WHAT DRIVES PREDICTIONS IN TRAFFIC FORECASTING?
DATA VALUATION FOR DEEP LEARNING ON TIME SERIES

SENIOR THESIS

Dylan V. Chou
Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213
dvchou@andrew.cmu.edu

Peter Freeman
Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213
pfreeman@cmu.edu

June 11, 2022

ABSTRACT

Research into forecasting highway traffic has explored advanced deep learning frameworks. However, deep learning model predictions lack explainability. The severe consequences of inaccurate traffic predictions on commute time and highway accidents makes explainability important for road traffic models. Explainable AI paradigms such as feature importance or individual data observation valuation have been applied to deep traffic models, but less work has measured importance over clusters of cyclical data. Our work will be part of a case study on the PEMS04 and PEMS08 datasets as well as the Highways England 2013 dataset to fill the gaps in traffic neural network explainability. To this end, we provide a data valuation framework over specified ranges of traffic data. We compare MAE prediction error distributions after substituting randomly sampled regions similar to the specified data portion, which allows us to compare importances of different regions. The estimation method is evaluated on the basis of stability, or that the averages between two MAE distributions from repeated runs are consistent (e.g. the relative ordering between regions is always maintained).

Keywords Data Valuation · Deep Learning · Traffic Forecasting

1 Introduction

1.1 Deep Learning in Traffic Forecasting

There has been interest in using deep neural network models to predict traffic information, such as future traffic speeds, and help evaluate traffic efficiency. Incorporation of spatial attributes along with temporal information in graph-based convolutional neural networks became prevalent since [8] when Yu and others presented a neural network framework combining spatial graph representations of roadway traffic connectivity and temporal convolution layers with a specified K_t kernel width to capture adjacent traffic observations in the time series. Additional spatial information collected through sensor stations in the Caltrans Performance Measurement System (PeMS). [31] built on top of the convolutional recurrent neural network paradigm by modeling the stochastic properties of traffic as a diffusion process that's represented by a random walk. Chen and others [4] also implemented a convolutional neural network and used multiple convolutions to model greater time dependencies between traffic observations, and expanded on understanding parameter impact on prediction errors in their deep convolutional neural network. Their network model captures local temporal patterns and global congestion trends of the data, measuring the impact of the number of time slots and number of days available in the training and testing datasets on model forecast errors. Geng and others [12] modeled non-Euclidean correlations between spatially distant regions for ride-hailing services using multi-graph convolutions. They defined correlation between regions as their connectivity, similarity between their surrounding regions, and their proximity. Regarding diffusion convolutional neural networks, [9] couples temporal attributes with spatial dependencies, which are captured in graph diffusion convolution, or a convolution block constructed from a graphical random walk with a restart probability and a trainable transition matrix. The base paradigm of a convolutional neural network for spatio-temporal data has been used to model stochastic properties of traffic, connectivity between regions in a graph of directed traffic flows, and correlation between those traffic regions.

Frameworks for general traffic forecasting have also been presented as [10] defined a concrete deep learning framework to follow. Yu et al proposed a mixture deep long short term memory (LSTM) network that models both normal traffic and less frequent road accidents. They forecast traffic flow during peak-hour periods with greater variation in truck traffic and vehicle movement patterns. Frameworks have also applied traffic forecasting to solve problems of general travel time prediction. [14] provided a framework to predict travel times along road routes using a spatio-temporal component and an attribute component where sampled vehicle trajectories are sampled and passed through a geo-convolutional layer. Attributes such as driver habits, weather, time and distance traveled along a route are embedded and passed into an attention layer that learns weights for local paths. Ultimately, the model learns a multi-task problem where it learns sampled local paths along with the overall path. [11] implemented a similar framework, but applied the model to accurate public bus transportation.

The application of deep learning frameworks to time-series forecasting has been met with interpretable models and explainability techniques. [13] forecasted multi-step travel times based on roadway traffic predictions through matched spatiotemporal traffic patterns in speed contour plots. Gray-level Co-occurrence matrices are used to identify pairwise speeds that co-occur at various distances along an urban expressway, which can identify similar traffic patterns during forecasts. To forecast traffic patterns, Chen and others [21] draw on an analogy between a single neural network layer and an iteration in the process of reducing noise in data, which allows for greater explainability of their model with the use of an analogy. Other papers were able to make intuitive conclusions about what data their models performed best on as well as the explainability of top performing models. Yang et al [5] trained a recursive neural network (RNN) among other machine learning models on the traffic data provided by the Minnesota Department of Transportation [6], where they found that prediction accuracy improves during the peak hours of traffic in the data. In Yang et al's model and similar deep learning techniques [2, 7], they are able to make intuitive conclusions that predictions further into the future result in larger errors and some models perform better than others based on comparisons of performance metrics. Baric et al [22] benchmarked the explainability of different models on about 10 datasets using confidence based on standard deviation of model parameter estimates. They focus on comparisons of explainability in different models. Ismail and colleagues [23] studied saliency methods in time series by first evaluating importance of time steps before computing the feature importance at a time step. Their analysis remedies issues with feature importance of multiple features over many time steps in a time series. Cho and colleagues [26] visualize temporally activated patterns by identifying network nodes with high activation values in a channel. Visualizations show portions of the traffic time series colored based on convolutional network channels that are activated the most when data during those portions are passed into the model. In turn, the visualization of most activated nodes can explain which portions of the data, such as inflection points, are most valuable during a model's learning process.

As described in [15], new deep learning architectures for short-term traffic prediction add to the already massive pool of papers that do not address the caveats of their approaches, such as the black-box nature of their models or computation costs to train the model. One of the challenges brought up in the paper is the lack of explainable AI perspectives on traffic forecasting. Some papers that do address explainability study ways that upstream and downstream traffic data impact model predictions [16], the features learned by the first layer in an autoencoder [18], and the importance of specific traffic flow time steps on forecasts [17]. More recently, research is trending towards understandable systems [42, 43], where decision models are equipped with visualizations of Local Interpretable Model-Agnostic Explanations (LIME) and SHap Additive exPlanations (SHAP) values.

