Probabilistic Deep Learning for Traffic Density Prediction

Transportation Research Record
0000, Vol. XX(X) 1–19

DOI: 10.1177/ToBeAssigned
journals.sagepub.com/home/trr
SAGE

Pedro Cesar Lopes Gerum[1], Andrew Reed Benton[2] and Melike Baykal-Gürsoy[2]

Abstract

Real-time, accurate predictions of recurrent and non-recurrent traffic congestion is essential for optimizing transportation systems and ensuring a smooth user experience. Traditional models often focus on long-term point estimates, limiting their use in scenarios requiring short-term predictions or probabilistic assessments (e.g., traffic signal optimization, dynamic tolling, emergency response). This study explores probabilistic deep learning for real-time traffic density distribution prediction. We demonstrate that an adapted Multi-Quantile Recurrent Neural Network (MQRNN), which we term MQRNN-monotonic, outperforms traditional time-series methods, particularly when handling non-recurrent disruptions. A novel loss function is introduced to address quantile crossing issues, ensuring valid distributional predictions. Experiments on two highway data sets show that probabilistic deep learning for traffic density prediction yields well-calibrated and sharp dynamic traffic congestion distributions. This research offers a promising new approach to real-time traffic density forecasting, paving the way for transportation systems that respond faster and smarter to changing road conditions, making traffic smoother, more sustainable, and more predictable for everyone.

1 Introduction

In 2017, traffic congestion caused annual delays and financial losses estimated at €130B (approximately $145B) in Europe [1]. Congestion not only impacts economic productivity but also contributes significantly to environmental damage through increased emissions and fuel consumption. Addressing this challenge is crucial for achieving sustainable transportation systems that are efficient, resilient, and environmentally friendly. Intelligent transportation systems have traditionally relied on real-time or short-term predictions of traffic flow and travel times. This study diverges from traditional methods by directly targeting traffic density forecasting. Because of the direct relationship between traffic density and traffic congestion, we cater to situations in which congestion forecasts are vital, such as optimizing traffic signal timings in smart cities, dynamic toll pricing, toll-booth readiness, dynamic estimation of probability of traffic breakdown [2], evacuation planning [3], and emergency response coordination.

While traffic density is related to traffic flow and speed through the fundamental diagram of traffic flow, its statistical properties are distinct. Research has shown that estimating the density from flow and speed measurements can be highly unreliable, particularly under congested conditions where the fundamental relationship breaks down [4]. Traffic density, as a direct measure of road crowdedness, often exhibits more abrupt changes and higher volatility, especially in response to non-recurrent events like accidents or large gatherings. In contrast, traffic flow can be smoother and may even decrease as density reaches critical levels (i.e., in a traffic jam).

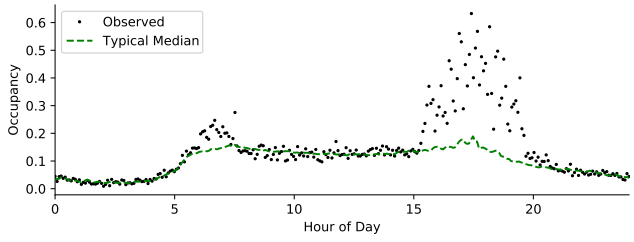
Historically, research on traffic density prediction has addressed long-term planning and recurrent congestion, utilizing adaptations of established models such as the Cell Transmission Model (CTM) [5] and the Cellular Automata Model [6]. Despite efforts to gauge the probability of traffic breakdowns [7] and recent strides in incorporating non-recurrent incidents through queuing theory principles [8, 9], a gap remains in real-time applicability where constant traffic changes and continuous data inflow are present. These applications demand up-to-date short-term traffic forecasts [10].

