A Meta-Learner Ensemble Framework for Real-Time Short-Term Traffic Speed Forecasting

Northwestern University Transportation Center

Divyakant Tahlyan†, Eunhye Kim† and Hani Mahmassani†

†Transportation Center, Northwestern University

Northwestern University

Background

- Problem of **real-time prediction** of traffic states like speed, flow, density or travel time is important from a variety of perspectives including:
  - **Traffic operations** like signal control, variable message signs or other congestion mitigation strategies.
  - **Users' perspective** like Navigation systems, trip planning and pricing algorithms for ride hailing services.
- Previous studies have used various sources of data to predict future traffic states including loop detectors, GPS data and video data.
- From methodological perspective, popular methods include:
  - **Statistical Methods** like ARIMA, State-Space Models etc.
  - **Machine-Learning Methods** like Neural Networks, KNN Regression, Support Vector Machines.
  - Traffic Flow Theory based **Traffic Estimation and Prediction Systems** (TrEPS) like Dynasmart-X, DynaMIT-R.
  - **Hybrid Methods** where two or more methods are combined in some form.

Potential Issues

We identify four potential issues with the existing approaches often used for real-time traffic prediction:

- **Restrictive Hypothesis Space** — true data generating process might lie outside the hypothesized model form like using a linear model form when the data generating process is non-linear.
- **Hyper-parameter tuning** — even when a generalized model form like neural networks are used, determining optimal hyper-parameters is tricky.
- **No single best model** — often a single model does not outperform all other models in all situations.
- **Learning from mistakes** — most approaches do not have a feedback loop to ensure learning from the mistakes made in the past.

Current Study

- We identify and address two questions in this study:
  - Can we combine different models to improve performance?
  - Can we learn from the prediction mistakes made in the past?
- We use ***ensemble learning*** to address these issues. Various types ensemble learning techniques include bagging, boosting, random subspace sampling & stacking.

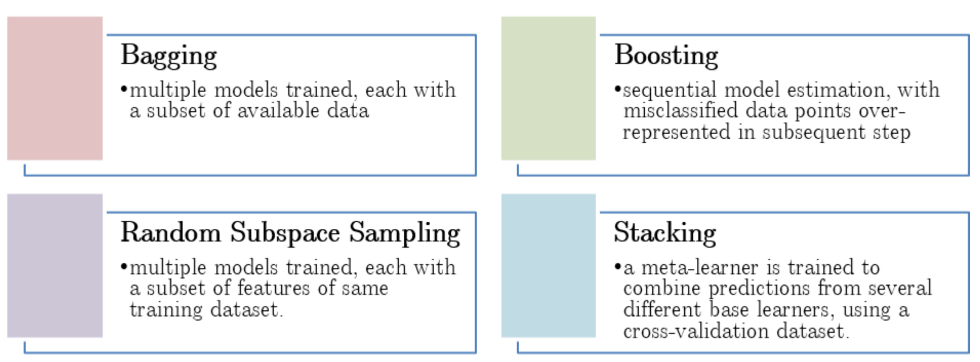

Fig. 1: Types of Ensemble Learning Methods

- We explore the use of ***stacking*** [1, 2] to combine predictions from four models (KNN regression, Multilayer perceptron, ARIMA, and Dynasmart-X) using predictive performance of the individual models in the recent past.
- We apply the proposed approach from 24-hour traffic speed data from 10 different loop detectors in Kansas City.

Stacking for Traffic-Speed Forecasting

- Stacking consists of two levels — **level-0** consists of multiple models making independent predictions and **level-1** consists of combining predictions from level-0 models using a **meta-learner**.
- While in a cross-sectional context stacking involves using cross-validation dataset to train the meta-learner, we use in-flowing data and the corresponding predictions in the previous steps to train the meta-learner.