1.2 Feature and Data Valuation on Time Series

Feature and data valuation approaches can offer explainability in deep traffic forecasting models. Popular approaches for data valuation on time series include shapley values and counterfactual explanations. Shapley values originated from cooperative game theory where a group of players needs to distribute the gain attained through collective cooperation. Some players may have contributed more than others, which is quantified in the importance of their contribution using shapley values. Shapley values are theoretically quite costly as the shapley value of a feature value is $\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!}(val(S \cup \{x_j\}) - val(S))$, which requires the sum of all prediction differences between the inclusion of feature x_j and its exclusion over all permutations of subsets of features in a model; the computation is in exponential-time and NP-complete. To circumvent the computation costs, Strumbelj and others [24] approximated shapley values using Monte Carlo simulations to construct new data examples where an average of m randomly sampled differences is computed: $\frac{1}{M}\sum_{m=1}^{M} \widehat{f}(x_{+j}^{m}) - \widehat{f}(x_{-j}^{m})$. $\widehat{f}(x_{+j}^{m})$ is the prediction when a random number of feature values in x are replaced by a randomly sampled vector z except for the j^{th} feature value. $\widehat{f}(x_{-j}^{m})$ takes a randomly sampled vector z that replaces some random number of features in x, but also replaces feature value x_j with z_j. With applications to air quality prediction based on nitrogen dioxide forecasts [32], sales deals prediction based on sales related activities [19], and general time series forecasts for trend cycles [20], shapley values can quantify and compare specific atmospheric factors for air quality prediction, types of tools used during the sales process and frequency of data logging for sales deals prediction, and lagged time series variables for trend-cyclical data, respectively. However, any domain-specific application of shapley values is limited by features with interventional effects and underlying causal dependencies between features [34]. Causal shapley values were then introduced by Heskes and others [37], which measures the total effect of a shapley value to be the sum of the direct effect and the indirect effect:

$$\begin{aligned}\phi_i(\pi) &= \mathbb{E}[f(\mathbf{X}_{\bar{S}}, \mathbf{x}_{\underline{S} \cup i})|do(\mathbf{X}_{\underline{S} \cup i} = \mathbf{x}_{\underline{S} \cup i})] - \mathbb{E}[f(\mathbf{X}_{\bar{S} \cup i}, \mathbf{x}_{\underline{S}})|do(\mathbf{X}_{\underline{S}} = \mathbf{x}_{\underline{S}})] \\ &= \mathbb{E}[f(\mathbf{X}_{\bar{S}}, \mathbf{x}_{\underline{S} \cup i})|do(\mathbf{X}_{\underline{S}} = \mathbf{x}_{\underline{S}})] - \mathbb{E}[f(\mathbf{X}_{\bar{S} \cup i}, \mathbf{x}_{\underline{S}})|do(\mathbf{X}_{\underline{S}} = \mathbf{x}_{\underline{S}})] + \\ &\quad \mathbb{E}[f(\mathbf{X}_{\bar{S}}, \mathbf{x}_{\underline{S} \cup i})|do(\mathbf{X}_{\underline{S} \cup i} = \mathbf{x}_{\underline{S} \cup i})] - \mathbb{E}[f(\mathbf{X}_{\bar{S}}, \mathbf{x}_{\underline{S} \cup i})|do(\mathbf{X}_{\underline{S}} = \mathbf{x}_{\underline{S}})]\end{aligned}$$

The direct effect is the expected change in prediction when feature X_i is replaced with x_i without changes to the other features while indirect effect accounts for distribution changes after the replacement of feature X_i. Provided a causal directed acyclic graph (DAG) of features, one can use the partial ordering of features in a causal graph and sample accordingly from empirical or conditional variable distributions.

Counterfactual explanations for time series model predictions are used to define importance of particular time series observations by replacing an observed value with a generated quantity. [25] used a forward feed counterfactual (FFC) to replace i^{th} time series observations with a generated one. The effect of this replacement is defined as the importance of the i^{th} observation. [27] proposed an instance-based counterfactual explanation technique to find a series of changes to a time series T_q predicted to be in class c that the system would predict to be in class $c^{'}$. Ates and others [28] presented a greedy algorithm to provide counterfactual explanations for multivariate time series, which offers a minimum number of time series substitutions that could maximize the probability of the time series being classified as class c. The counterfactual explanations were not generated, but instead used an x_{dist} would be chosen from the time series training set. However, Keane and others [29] presented three main deficiencies of current counterfactual methods: the lack of user studies or proven preference by end-users, absence of proxies estimating the "psychological" distance of counterfactual explanations, and the open-ended nature in defining how few features a counterfactual explanation should have or "plausibility" of the explanations.

1.3 Our Contribution

Past papers on traffic forecasting are saturated with spatio-temporal models, general deep learning frameworks, comparisons in the explainability of different models, and interpretable traffic models. Data valuation in time series forecasting has primarily concerned minimal changes in variable features required to change prediction outcomes or average changes in prediction error after randomly sampling feature values. Existing data valuation methods, such as in [17], only measure the impact of individual traffic observations on model predictions and have gone in the direction of accounting for underlying causal relationships when measuring the importance of multiple time series variables. However, for the univariate traffic forecasting problem, there is a dearth of explainability techniques that evaluate data importance on model predictions over contiguous time series observations, which is a gap that we fill. While the importance of traffic flow data has been studied in [17], we undertake a methodical and interpretable approach by considering contributions of regions of historical traffic defined through automated top-k searches of the steepest and least steep traffic flow changes and manual labeling of start of the day and end of the day traffic. We examine three cyclical time series datasets – PEMS04, PEMS08, and Highways England A414 2013 datasets – that aim to forecast traffic flow. Our explainable AI framework estimates the impact of traffic observations during the start or end of the day on Du et al's LSTM based encoder-decoder neural network model, and allow practitioners to probe forecast contributions over different portions of cyclical data. Our work can extend current data valuation methods in traffic forecasting as we may flexibly test hypotheses on periodic data of interest that traffic operators can understand. Our probing framework may be incorporated in future tests to ascertain whether its addition will increase the accuracy of traffic operators in predicting traffic model behavior. The dearth of evaluation metrics predictive of model explanation efficacy motivates the creation of probing tools. These tools may boost efficacy when tested with the appropriate metrics on users in specific fields such as traffic monitoring [39].

