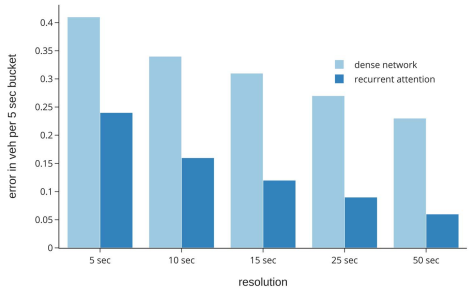

Fig. 7. Plot showing error comparison for temporal attention encoder decoder vs. feed-forward model for inflow reconstruction. This plot shows that encoder decoder with attention model has the best prediction accuracy.

This model employs a two-step strategy to predict the downstream waveforms at a distance x:

1) First, the exit waveform is shifted in time, assuming average vehicle speeds to cover the distance x. To derive the exit waveform is used to impute the waveform at distance d based on platoon dispersion. We tried out two different strategies (1) Shift the exit waveform based on distance (d) - $S(E, d)$. This assumes that the waveform seen at the exit detector is largely unperturbed as it progresses downstream. (2) Use Robertson platoon dispersion model - $R(E, d)$.
2) Next, this shifted waveform is fed to a neural network model that modifies the waveform to incorporate non-linearities due to variable velocities, different driver behavior and signaling plan on the next intersections

It is important to note that the first step of shifting the waveform effectively captures the distance information between the two intersections. This allows the neural network model in the second step to focus on modifying the shifted waveform; thus, it is independent of the distance between the two intersections. Our experimental results suggest that using Robertson model followed by Dual Attention Encoder Decoder (DA-ED) model has better prediction accuracy

This model can be used to predict an inflow waveform for the downstream intersection. This inflow waveform is not affected by the signal timing state of either intersection. We can use the inflow waveform along with downstream signal timing information to reconstruct the waveforms at advance and stop bar detectors (M_{sa} model).

Figure 7 shows the errors at different resolutions for M_{in} comparing the dense network with temporal attention encoder decoder architecture. It can be seen that we could reconstruct the inflow waveform with good accuracy using stop bar and advance detector waveforms. At 50-s aggregation, the error in vehicles per bucket for the recurrent model is 0.06, equivalent to 0.6 vehicles (over- or under-counting by 0.6 vehicle for a 50-s period), whereas with the dense network, it is 0.27 or 2.7 vehicles.

Figure 9 shows the attention scores generated for each decoding step. The attention scores suggests that at each

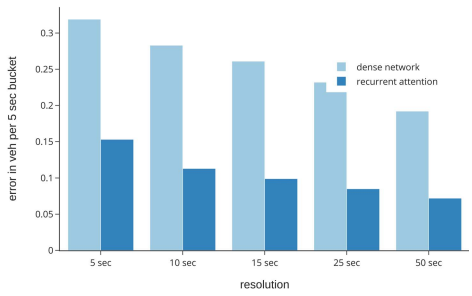

Fig. 8. Plot showing error comparison for recurrent attention vs. feed-forward model for stop bar and advance waveform reconstruction. This plot shows that encoder decoder with attention model has the best prediction accuracy.

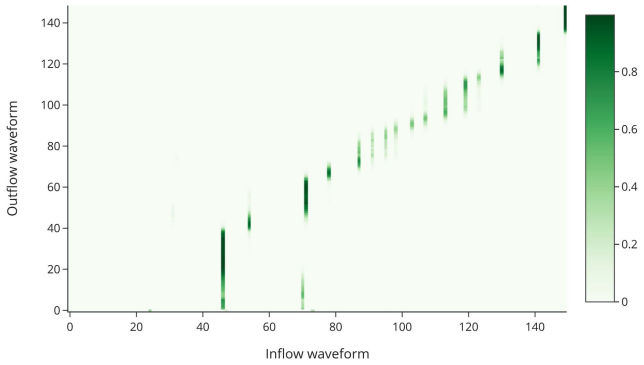

Fig. 9. Plot showing heat map of temporal attention scores. The X-axis indicates the time buckets for the inflow waveform and Y-axis indicates time buckets for the predicted outflow waveform. This plot suggests that attention module is helping the network to focus on relevant temporal information, around the time for which the prediction is made.

decoding step, more importance is given to hidden states of the encoder corresponding to that particular time step. For example, for predicting output at step 80, more importance is given to encoder hidden states corresponding to time steps 70-80. This suggests that the attention module is helping the model to focus on relevant temporal information at each step.

Also, It is worth noting that these methods are three to four orders of magnitudes faster than using microscopic simulations. For simulating input output patterns using SUMO, for a single intersection took 9 sec per simulation on a 32 core machine. While the trained neural network models when used in inference mode are able to generate output in less than a millisecond for a batch size of 5000. This suggests that neural network based model is atleast 4 orders of magnitudes faster compared to traditional simulation approaches.