Moreover, efforts focused on addressing recurrent congestion have mostly neglected unforeseen events, such as extreme weather conditions or large gatherings, which can abruptly alter traffic patterns. Figures 1a and 1b depict instances in which unusual traffic congestion occurred due to a football game and a snowstorm. These situations highlight the need for adaptable and robust forecasting methods

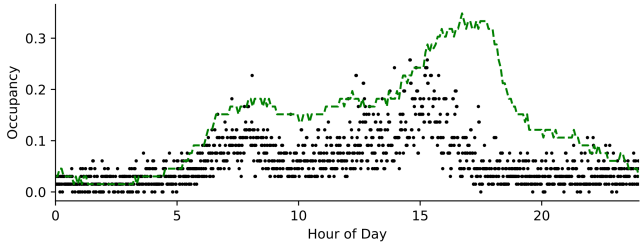
[1]Operations & Supply Chain Management Department, Cleveland State University
[2]Department of Industrial and Systems Engineering, Rutgers University

Corresponding author:
Pedro Cesar Lopes Gerum, pedro.gerum@gmail.com

that can inform traffic management strategies and promote sustainable transportation choices even under unpredictable conditions. On October 10, 2013, a football game in Chicago led to a surge in traffic beginning at around 4 pm, peaking at approximately three times the median congestion of other days at 5:30 pm. In constrast, a large snowstorm in Milwaukee on February 5, 2008, disrupted the regular flow of traffic, suggesting that a considerable number of people chose to work from home, which significantly altered the typical rush hour patterns. These real-world scenarios underline the urgency for real-time, adaptable solutions that can respond to dynamic traffic conditions. However, the customary long-term prediction models often overlook these cases because their effects are diluted over the long planning horizon. The models in the literature that consider them are limited to classifying traffic behavior as belonging to an accident or not, instead of capturing the true impact of the accident in congestion [11].



(a) Bear's game, Chicago, IL



(b) Snow day, Milwaukee, WI

Figure 1. Traffic occupancy during out-of-the-ordinary events in Chicago and Milwaukee

A third significant limitation of current methodologies is their provision of point estimates rather than distributions [12]. Point estimates lack the depth of probabilistic predictions, which afford a full spectrum of traffic density distributions, thus equipping practitioners with a robust tool for understanding associated risks [13].

This research intends to bridge these gaps by investigating the performance of a probabilistic deep learning framework for traffic density forecasting. We choose the Multi-Quantile Recurrent Neural Network (MQRNN) architecture [14] for traffic density forecasting because it has been tested and validated in several other applications. Central to this architecture is its ability to provide multi-horizon, multi-quantile estimates, which offer a comprehensive view of potential outcomes and associated risks. However, quantile-based probabilistic deep learning frameworks, such as MQRNN, still possess an inherent issue of quantile crossings, a phenomenon where the generated quantile estimates are not monotonically increasing. As a result, the model occasionally yields invalid probabilistic forecasts. To address this problem, we propose a straightforward yet innovative adaptation that removes quantile crossings, thus enhancing the reliability of the predictions and ensuring their validity. This adaptation maintains the performance of the MQRNN while offering a solution that is adaptable to other architectures with minimal alterations.

In recent years, newer deep learning architectures for time-series forecasting, such as N-HiTS [15] and Transformers [16], have emerged, showcasing promising results in capturing complex temporal dependencies and long-range patterns. Notably, some of these newer architectures, like N-HiTS, inherently address the issue of quantile crossings through their specific design. Despite these advancements, MQRNN remains a strong contender for probabilistic time series forecasting, particularly in applications where interpretability and computational efficiency are crucial. Moreover, at the time of this study, MQRNN had a more established track record in probabilistic forecasting compared to the newer models, which had limited empirical validation and community adoption.

Figure 2 illustrates the advancements introduced in this study compared to the short-term traffic forecasting current literature. The figure also emphasizes some of the applications whose decisions may be improved by this research. In summary, we provide a discerning approach to real-time traffic density prediction that not only addresses existing gaps but also sets a precedent in employing MQRNN for traffic density predictions, advancing promises for a considerable leap toward fluid traffic dynamics and smarter cities. As depicted in the yellow boxes in Figure 2, the main contributions and innovations of this research are:

1. pioneering the use of probabilistic deep learning for real-time traffic density predictions by implementing the MQRNN model for traffic density prediction and validating it with real data;
2. mitigating the issue of quantile crossings in MQRNN, ensuring the validity of probabilistic forecasts;
3. verifying that deep learning methods can outperform traditional methods when predicting non-recurrent congestion.