2 Problem Domain

2.1 Objective

The time series data D in the A414 2013 dataset is composed of traffic flow, speed, and journeytime observations while the PeMS datasets contain average speed (average speeds of vehicles every 5 minutes), occupancy (fraction of a 5 minute interval where a vehicle is over a detector), and traffic flow (number of vehicles passing over detector in

5 minute interval). The 5 minute interval averages are aggregated from 30-second estimates. We want to answer the question of whether there are parts of the periodic data in the highway dataset that have a substantial contribution to Du et al's [2] model predictions. Namely, we consider the following two questions:

- Are highway traffic observations towards the start or end of a day more important in Du et al's model predictions than data in the middle of the day?
- How does the impact of traffic observations with larger changes in traffic flow on Du's model predictions compare to those with smaller changes?

Our objective is to identify regions of highway traffic data that drive model predictions, which can be helpful in adopting holistic, local model explainability, as defined in a 2018 paper by Lipton [3]. The two aforementioned questions serve to guide our analyses. We hypothesize that data with drastic changes, or large negative or positive consecutive differences between observations, are more important in Du et al's and Guo et al's model predictions than traffic that changes less over time. We also claim that the impact of data within a cycle or at the border of two cycles on Du's model predictions depends on the data's rate of change in the time series. In general, forecasting traffic information is important in helping traffic operators design strategies to mitigate traffic congestions depending on the traffic flow data. Adding explainability to an existing deep learning model can inform traffic operators about real-time decision-making alongside Intelligent Transportation Systems (ITS).

2.2 Datasets

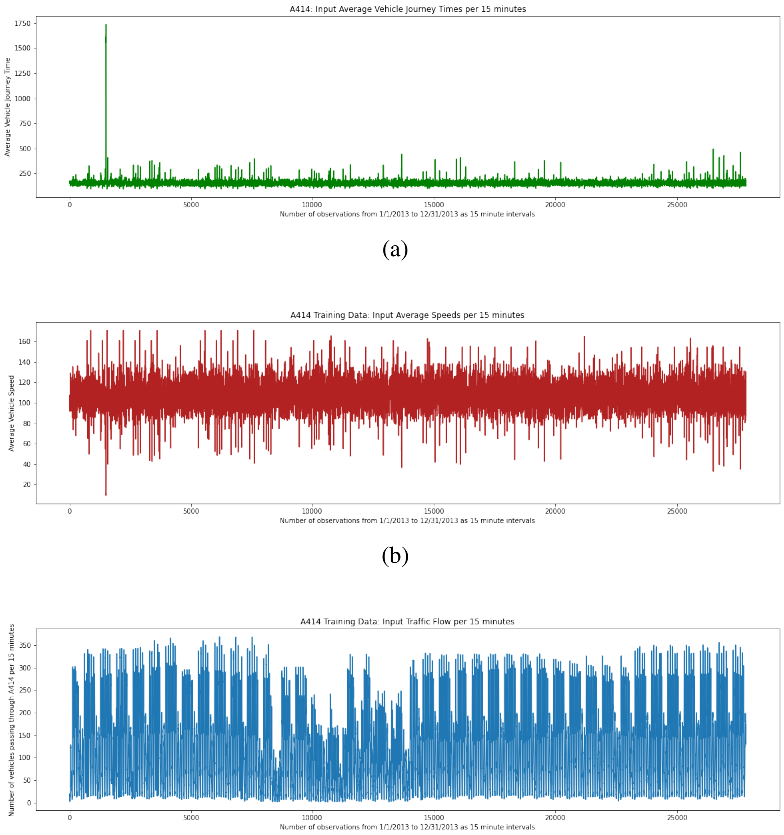
Overview We test our probing tool on the three datasets: Highways England 2013 A414 traffic data, California Department of Transportation Agency Performance Measurement System (PeMS) 04, and PeMS08. The Highways England dataset [41] provides information on average speeds, flows and journey times [1]. There are two types of data in the dataset: Motorway Incident Detection and Automatic Signalling (MIDAS) and Traffic Monitoring Units (TMU) or inductive loops. MIDAS traffic sites use induction loops, and some radar technology, 500 meters apart that detect incidents on the road whereas TMU solely uses induction loops. The PeMS datasets were collected by 3900 sensors throughout the metropolitan areas of California, where PeMS04 is traffic data recorded in the San Francisco Bay area by 3848 sensors on 29 roads from Jan. 1, 2018 to Feb. 28, 2018 and PeMS08 is traffic data in the San Bernardino area detected by 1979 sensors on 8 roads from 7/1/2016 to 8/31/2016. Table 1 summarizes the aforementioned datasets in further detail.

Dataset Name	# Observations	Recording Device(s)	Data Types	Duration
PeMS04	16,992 (307 detectors)	Induction loops	Traffic Flow, Speed, Journey Time	5 minutes
PeMS08	17,856 (170 detectors)	Induction loops	Traffic Flow, Speed, Journey Time	5 minutes
Highways England	37,564	Loops and Radar	Traffic Flow, Speed, Journey Time	15 minutes

Table 1: An overview of the periodic traffic datasets.