VII. Related Work

Machine learning techniques, including deep neural networks, have been successfully applied to traffic data for traffic state prediction [19] for short-term (5-30 minutes), medium-term (30-60 minutes) and long-term (1+ hour) time windows.

Most of the previous work is in volume prediction or predicting flows at downstream intersections. We outline this work below:

1) Predicting volumes at cycle level: A number of techniques have been used for predicting volumes at cycle-length resolution (generally 2 minutes). This includes ensembled kernelized matrix completion [10], shock-wave analysis, and Bayesian networks [17].
2) Predicting volumes at 5-minute intervals: Techniques that have been used include ARIMA [20], deep learning with non-parametric regression [2], multisegments (with recurrent and convolutional layers), deep neural network [18], graph embedding coupled with a generative adversarial network [25], and a combination of linear genetic programming (LGP), multilayer perceptron (MLP), and fuzzy logic [26].

However, these works primarily focus on predicting volumes for the same loop detector or location at a future point in time. They do not attempt to model the outflow waveform exiting a signalized intersection or its modification downstream. Somewhat relevant to our work are Ehlers [7] and Wright *et al.* [22], which use geometric deep learning architectures, and Sun and Zhang [16], which uses a linear model to predict flow at the downstream intersection, given upstream intersection detector flows.

Platoon dispersion models have been proposed in the literature to model flow rates of a platoon as it traverses through a corridor. Lightlhill and Witham modelled platoon dispersion using kinematic wave theory [11]. Platoon dispersion models as analogous to continum fliud based on shock wave theory is proposed by Pacey [13]. The model that is being widely used is based on Robertson platoon dispersion model [15]. Some recent studies also proposed variations of Robertson's model to account for heterogeneous traffic flow condition [8], [23], [24]. These dispersion models however are not targeted towards determining the impact of signal timing or traffic entering from the side streets. The flow rates at a point are similar to waveforms described above. By using a composition of deep neural networks and Robertson model, our novel approach can incorporate the impact of signal timing of the next intersection on the platoon dispersion (cf. Section III).

VIII. Conclusion and Future Work

We described our work on modeling waveforms at signalized intersections at subcycle resolutions. We generated a large dataset based on real-world traffic flows and signal timing plans. We then trained deep learning models and evaluated them. We arrive at the following conclusions:

1) We are able to decompose arterial traffic flow dynamics by considering local interactions (M_{exit}) and coupling interactions (M_{down}), and show that they can effectively approximate the dynamics of a pair of intersections. An important advantage of using (M_{down}) is that the time-shifted input to the trained neural network model is independent of the distance between the two intersections.
2) We are able to effectively reconstruct the unperturbed incoming inflow waveform to an intersection from the stop bar and advance detectors and signal timing information waveforms (M_{in}). We verify that this

reconstruction is accurate enough to reasonably estimate the observed stop bar and advance detector waveforms (M_{sa}). An important point to note is that the reconstructed inflow waveform is largely independent of the signal timing plan of the approaching intersection because it is still a significant distance away.

3) We see the accuracy measures of our predictions and reconstructions improve with larger aggregation times, from 5 seconds to 60 seconds.

Going forward, we hope to expand on this work as follows:

1) We aim to use our trained models (M_{in}) to reconstruct inflow patterns based on real-world recorded traffic data along arterials.
2) With reconstructed inflow data, we intend to use our trained models (M_{exit}/M_{down}/M_{sa}) to predict exit waveforms and their progression downstream, given a candidate signal plan for the two intersections.
3) Also, we intend to modify these models to predict level of service (LOS) measures based on the above techniques.
4) With exit waveform and combined LOS measures estimated, we will repeat the above process for the next pair of intersections along the arterial.
5) We then evaluate the efficiency of multiple candidate signal plans in parallel, using a Monte Carlo tree search (MCTS) [3] algorithm to find the best signal plans for the corridor.8
6) By identifying key corridors within the urban traffic grid, we hope to perform city-scale grid optimization.

ACKNOWLEDGMENT

The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of NSF.

The authors also acknowledge University of Florida Research Computing for providing computational resources and support that have contributed to the research results reported in this publication.