Next, in Section 2, we discuss the current state-of-the-art in short-term forecasting and deep learning for traffic applications and their limitations. In Section 3, we introduce the preliminary methodology of deep learning models for dynamic traffic density forecasting. What follows in Section

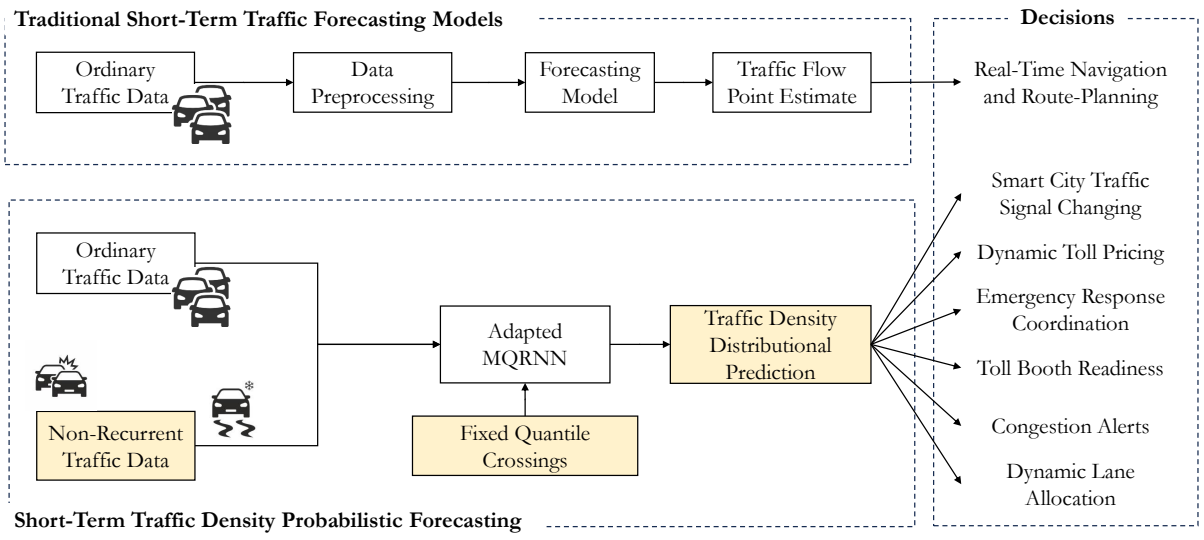


Figure 2. Flowchart of Traffic Forecasting Models and the Contributions of this Research (in yellow)

4 is our proposed adaptation to neural-network probabilistic models to enforce monotonicity. Section 5 describes the data used, and Section 6 discusses the validation of the investigated models and other traditional benchmarks using two data sets. Section 7 presents the performance of various models and compares them when predicting non-recurrent congestion. Then, we present a decision-making scenario in Section 8 to validate how the probabilistic forecasts inform a real-world cost-benefit analysis for traffic management. Finally, Section 9 details our conclusions.

2 Literature Review

The literature on short-term traffic forecasting is extensive. Vlahogianni, Karlaftis, and Golias [17] compiles several representative papers that have proposed significant models. The authors also outline some of the remaining challenges that this research addresses, including accurately forecasting unexpected events, providing measures of reliability for forecasts such as distributions, and working with incomplete or problematic data sets. We note that the literature has given little attention to congestion prediction (i.e., traffic occupancy and density), instead focusing on flow, speed, travel time predictions, and binary classification of congestion, as discussed next.

2.1 Traditional Traffic Congestion Forecasting Approaches

Traditional CTM-based models can theoretically compute transient congestion behaviors, but they require extensive simulations. This barrier renders them cumbersome in practice. Kurzhanskiy [18] presents a variation of the original CTM designed explicitly for transient traffic density prediction. As a possible alternative for simulation-based models, Chrobok et al. [19] describe two simple prediction models for short-term forecasting using historical data. The constant model forecasts the same value for all time horizons, while the linear model fits a linear curve from the most recent N measurements. Zhang, Yu, and Lei [20] predict short-term congestion with a Genetic Hierarchical Fuzzy-rules based model. In both papers, the parameters used are deterministic. Conversely, Zhang et al. [21] include the probabilistic nature of traffic in their genetic algorithm-based approach for congestion prediction. The output, in this case, is binary (congestion/no-congestion).