Highways England A414 dataset and Du et al With respect to the Highways England Dataset [1], our problem is the following: we are given multivariate data D as a time series of traffic flow, traffic speed, and traffic journeytime. For instance, traffic flow has observations $flow_1, flow_2, ...flow_t$ denoting the observed traffic flow at the specified time step t per 15 minutes. The three variables are shown in Figure 1, where it's evident that traffic flow has concrete periodicity that will allow us to identify sufficient similar regions to resample from in our estimation procedure. In Figure 2, the seasonal component of traffic flow is most evidently seasonal with a relatively concrete period. This allows for sufficiently many similar data portions to be resampled during perturbations. In turn, we focus on analyzing the importance of traffic flow. The experimental dataset used by Du et al [2] contains 37,564 traffic flow data points from 1/1/2013 to 12/31/2013 and 2/1/2014 to 2/28/2014 in 15 minute intervals for the major road "Site A414" in England. The data is split 80%/20% into training and validation data, respectively, over the 2013 data for Du et al's model and the 2014 data is used for testing. The A414 dataset is a dataframe that contains the feature columns of "AverageJT" (average vehicle journey time along highway A414 between A405 and junction M1 J7), "AverageSpeed" (in kilometers per hour, the average speed measured at the start and end of the A414 road), and Flow (the number of vehicles expected to be detected by the National Traffic Information Service (NTIS) link to road A414 in the span of 15 minutes). The portion of highway A414 is illustrated in the map in Figure 4.

PeMS datasets and Guo et al The PeMS datasets provide the aforementioned multivariate data, but are stored in a 3d `numpy array (# observations, # detectors, # variables recorded, # points ahead of current)` rather than a dataframe as in the Highways England Dataset. The PeMS04 datasets have 307 detectors and have



(a)

(b)

(c)

Figure 1: (a) Plot of Average Journey Time Training Data (1/1/2013-12/31/2013) (b) Plot of Average Speed Training Data (1/1/2013-12/31/2013) (c) Plot of Traffic Flow Training Data (1/1/2013-12/31/2013)

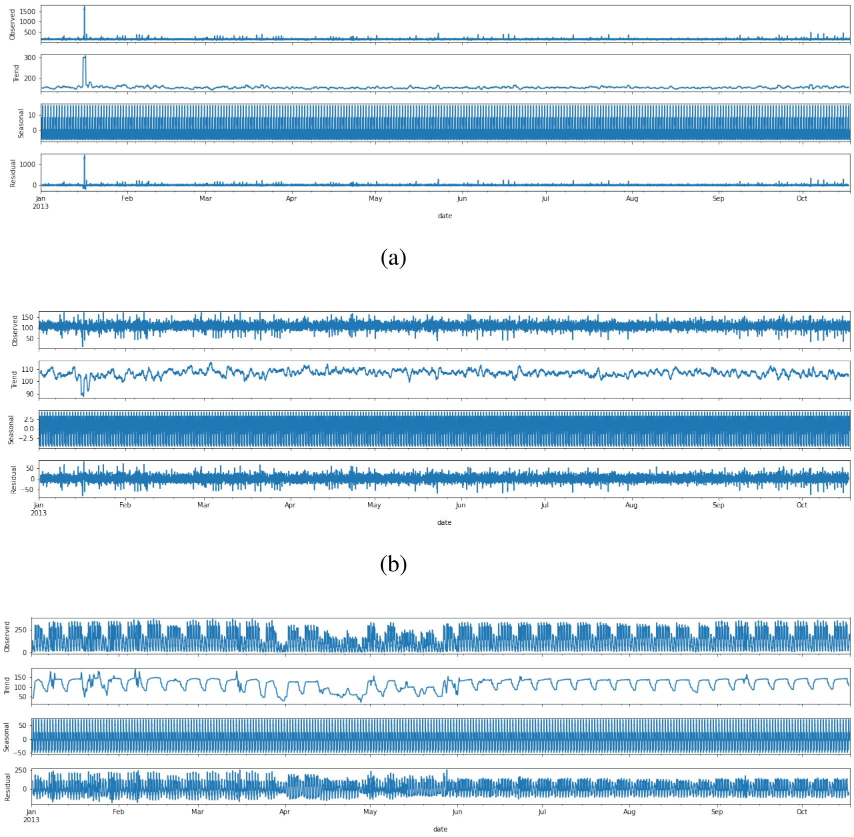
up to 11 points following each observation i. The three dimensions of the input training data and single traffic flow training output are illustrated in Figure 3. During our perturbations, we focus on importances of traffic flow data. The PeMS08 dataset has the same number of data points following each observation, but instead has 170 detector sites recording data on flow, speed, and occupancy, or how long vehicle spends over a detector.

3 Methodology and Analysis

In this section, the problem formulation is presented on how the importance of contiguous traffic data "R" is estimated, followed by the procedure: first identifying the appropriate dataset that the user should call perturbations on, then running the perturbations that substitute out "R" with an independent block of data most similar to R. The resulting changes in mean average error (MAE) are averaged, whose distributions are used during evaluation to ensure the estimates are stable; we observe roughly consistent conclusions across repeated runs of the estimation method.

3.1 Problem Formulation

Given a sequence of n time series observations to perturb $D = [x_1, ..., x_n]$ along with a region of interest (ROI) over a range of observations $R = D[i : j]$, we resample R with the top $k = 15$ most similar regions to R: $[R_1, ..., R_{15}]$. In turn, we would either substitute out the region that follows the other to avoid look-ahead bias, which occurs when the model is exposed to batch data that's has not been available yet. The paradigm used to classify the importance of features most similar to R is illustrated in Figure 5. The region of importance is effectively substituted out with data from another region, presumed to be virtually the same as R. Effectively, we view the portions virtually the same as R to be a "concept" or feature along the traffic flow time series in the time series D, which we resample from to estimate its importance.