REFERENCES

[1] *US DOT Signal Design*, Federal Highway Admin., 2020. [Online]. Available: https://ops.fhwa.dot.gov/publications/fhwahop08024/chapter4.htm
[2] M. Arif, G. Wang, and S. Chen, "Deep learning with non-parametric regression model for traffic flow prediction," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervas. Intell. Comput., 4th Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Aug. 2018, pp. 681–688.
[3] C. B. Browne *et al.*, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.
[4] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," 2019, *arXiv:1904.02874*. [Online]. Available: http://arxiv.org/abs/1904.02874
[5] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation," 2016, *arXiv:1610.02583*. [Online]. Available: http://arxiv.org/abs/1610.02583
[6] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078
[7] S. F. G. Ehlers, "Traffic queue length and pressure estimation for road networks with geometric deep learning algorithms," 2019, *arXiv:1905.03889*. [Online]. Available: http://arxiv.org/abs/1905.03889
[8] Y. Jiang, Z. Yao, X. Luo, W. Wu, X. Ding, and A. Khattak, "Heterogeneous platoon flow dispersion model based on truncated mixed simplified phase-type distribution of travel speed," *J. Adv. Transp.*, vol. 50, no. 8, pp. 2160–2173, 2016.
[9] C. R. Lattimer, "Automated traffic signal performance measures (ATSPMs)," FHWA, Washington, DC, USA, Tech. Rep. FWHA-HOP-20-002, 2020.
[10] W. Li, C. Yang, and S. Eddin Jabari, "Short-term traffic forecasting using high-resolution traffic data," 2020, *arXiv:2006.12292*. [Online]. Available: http://arxiv.org/abs/2006.12292
[11] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London A. Math. Phys. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.
[12] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
[13] G. M. Pacey, "The progress of a bunch of vehicles released from a traffic signal," Road Res. Lab., Growthorne, U.K., Tech. Rep. Rn/2665/GMP, 1956.
[14] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
[15] D. I. Robertson, "TRANSYT: A traffic network study tool," Road Res. Lab., Crowthorne, U.K., RRL Rep. LR 253, 1969.
[16] J. Sun and L. Zhang, "Vehicle actuation based short-term traffic flow prediction model for signalized intersections," *J. Central South Univ.*, vol. 19, no. 1, pp. 287–298, 2012.
[17] S. Wang, W. Huang, and H. K. Lo, "Traffic parameters estimation for signalized intersections based on combined shockwave analysis and Bayesian network," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 22–37, Jul. 2019.
[18] Y. Wang, S. Xu, and D. Feng, "A new method for short-term traffic flow prediction based on multi-segments features," in *Proc. 12th Int. Conf. Mach. Learn. Comput.*, Feb. 2020, pp. 34–38.
[19] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, Feb. 2019.
[20] Y. Wang, L. Zhao, S. Li, X. Wen, and Y. Xiong, "Short term traffic flow prediction of urban road using time varying filtering based empirical mode decomposition," *Appl. Sci.*, vol. 10, no. 6, p. 2038, Mar. 2020.
[21] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.
[22] M. A. Wright, S. F. G. Ehlers, and R. Horowitz, "Neural-attention-based deep learning architectures for modeling traffic dynamics on lane graphs," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3898–3905.
[23] W. Wu, W. Jin, and L. Shen, "Mixed platoon flow dispersion model based on speed-truncated Gaussian mixture distribution," *J. Appl. Math.*, vol. 2013, Jun. 2013, Art. no. 480965.
[24] W. Wu, L. Shen, W. Jin, and R. Liu, "Density-based mixed platoon dispersion modelling with truncated mixed Gaussian distribution of speed," *Transportmetrica B, Transp. Dyn.*, vol. 3, no. 2, pp. 114–130, 2015.
[25] D. Xu, C. Wei, P. Peng, Q. Xuan, and H. Guo, "GE-GAN: A novel deep learning framework for road traffic state estimation," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102635.
[26] S. A. Zargari, S. Z. Siabil, A. H. Alavi, and A. H. Gandomi, "A computational intelligence-based approach for short-term traffic flow prediction," *Expert Syst.*, vol. 29, no. 2, pp. 124–142, 2012.

Yashaswi Karnati received the bachelor's degree in electrical engineering from the Indian Institute of Technology, Dhanbad, and the master's degree in computer science from the University of Florida, Gainesville, FL, USA, where he is currently pursuing the Ph.D. degree with the Department of Computer and Information Science and Engineering. His current research interests include developing data driven machine learning algorithms for practical applications in intelligent transportation, health care, and climate science.

Rahul Sengupta is currently pursuing the Ph.D. degree with the Computer and Information Science Department, University of Florida, Gainesville, USA. His research interest includes applying machine learning models to sequential and time-series data, especially in the field of transportation engineering.

Anand Rangarajan (Member, IEEE) is currently a Professor with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. His research interests include computer vision, machine learning, medical and hyperspectral imaging, and the science of consciousness.

Sanjay Ranka (Fellow, IEEE) is currently a Distinguished Professor with the Department of Computer Information Science and Engineering, University of Florida. His research is currently funded by NIH, NSF, USDOT, DOE, and FDOT. From 1999 to 2002, he was the Chief Technology Officer and the Co-Founder of Paramark, Sunnyvale, CA, USA, where he conceptualized and developed a machine learning based real-time optimization service called PILOT for optimizing marketing and advertising campaigns. His current research interests include developing algorithms and software using machine learning, the Internet of Things, GPU computing and cloud computing for solving applications in transportation, and health care. He is a fellow of AAAS and Asia–Pacific Artificial Intelligence Association (AIAA) and a past member of IFIP Committee on System Modeling and Optimization. He was awarded the 2020 Research Impact Award from IEEE Technical Committee on Cloud Computing. He was recognized by VentureWire/Technologic Partners as a Top 100 Internet Technology Company in 2001 and 2002 and was acquired in 2002.