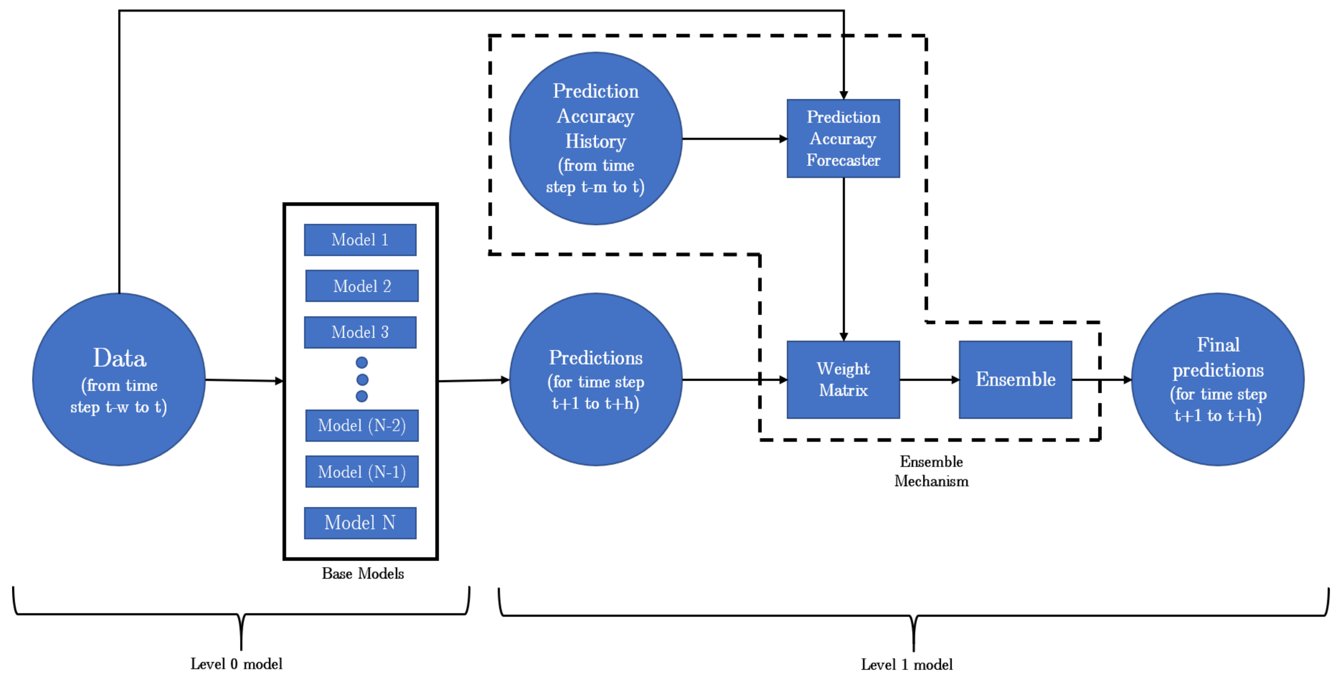

Fig. 2: Conceptual Framework for Stacking

- We operate at a **rolling-horizon** basis, i.e. at time step t, use data from time step $t - w$ to t to predict speed values at future time steps.
- For meta-learner, we explore two different algorithms:
  - Non-negative least square estimation (NNLS) [1]
  - K-Nearest Neighbors

Case Study

- 24-hour traffic speed data, available as space mean speed at 1-minute interval, from 10 different loop detectors on interstate 435 in Kansas City is used.
- At every time step, data is used to predict traffic speed 15 minutes in future.