Due to the time dependency of real-time predictions, time-series forecasting is a sensible approach for this type of problem. However, traditional methods, such as Auto-Regressive Integrated Moving Average (ARIMA) [22], Smooth Transition Auto-Regressive (STAR) [23], and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) [24], perform poorly on multi-step forecasting problems and have not been designed for irregular traffic with sudden instabilities unless they are used for anomaly detection. They are also computationally expensive. To improve traditional Seasonal ARIMA (SARIMA) models, Ghosh, Basu, and O'Mahony [25] propose a Bayesian approach to estimate the coefficients when predicting traffic flow. As a result, they obtain the distribution for each linear coefficient of the traditional SARIMA models. Petridis et al. [26] introduce the Bayesian Combined Model for time-series forecasting. The innovative idea is to use local predictors from different models and combine them using posterior probabilities that *vote* for the best prediction. Wang, Deng,

and Guo [27] and Gu et al. [28] use this idea to combine forecasts from ARIMA, Kalman Filters, and Deep Learning Methods to forecast traffic flow.

2.1.1 Out-of-the-ordinary Events Several authors have focused on the challenge of predicting unexpected events when forecasting traffic conditions, as non-recurrent events may have a high impact on traffic and cause significantly elevated congestion levels. Many traffic forecasting models include these unexpected events by making the models multi-regime. Vlahogianni [29] and Kamarianakis, Gao, and Prastacos [30] study traffic prediction models that function separately for congested and non-congested situations. Similarly, [31] decomposes traffic deterioration into six sources. [32] focuses on travel time prediction during incidents, and Fei, Lu, and Liu [33] and Min and Wynter [34] include weather, incident data, and current or planned roadwork as regimes. All these models assume that only a discrete number of regimes may occur and that the state of the system is known. They also do not predict traffic congestion (density or occupancy), but traffic volume and speed. Gerum, Benton, and Baykal-Gürsoy [35] and Gerum and Baykal-Gürsoy [36] extend the queuing theoretic model of Baykal-Gürsoy and Xiao [37] and Baykal-Gürsoy, Xiao, and Ozbay [38] to predict the long-term distribution of traffic density in corridors that may experience unexpected events. Baykal-Gürsoy et al. [39] shows that queueing theory could also be used to derive a closed-form solution for the stationary travel time distribution in such corridors. Finally, Gao et al. [40] studied travel time prediction subject to illegally parked vehicles using a similar queuing framework. While these models address the randomness aspect of the different regimes' occurrence, they still assume a finite, known number of regimes.

Except [35, 36, 37, 38, 39], the majority of these models, however, produce only point-estimate predictions and can be ineffective when making a significant number of forecasts quickly. Many of them still rely on ARIMA processes and thus face similar challenges. These models are computationally expensive and heavily dependent on linear relationships. Finally, none of the methods have been tested for short-term traffic density or occupancy forecasts. These limitations have led many researchers to seek alternative methods, such as direct deep learning approaches for traffic forecasting.

2.2 Deep Learning Methods for Traffic Forecasting

The recent influx of data from new sources has been a boon for contemporary research on traffic congestion in dynamic systems, enabling machine learning approaches to advance. Initially proposed by Dougherty, Kirby, and Boyle [41], these seminal models divide roadways into segments observed at discrete time units. Succeeding work by Dia [42] suggests that dynamic-architecture works, such as Recurrent Neural Networks (RNNs), can outperform multi-layered perceptrons (MLPs). He uses metrics of speed and flow as inputs in his model and defines congestion as the combination of speed and flow parameters. These early successes highlighted the potential of deep learning, particularly when paired with novel data sources. For instance, Zhu et al. [43] showed that DL models were significantly more effective than traditional computer vision for traffic video analysis, a success enabled by their creation of a large, high-resolution dataset from Unmanned Aerial Vehicles (UAVs).