(a)

(b)

(c)

Figure 2: (a) Seasonality Plot for Average Journey Time (1/1/2013-12/31/2013) (b) Seasonality Plot for Average Speed (c) Seasonality Plot for Traffic Flow

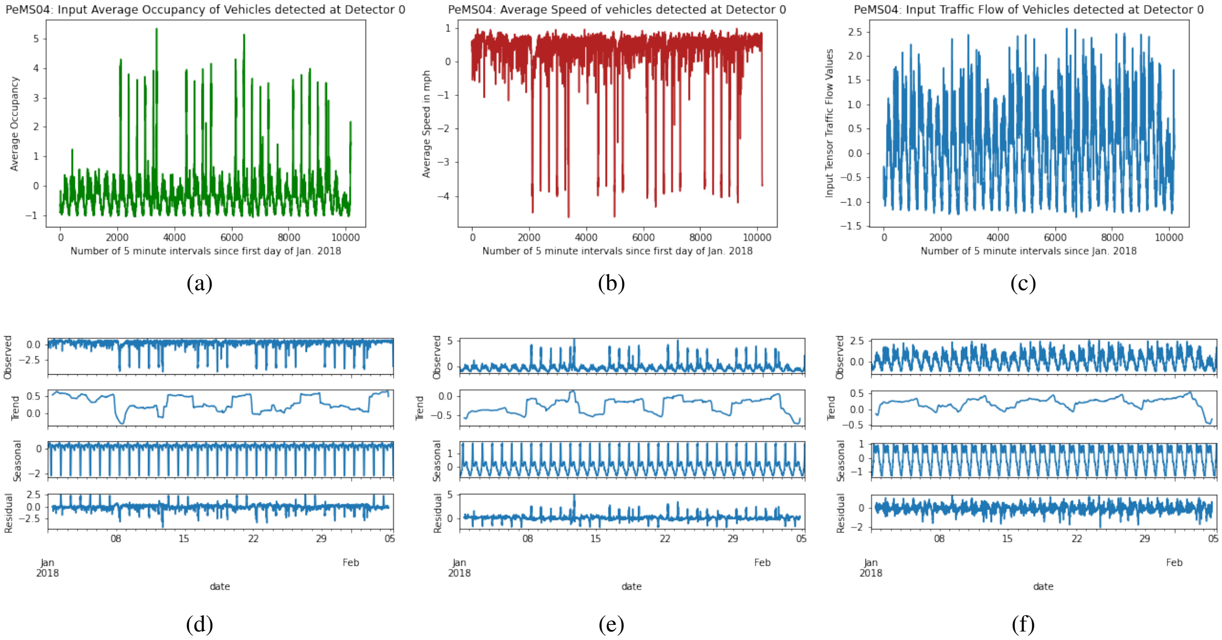
3.2 Defining the Regions

In response to the two questions under the *Objective* section, the first and last four hours per day within the input data of the A414 and PeMS datasets are regions of interest that we want to extract importances from. The top k regions of traffic flow that have the largest and smallest changes are used to compare the impact of rate of change in traffic flow on model predictions. Our experiments would run find_optim_change, as shown in Algorithm 1, that takes in a minimum threshold of consecutive points that need to monotonically increasing or decreasing. In Algorithm 1, we scan over the time series with a fixed block size s and identify regions of low and high rates of change in traffic flow data.

The top k regions satisfy the threshold constraint and have either the least or greatest euclidean norm (i.e. the former being the regions with lowest rate of change and latter being highest). k is maximized so we may end up with the maximum number of regions with the highest and lowest rates of change. After the k regions have been chosen, some may be overlapping, which would not be independent with each other. In turn, we run a dynamic programming algorithm that maintains a table `OPT` of time series intervals and maximizes the number of non-overlapping intervals we choose among the k regions, as shown in `max_num_intervals` in Algorithm 2.

3.3 Perturbation-Data Compatibility

Our probing tool is intended to be a wrapper for models trained on periodic traffic data, which also requires perturbations to be applied to the correct data. Based on Strumbelj's Monte-Carlo method, we perturb a given testing sample that we want to predict the output for. The testing sample in the case of the A414 and PeMS datasets is the input tensor for testing data, which we would observe the predicted traffic flow from. In turn, for both the A414 and PeMS datasets, the test inputs `pems08_data['test_x'][:,detector,0,0]` and `pems04_data['test_x'][:,detector,0,0]` for traffic flow data recorded at detector “detector", and `A414df["Flow"]` are perturbed. We hold the average occupancy



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: (a) PeMS04: Plot of Average Occupancy Input Tensor Values at Detector 0 (b) PeMS04: Plot of Average Speed Input Tensor Values at Detector 0 (c) PeMS04: Plot of Traffic Flow Input Tensor Values at Detector 0 (d) Seasonality Decomposition of Average Occupancy. Third graph from the top is the seasonal trend. (e) Seasonality Decomposition for Average Vehicle Speed (f) Seasonality Decomposition of Traffic Flow

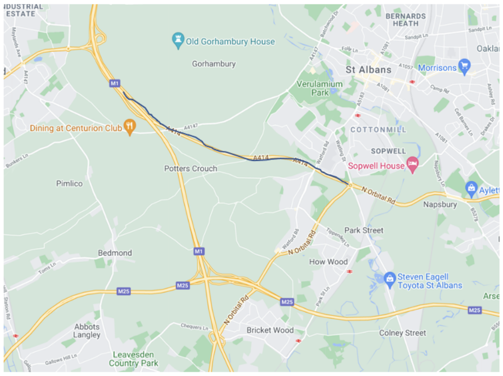


Figure 4: Region of Highway A414 between junction M1-J7 and A405 where average speed and journey time as well as traffic flow are being measured.

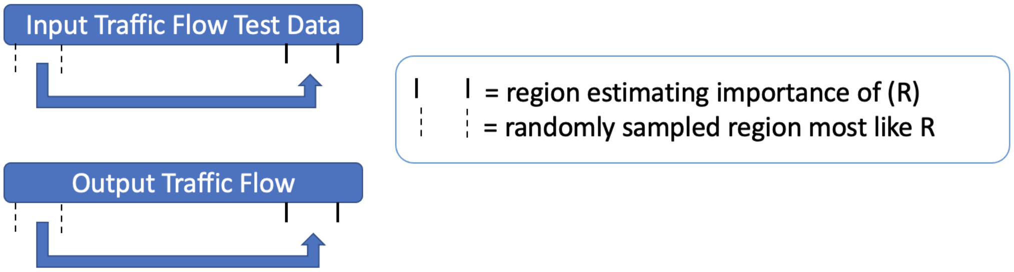

Input Traffic Flow Test Data
Output Traffic Flow
= region estimating importance of (R)
= randomly sampled region most like R