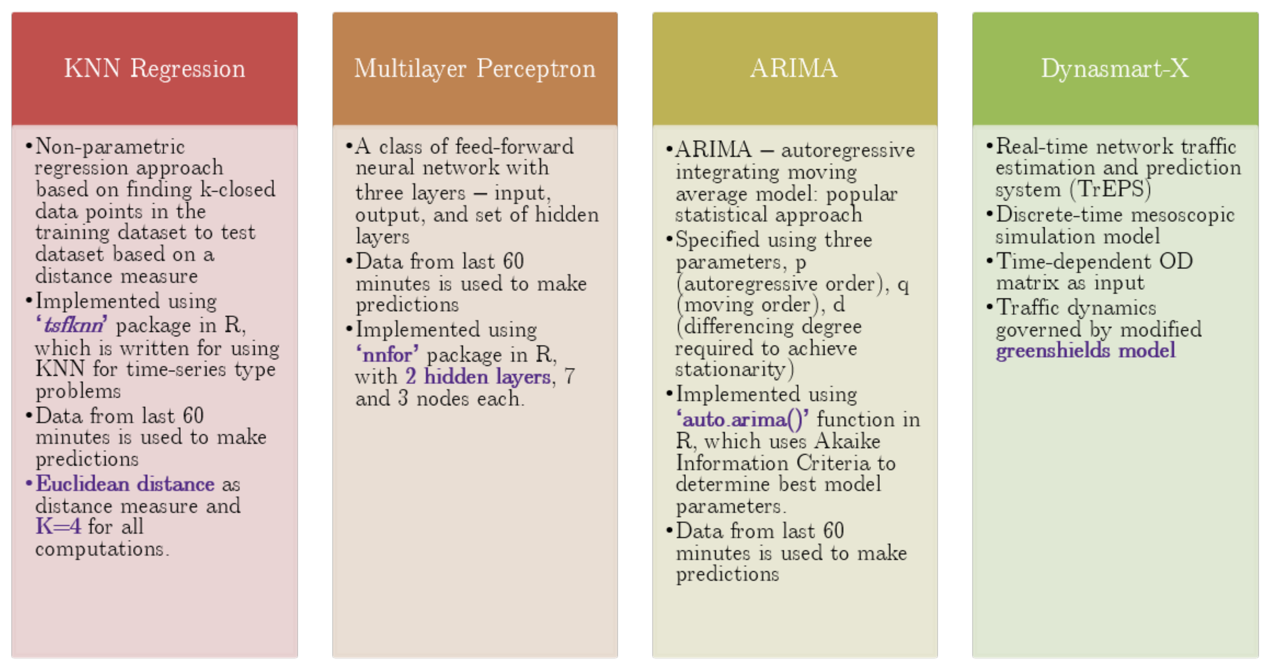

Fig. 3: Level-0 Models

- For meta-learning, a **warm-up period** is assumed, which involved taking average of level-0 predictions so that sufficient data to train meta-learner is collected.
- Different lengths of **historical performance data** explored to train meta-learner — 5 to 60 minutes in the history at 5 minutes interval.
- **Performance Measures**: Absolute Percentage Error; Number of instances with least APE value for different methods.

Results

- For NNLS, total APE first decreases then increases with increasing length of the performance history. For KNN, total APE mostly increases with increasing length of the performance history.
- In most cases, for at least one value of performance history, total APE values are either better or close to the best performing level-0 model.
- Cumulative APE is relatively higher for NNLS meta-learner around the evening peak but does not deteriorate that much for the KNN meta-learner.
- However, NNLS performes better than KNN meta-learner and other level-0 models in terms of number of instances with least APE.