Notably, Polson and Sokolov [44] describe how a deep fully-connected neural network can predict changes in traffic flow caused by external events such as sports games or accidents. The authors conclude that recent information (i.e., within the last 40 min) provides stronger predictors than older historical information. They predict the traffic flow 40 minutes in the future. Unfortunately, their approach requires preprocessing the data using a median filter in order to improve the forecasts because of the sparsity of the data. Zhao et al. [45], and Zhong et al. [46] follow on the work of Dia [42] and show that deep Long-Short Term Memory networks (LSTMs) are well suited to account for the time-dependence in traffic congestion. Chen, Yu, and Liu [47] developed a convolutional neural network (CNN) based approach to the same problem, yet their results are weaker than those obtained with other proposed architectures. These models also differ in the inputs used. Most use common variations of traditional traffic metrics (flow and speed). One particular example that predicts occupancy is described in Aqib et al. [48]. Lastly, Yao, Zhang, and Zhang [49] create a model that splits the data into several groups and chooses between linear and non-linear methods for each group, depending on the data volatility. The suitability of these architectures is continually reaffirmed in the latest research; for instance, Chinthakunta, Sunkavalli, and Koduru [50] employ LSTMs and GRUs as part of a multi-faceted system for predicting traffic flow patterns from drone data, and Ismaeel et al. [51] uses RNNs to predict traffic patterns in smart, sustainable cities. As the field has matured, newer architectures like Transformers [52] and Graph Neural Networks [53] have also been proposed for traffic prediction.

A more recent paradigm seeks to merge data-driven methods with fundamental domain knowledge. Wilkman et al. [54], for example, pioneer a framework using Physics-Informed Neural Networks (PINNs) that embeds a macroscopic traffic flow model directly into the neural network's loss function. This hybrid approach is designed for online, real-time adaptation from sparse data, showing a promising direction for creating more robust and generalizable models.

A comprehensive list of studies can be found in the four surveys that have compiled the most relevant methods [55, 56, 57, 58]. Overall, there appears to be a consensus in the

literature on the promising results of using deep learning methods for traffic forecasting [59]. However, there is still no consensus regarding the most appropriate input for traffic density deep-learning-based regressors.

2.2.1 Limitations The four surveys [55, 56, 57, 58] agree that, despite the advancements in prediction accuracy, some of the challenges discussed in Vlahogianni, Karlaftis, and Golias [17] remain. Most current deep learning models have unsatisfactory performance during out-of-the-ordinary events. Like previous approaches, most deep learning models do not directly focus on density or occupancy as measures for congestion assessment. A notable exception is the work of [48], which utilized a deep neural network for occupancy prediction; however, their approach was confined to deterministic point-estimates, providing no information about predictive uncertainty.

Furthermore, typical solutions still only provide point estimates, and their accuracy decreases as the forecast horizon increases. Conversely, distributional predictions give decision-makers a complete picture of the risks and probabilities. Many reliability metrics (e.g., probability of breakdown) are immediately obtained from the distribution but not from the expectation. Traffic breakdown is triggered when a substantial speed decrease from the free flow speed occurs between two consecutive time intervals [60]. This speed decrease causes a drastic increase in density and a sudden plunge in capacity [61]. Complete or extreme traffic breakdown occurs when capacity reaches zero.

2.3 Probabilistic Deep Learning

The literature discussing neural networks for probabilistic forecasting is much more limited than that for point-estimate forecasting. Within this domain, two primary paradigms exist: parametric and non-parametric approaches. Parametric models assume the data follows a specific probability distribution (e.g., Gaussian or Gamma). The neural network is then trained to predict the parameters of this chosen distribution. While this can be highly efficient, it risks poor performance if the true data distribution is complex or misspecified. The DeepAR model of Salinas et al. [62], which we use as a benchmark, exemplifies this parametric approach.