Figure 5: Illustration of perturbation method

Algorithm 1 find_optim_change algorithm: We estimate the regions with maximum and minimum rate of change with length s. vec is the time series sequence $x_1, ..., x_t$ that we are finding the maximum or minimum rate of changes in. $length$ is the number of time series observations per time series interval we examine the rate of change of, which has been ascertained empirically to be 150. k is a maximized number of intervals over vec to return with the highest or lowest rates of change. $threshold$ is the minimum number of consecutive differences that may be positive or negative. $neighbors$ is the range of the points that gaussian smoothing averages. The argument is used to denoise regions of data that may truly be increasing or decreasing, but would not satisfy the threshold of consecutive changes all being negative or positive. $is\_smallest$ is a boolean specifying whether we are finding the regions of data with the least or greatest rates of change (i.e. the least or greatest euclidean norm of consecutive positive or negative differences in traffic flow).

```
**procedure** FIND_OPTIM_CHANGE($vec, length, k, threshold, neighbors, is\_smallest$)
    $optim\_changes \leftarrow heap([])$
    **for** $i\ in\ range(length, len(vec))$ **do**
        $diffs \leftarrow deltas(np.array(vec[(i - length) : i]))$;
        $smoothed\_points \leftarrow linear\_smoother(diffs, neighbors)[1]$;
        **if** $get\_consecutive\_diffs(threshold, smoothed\_points)$ **then**
            $diffs \leftarrow list(filter(lambda\ x : x >= 0, diffs))$;
            $euclidean\_norm \leftarrow (-1\ if\ is\_smallest\ else 1) * np.sqrt(np.sum(np.square(np.array(diffs))))$;
            **if** $len(optim\_changes) < k$ **then**
                Push $[i - length, i, euclidean\_norm])$ onto $optim\_changes$;
            **else**
                $smallest\_change \leftarrow$ euclidean norm of smallest change in $optim\_changes$;
                **if** $(euclidean\_norm > smallest\_change)$ **then**
                    Remove $low\_val$ from $optim\_changes$;
                    Push $[i - length, i, euclidean\_norm]$ onto $optim\_changes$;
                **end if**
            **end if**
        **end if**
    **end for**
    $optim\_intervals \leftarrow list(map(lambda\ l : l[: 2], optim\_changes))$    ▷ [start,end,norm] → [start, end]
    $optim\_intervals \leftarrow MAX\_NUM\_INTERVALS(optim\_intervals)$    ▷ Calls Algorithm 2
**end procedure**
```

and speed of the vehicles constant as region R is effectively substituted out from the traffic flow time series, remeasured with the information from R masked out.

3.4 Resampling from Similar Regions

For similar regions to $R = D[i : j]$ of length $l_R = |R|$ to be selected, we set a threshold for the number of most similar regions to examine. Our case study examines 15 non-overlapping most similar regions. A heap data structure is maintained, where a stationary block-bootstrap-type of resampling is done; from one below the lower bound of R, or $i - 1$, contiguous blocks of traffic flow are generated such that their lengths are sampled from a geometric distribution with a success probability of $p = \frac{1}{window\_size}$, where the $window\_size$ is the expected length ($l_R = |R|$) of a contiguous block that's being resampled from the time series. The same is done for non-overlapping contiguous blocks that start from one above the upper bound of R, or j. The similar regions of the time series that will be resampled is illustrated in Figure 6. Two of these blocks that are presumed to be equivalent “concepts" or “features" of the time series are randomly chosen and the later block is substituted with earlier traffic flow data to avoid look-ahead bias. After effectively masking out the information from R or a region of the time series “equivalent" to R, we run the trained model over the data, obtain the prediction output in batches, and retrieve the new prediction MAEs that would have increased from the original data.

3.5 Evaluation

For the PEMS04 and PEMS08 datasets, we examine how the estimation method performs on the spatial-temporal graph convolution network implemented by Guo et al [40] and on the hierarchical GRU model by Du et al [2]. For a region with high rate of change defined in Algorithm 1 R_h and another with low rate of change R_l, we run the estimation method for these regions in batches of 25 observations. Each batch is averaged and each respective region, R_h and R_l, contains 30 of these means. The distribution of these means are compared, where a bootstrapped 2-sample test is done

Algorithm 2 are_overlapping, get_prec, max_num_intervals

```
procedure GET_PREC($intervals,\ j$)
    $prec \leftarrow None$;
    $curr\_idx \leftarrow j - 1$;
    while $curr\_idx \geq 0$ do
        $curr\_interval \leftarrow intervals[curr\_idx]$
        if $!are\_overlapping(curr\_interval, intervals[j])$ then
            $prec \leftarrow curr\_idx$;
            $break$;
        end if
        $curr\_idx - = 1$;
    end while
    return $prec$
end procedure

procedure MAX_NUM_INTERVALS($intervals$)
    $intervals.sort()$;
    $OPT \leftarrow [0] * len(intervals)$;
    for $j\ in\ range(len(intervals))$ do
        $prec\_idx \leftarrow get\_prec(intervals, j)$
        if $prec\_idx$ is None then
            $OPT[j] \leftarrow (1, [intervals[j]])$;
        else
            $OPT[j] \leftarrow ((OPT[prec\_idx][0] + 1), OPT[prec\_idx][1] + [intervals[j]])$
                    $if(OPT[prec\_idx][0] + 1) > OPT[j-1][0]$
                    $else\ OPT[j-1]$
        end if
    end for
    return $OPT[-1]$
end procedure
```

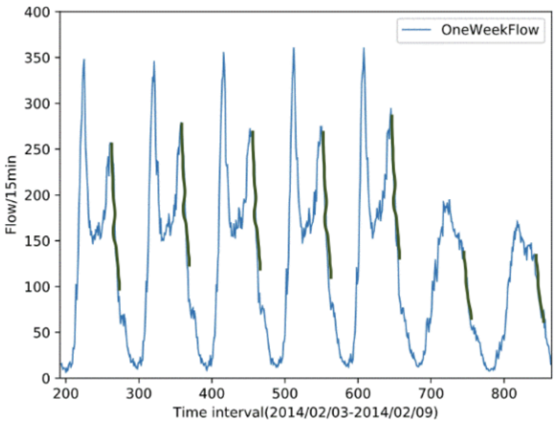


Figure 6: One week traffic flow data from Site A414 of the Highway Agency in England, based on [2]. The highlighted green is one example of regions of the time series that have similarly large changes in traffic flow.