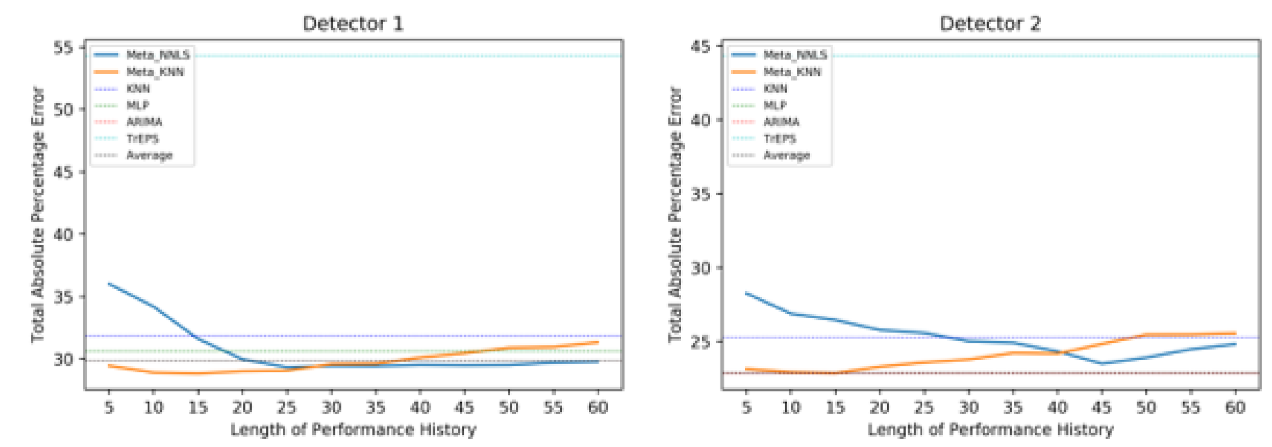

Fig. 4: Total absolute percentage error for different lengths of performance history for detector 1 and 2

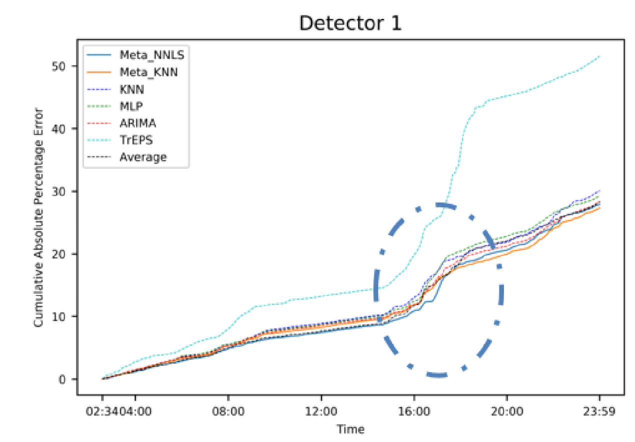

Fig. 5: Cumulative absolute percentage error for different models for detector 1

	Meta_NNLS	Meta_KNN	KNN	MLP	ARIMA	TrEPS	Average
Detector 1	251	192	173	155	154	222	140
Detector 2	180	171	175	165	193	214	189
Detector 3	200	187	145	172	211	179	189
Detector 4	206	156	196	150	148	229	202
Detector 5	199	174	163	141	201	234	175
Detector 6	195	164	181	144	175	195	232
Detector 7	188	229	165	151	197	160	206
Detector 8	184	216	228	139	155	208	157
Detector 9	212	274	186	176	197	104	137
Detector 10	205	191	168	182	223	91	226
Total	**2020**	**1954**	**1780**	**1575**	**1854**	**1836**	**1853**

Tab. 1: No. of instances with least APE for a particular methods for each detector

- Overall, while the meta-learners used in this study do not always guarantee superior performance than level-0 models, the explored stacking approach seems to work better during the non-peak hours.
- Perhaps, an alternative here would be to not use just immediate performance history but also incorporate performance from previous days.

Acknowledgement

This work is based in part on data from a project funded by the US DOT, FHWA through Leidos, Inc. on the development and testing of an Integrated Modeling for Road Condition Prediction (IMRCP), in partnership with Synesis, Inc.

References

[1] Breiman, L., 1996. Stacked regressions. Machine learning, 24(1), pp.49-64.

[2] Wolpert, D.H., 1992. Stacked generalization. Neural networks, 5(2), pp.241-259.

100th Annual Meeting of the Transportation Research Board (Paper ID:21-02198)