In contrast, the non-parametric approach of quantile regression, which we adopt in this study, makes no prior assumptions about the shape of the distribution. Common deep-learning probabilistic models forecast single quantiles for the forecasted distribution, instead of the traditional expected value. By predicting multiple quantiles, one can empirically map out the cumulative distribution function. This provides greater flexibility to capture the arbitrary, often skewed, distributions found in real-world traffic data. This approach has been implemented using linear models [63], random forests [64], gradient boosted trees [65], and recently, deep learning [14, 66]. When developing quantile models, two main challenges are selecting a sensible loss function and enforcing quantile monotonicity. We address these questions in Section 3.3.

Meinshausen and Ridgeway [64] apply probabilistic deep learning models to several applications, including the prediction of fuel utilization, labor worth, and house pricing. None of the three applications includes time series as inputs, and the number of training samples is small for all implementations. For example, Landry et al. [65] focus on the prediction of wind power. To the best of our knowledge, no probabilistic deep learning models have focused on traffic density. In fact, only a few research works directly discuss traffic forecasting. Looking at the two closest studies, Rodrigues and Pereira [66] forecast taxi demand and traffic speed, and Salinas et al. [62] include occupancy as one of many benchmark data sets. The data contains hourly occupancy rates for 963 lanes, and the authors only report the point forecast accuracy of their algorithm in [62]. While these works show the effectiveness of deep learning for distributional predictions, they have left room for considerable model specialization.

More recently, with the boon of generative methods, Tang and Matteson [67] propose a new method where the deep learning model generates a synthetic sample from which empirical distributions can be derived. This method, although innovative, is computationally intensive and requires more data for training than other methods. Furthermore, many new solutions still rely on heavy data preprocessing [44, 45, 47, 49, 68, 69] that may vary across data sets. Due to the subjectivity of the chosen preprocessing steps, practitioners may also benefit from a robust and consistent solution across datasets. Finally, Yoon et al. [70] show that probabilistic deep learning models can be sensitive to input perturbations. In these cases, a slight change in the input can result in significant variations in the output distribution.

In summary, to the best of our knowledge, no study has implemented machine-learning-based probabilistic forecasting models that directly use density or occupancy as the response variable. The existing probabilistic forecasting literature gives little attention to traffic distributional forecasting. So we set out to consider this task.

3 Preliminary Methodology

Traffic density can be affected by many factors. Some are predictable and, therefore, easier to implement into the models (e.g., as seasonality factors). Others may not be. For example, certain events such as sporting matches and festivals, and weather events such as snowstorms and floods may significantly impact traffic congestion. The influence of these factors is typically present in data, but it is often hard to extract such high-level features directly. The main idea of deep learning is to break down such complex abstract factors into simpler representations [71].

3.1 Deep Learning

Deep learning models approximate high-dimensional functions with a sequence of nested linear transformations, each followed by an element-wise non-linear transformation (called an activation function). Many popular models can be represented as an acyclic graph. In these graphs, nodes are organized in layers f^i's, which represent the intermediate transformations, and the edges connecting the layers represent the parameters θ for the linear transformations (also known as weights and biases). The value for each subsequent node is computed using the linear transformations and the activation function.

Mathematically, an artificial neural network is defined to be the composition of parameterized linear and non-linear mappings

$$f_\theta(\mathbf{x}) = \left(f^L \circ f^{L-1} \circ \cdots \circ f^1\right)(\mathbf{x}),$$

with $\mathbf{x}$ as a high dimensional vector containing the input information and $L \in \mathbb{N}$ as the number of mappings or "layers" used. The universal approximation theorem [72, 73] states that feedforward networks with a linear output layer and at least one "squashing" non-linear activation function can approximate any continuous function on a closed and bounded subset of $\mathbb{R}^n$. This theorem guarantees that there exists a deep learning framework that can approximately represent any non-linear function $f_\theta(\mathbf{x})$, but does not determine how large a network is necessary to obtain the correct representation [71]. Fortunately, tight bounds have been identified for the complexity required to achieve arbitrary accuracy [74].