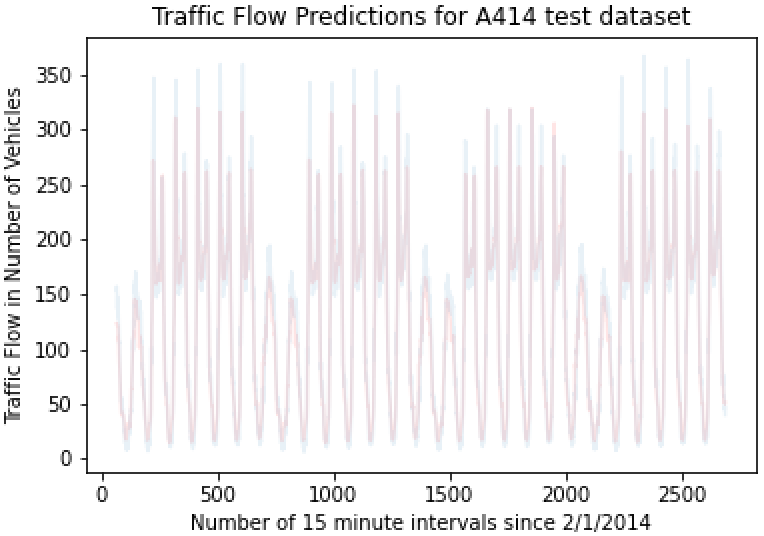


Figure 7: Traffic Flow predictions for A414 test dataset: after 35 epochs, the best weights were chosen and predictions were outputted from the resulting hierarchical neural model.

to identify whether to reject the hypothesis that reruns on average result in consistently higher or lower mean MAEs. Our evaluation of our estimation method comes from *stability*, which is the consistency of our estimation method to produce MAEs of perturbed data where, on average, masking out information from one region always results in worse average prediction error than the other.

4 Results

4.1 Baseline

The ground truth after running 65 epochs, that uses early stopping, with prediction over 11 points for the PEMS04 dataset is:

Dataset (3394 observations)	**MAE**	**RMSE**	**MAPE**
Original ASTGCN pytorch model	26.65	41.37	0.17

After running 77 epochs due to early stopping for the PEMS08 dataset, the prediction error over 11 data points is:

Dataset (3567 observations)	**MAE**	**RMSE**	**MAPE**
Original ASTGCN pytorch model			

Having recreated the following neural architecture for the A414 highway traffic dataset, we have the best weights after 35 epochs with early stopping. Prediction over the test data (2/1/2014-2/28/2014) is shown in Figure **??**.

4.2 Overall Estimations

4.2.1 PeMS dataset

4.2.2 A414 dataset

4.3 Stability

4.3.1 PeMS dataset

4.3.2 A414 dataset

5 Limitations and Future Work

Beyond the three datasets that we have applied our estimation method to, we can explore other traffic forecasting datasets collected from more recently recorded traffic sites and further test the stability of our estimation method. Particularly with the wide adoption of connected vehicle environments, more fine-grained data may also be collected about traffic congestion on highways and intersections.

Also, when we define the regions that we intend to perturb, we may consider not just traffic flow, but multiple variables and account for underlying causal relationships by summing indirect and direct effects in causal shapley values, as presented by Heskes et al. In addition to the regions that we perturbed, we could also increase the number of reruns of the estimation algorithm on the datasets given that time permits. This can result in lower variability and may give us a better idea whether the mean of the average change in prediction errors for one region is always in the same ordering relative to the other regions of a time series.

Our methodology makes the assumption that resampled regions from traffic flow that are most similar to some region R are virtually equivalent, may be treated as the same “feature" in a periodic traffic time series, and can be substituted with one another. However, similarity may not imply that these regions can replace one another as a means to “mask" out information from the input data for the model. The assumption enables flexible preliminary analysis of our estimation method as we can test a variety of hypotheses about importance of any region of the time series. However, we may afford further correctness by restricting the data to individual time steps $t_1, ..., t_n$ (i.e. reducing the flexibility of our estimation method) that are each considered a feature and implement a hypothesis testing framework that can ascertain the number of perturbations required to guarantee stability, as in Zhou et al [44].

Future work would also involve testing additional hypotheses under our estimation framework or alternative data importance approaches, which can allow us to further verify the stability of our estimation method and see whether adding estimation methods to offer an idea of data importance can help laymen better understand the behavior of black-box models.

6 Acknowledgements

This work was done for my Senior Honors Thesis under the supervision of Professor Freeman from the CMU Statistics Department, who I’d like to thank for guiding me through the process of brainstorming ideas and organizing the thesis paper.

References

[1] Highways Agency network journey time and traffic flow data, 2017. Last Updated 2018. Available: https://data.gov.uk/dataset/dft-eng-srn-routes-journey-times/

[2] Du, Shengdong, et al. “An LSTM based encoder-decoder model for MultiStep traffic flow prediction." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.

[3] Lipton, Zachary C. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." Queue 16.3 (2018): 31-57.

[4] Chen, Meng, Xiaohui Yu, and Yang Liu. “PCNN: Deep convolutional networks for short-term traffic congestion prediction." IEEE Transactions on Intelligent Transportation Systems 19.11 (2018): 3550-3559.

[5] Yang, Xiaoxue, et al. “Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models." Journal of Advanced Transportation 2020 (2020).

[6] Minnesota Department of Transportation. “Mn/DOT Traffic Data". *Datatools*, 22 March http://data.dot.state.mn.us/datatools/

[7] Tang, Jinjun, et al. “An improved fuzzy neural network for traffic speed prediction considering periodic characteristic." IEEE Transactions on Intelligent Transportation Systems 18.9 (2017): 2340-2350.

[8] Yu, Bing, Haoteng Yin, and Zhanxing Zhu. “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting." arXiv preprint arXiv:1709.04875 (2017).

[9] Lu, Huakang, et al. “St-trafficnet: A spatial-temporal deep learning network for traffic forecasting." Electronics 9.9 (2020): 1474.

[10] Yu, Rose, et al. “Deep learning: A generic approach for extreme condition traffic forecasting." Proceedings of the 2017 SIAM international Conference on Data Mining. Society for Industrial and Applied Mathematics, 2017.

[11] Tran, Luan, et al. “DeepTRANS: a deep learning system for public bus travel time estimation using traffic forecasting." Proceedings of the VLDB Endowment 13.12 (2020): 2957-2960.

[12] Geng, Xu, et al. “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.

[13] Zhang, Zhihao, et al. “Probe data-driven travel time forecasting for urban expressways by matching similar spatiotemporal traffic patterns." Transportation Research Part C: Emerging Technologies 85 (2017): 476-493.

[14] Wang, Dong, et al. “When will you arrive? estimating travel time based on deep neural networks." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[15] Manibardo, Eric L., Ibai Laña, and Javier Del Ser. “Deep learning for road traffic forecasting: Does it make a difference?." IEEE Transactions on Intelligent Transportation Systems (2021).

[16] Sun, Shiliang, Changshui Zhang, and Guoqiang Yu. “A Bayesian network approach to traffic flow forecasting." IEEE Transactions on intelligent transportation systems 7.1 (2006): 124-132.

[17] Barredo-Arrieta, Alejandro, Ibai Laña, and Javier Del Ser. “What lies beneath: A note on the explainability of black-box machine learning models for road traffic forecasting." 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019.

[18] Wu, Yuankai, et al. “A hybrid deep learning based traffic flow prediction method and its understanding." Transportation Research Part C: Emerging Technologies 90 (2018): 166-180.

[19] Saluja, Rohit, et al. “Towards a Rigorous Evaluation of Explainability for Multivariate Time Series." arXiv preprint arXiv:2104.04075 (2021).

[20] Selvam, Santhosh Kumar, and Chandrasekharan Rajendran. “tofee-tree: automatic feature engineering framework for modeling trend-cycle in time series forecasting." Neural Computing and Applications (2021): 1-20.

[21] Chen, Siheng, Yonina C. Eldar, and Lingxiao Zhao. “Graph unrolling networks: Interpretable neural networks for graph signal denoising." arXiv preprint arXiv:2006.01301 (2020).

[22] Barić, Domjan, et al. “Benchmarking Attention-Based Interpretability of Deep Learning in Multivariate Time Series Predictions." Entropy 23.2 (2021): 143.

[23] Ismail, Aya Abdelsalam, et al. “Benchmarking Deep Learning Interpretability in Time Series Predictions." arXiv preprint arXiv:2010.13924 (2020).

[24] Štrumbelj, Erik, and Igor Kononenko. “Explaining prediction models and individual predictions with feature contributions." Knowledge and information systems 41.3 (2014): 647-665.

[25] Tonekaboni, Sana, et al. “Explaining time series by counterfactuals." (2019).

[26] Cho, Sohee, et al. “Interpreting Internal Activation Patterns in Deep Temporal Neural Networks by Finding Prototypes." Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021.

[27] Delaney, Eoin, Derek Greene, and Mark T. Keane. “Instance-based counterfactual explanations for time series classification." International Conference on Case-Based Reasoning. Springer, Cham, 2021.

[28] Ates, Emre, et al. “Counterfactual Explanations for Multivariate Time Series." 2021 International Conference on Applied Artificial Intelligence (ICAPAI). IEEE, 2021.

[29] Keane, Mark T., et al. “If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques." arXiv preprint arXiv:2103.01035 (2021).

[30] Parvatharaju, Prathyush S., et al. “Learning Saliency Maps to Explain Deep Time Series Classifiers." (2021).

[31] Li, Yaguang, et al. “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting." arXiv preprint arXiv:1707.01926 (2017).

[32] García, María Vega, and José L. Aznarte. “Shapley additive explanations for NO2 forecasting." Ecological Informatics 56 (2020): 101039.

[33] Saluja, Rohit, et al. “Towards a Rigorous Evaluation of Explainability for Multivariate Time Series." arXiv preprint arXiv:2104.04075 (2021).

[34] Kumar, I. Elizabeth, et al. “Problems with Shapley-value-based explanations as feature importance measures." International Conference on Machine Learning. PMLR, 2020.

[35] Malinsky, Daniel, and Peter Spirtes. “Causal structure learning from multivariate time series in settings with unmeasured confounding." Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery. PMLR, 2018.

[36] Aas, Kjersti, Martin Jullum, and Anders Løland. “Explaining individual predictions when features are dependent: More accurate approximations to Shapley values." arXiv preprint arXiv:1903.10464 (2019).

[37] Heskes, Tom, et al. “Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models." arXiv preprint arXiv:2011.01625 (2020).

[38] Breunig, Markus M., et al. “LOF: identifying density-based local outliers." Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000.

[39] Hase, Peter, and Mohit Bansal. “Evaluating explainable AI: Which algorithmic explanations help users predict model behavior?." arXiv preprint arXiv:2005.01831 (2020).

[40] Guo, Shengnan, et al. “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.

[41] Highways England - WebTRIS - Faqs, https://webtris.highwaysengland.co.uk/Home/Faqs

[42] Anderková, Viera, and František Babič. "Better understandability of prediction models: a case study for data-based road safety management system." 2021 IEEE 21st International Symposium on Computational Intelligence and Informatics (CINTI). IEEE, 2021.

[43] Yuan, Chen, et al. “Application of explainable machine learning for real-time safety analysis toward a connected vehicle environment." Accident Analysis & Prevention 171 (2022): 106681.

[44] Zhou, Zhengze, Giles Hooker, and Fei Wang. “S-lime: Stabilized-lime for model explanation." Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021.