In practice the function $f^i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ is usually of the form

$$f^i(\mathbf{x}) = \phi(\mathbf{W}_i(\theta)\mathbf{x} + \mathbf{b}_i(\theta)),$$

with matrices $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$, and vectors $\mathbf{b}_i \in \mathbb{R}^{n_i}$ in θ, and ϕ as a nonlinear function. Traditionally, ϕ is the sigmoid function $\phi(x) = \frac{1}{1+\exp(-x)}$, although modern implementations often prefer simpler and more numerically stable functions, such as $\phi(x) = \max(x, 0)$. Many other options are possible and reasonable, particularly on the output layer of f^L, where special forms of ϕ are often required to ensure the correct range for f_θ.

We say that we train a neural network when we select optimal parameters $\theta^\star$ to minimize a particular loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$. In a sense, the loss function measures the prediction error between $f_\theta(\mathbf{x})$ and some observed y, evaluated as $\ell(f_\theta(\mathbf{x}), y)$. The output of the neural network, $f_{\theta^\star}(\mathbf{x})$, depends on the chosen loss function and is typically a point estimate prediction (the median and mean in the case of mean absolute error, or ℓ_1, and mean squared error, or ℓ_2 losses, respectively) of the underlying distribution of the observed y. For some applications, point estimate predictions provide a limited picture of the system. Ideally, models should obtain the complete distribution of traffic density. Probability distributions provide information on risks and likelihoods of extreme events, leading to better incident and congestion management practices [35, 36].

Linear mappings are usually easy to train because of the convexity properties of the most common loss functions. On the other hand, optimizing non-linear mappings exactly is generally nonviable. Instead, we approximate $\theta^\star$ with sub-optimal parameters obtained using first-order methods such as stochastic gradient descent [75, 76]. For a data set $\{(X_1, y_1), \ldots, (X_k, y_k)\}$, the classical gradient descent method iterates over the data set

$$\theta_{n+1} = \theta_n - \eta \sum_{i=1}^{k} \nabla \ell(f_{\theta_n}(X_i), y_i),$$

with $\eta > 0$ as a step-size constant. The gradient descent method is a dominant approach because the gradients are easily computed for broad forms of f_{θ_n} by a software application of the chain rule (called auto-differentiation). Furthermore, it is possible to compute gradient observations only over subsets of the data set (mini-batches). Hence, the entire dataset does not need to be loaded into memory simultaneously, and the gradient computation can be parallelized across the mini-batch at each iteration. Modern implementations add many refinements to this approach, for example, by allowing the step size η to vary over time and adapt with the history of the process or incorporating "momentum" terms that average over previous gradients. See Ruder [76] for an overview of popular methods.

The exact architecture of a network varies significantly depending on the task. For standard tabular data sets, in which each observation is described by a vector, the structure we have described, called a multi-layered perceptron, is sufficient. For sequential data, Rumelhart, Hinton, and Williams [77] developed Recurrent Neural Networks (RNNs), which have become the standard in the literature. The clever idea behind RNNs is to keep the architecture as an acyclic graph by unrolling (or unfolding) the recurrent computations into a graph with a repetitive structure. Hence, each hidden layer $f^i_{(t)}$ is no longer a function of $\mathbf{x}_{(t)}$, but of all the previous history $\left(\mathbf{x}_{(t-1)}, \ldots, \mathbf{x}_{(1)}\right)$. This family of neural networks is instrumental in traffic forecasting because they allow models to incorporate time-related correlation patterns in their predictions.

3.2 Short-term Probabilistic Traffic Density Forecasting

The most successful probabilistic deep learning models are either well-established simple multi-output RNNs [78, 79] or variants of the sequence-to-sequence model of Sutskever, Vinyals, and Le [80]. Such models have achieved success in many fields [81, 82, 83].

These models are divided into an encoder and a decoder. Given a time series $z_{1:t}$ (in this case, the most recent

contiguous history of traffic density up to time t), the encoder reduces the time series to a hidden state of fixed length:

$$\mathbf{u}_t = h(z_{1:t}).$$

The encoder is typically an RNN, such as the LSTM architecture [78], or some other variant [84]. RNNs are deep learning architectures representing the observations up-to-the current time t as a hidden internal state $\mathbf{u}_t$, so this usage is natural. One apparent limitation of the encoder is that the output $\mathbf{u}_t$ may not capture all the information contained in the sequence $z_{1:t}$.

The decoder is then conditioned on this hidden state and other possible exogenous variables to predict, for example, the median of the next time step:

$$\hat{q}_{t+1,0.5} = g(\mathbf{u}_t, \mathrm{ex}_t).$$

The decoder is typically a multilayer perceptron, although an RNN [85] or a Convolutional Neural Network (CNN) [86] may also be used. Both the MQRNN architecture of Wen et al. [14] and the DeepAR architecture of Salinas et al. [62] follow this procedure, although in Salinas et al. [62] the decoder only serves to project $\mathbf{u}_t$ into the space of valid parameters. This architecture has the advantage of training on each observation without the computational overhead of a more naive implementation. Figure 3 illustrates the encoder-decoder sequence-to-sequence architecture we implement in this paper. In the figure, the history of the time series, starting from the beginning to the current moment t, is encoded in the vector $\mathbf{u}_t$. This vector is obtained by updating the encoded $\mathbf{u}_{t-1}$ with the current data point in the time series, z_t. Then, $\mathbf{u}_t$, along with exogenous factors represented in the vector ex_t, serve as inputs for the decoder to predict a set of quantiles that approximate the distribution of traffic density at time point $t + n$. In this case, n represents the lag for the prediction.

In addition to demonstrating that the baseline MQRNN is effective for traffic density forecasting, we extend the model to create the MQRNN-monotonic. The main innovation proposed is a modification to the MQRNN framework that prevents quantile crossings by enforcing monotonicity of the cumulative distribution function (CDF). This change is described in detail in Section 4.

This type of encoder-decoder sequence-to-sequence architecture allows for time-series inputs of various sizes, as the encoder maps the time series to a fixed-size hidden state [71]. As more information becomes available, the model can lengthen the time series used for prediction. This type of architecture also accepts vector outputs, allowing the prediction of multiple quantiles or parameters simultaneously. Finally, the decoder admits additional exogenous features. The implementation discussed in this paper incorporates exogenous, time-related variables that enable the model to understand seasonality factors.

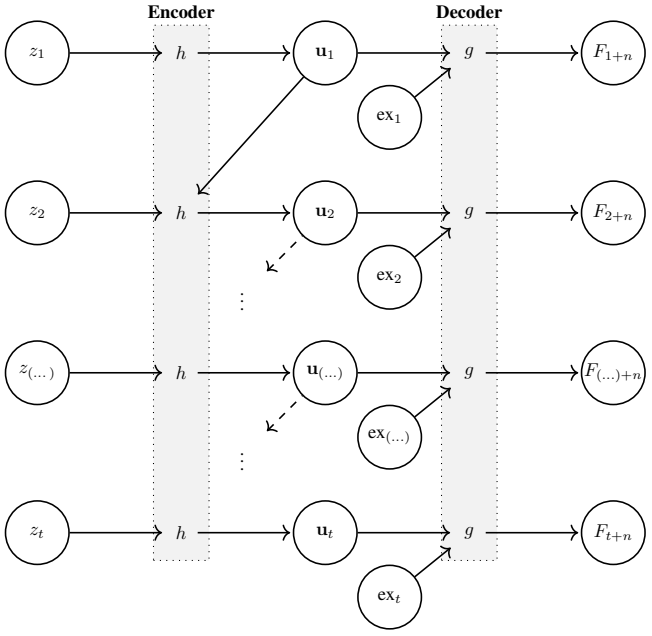


Figure 3. Representation of the probabilistic deep learning model used for traffic density forecasting. The history of traffic density $z_{1:t}$ is encoded to a $\mathbf{u}_t$ of fixed length via the function h. The decoder function g uses this $\mathbf{u}_t$, along with exogenous information ex_t, to predict the distribution of traffic density for time $t + n$, F_{t+n}.

The primary disadvantage of this architecture is that it assumes each time series is one contiguous series. However, traffic sensors frequently malfunction for days or months, resulting in long segments of missing data – we see this behavior in the two data sets we study. While sequence-to-sequence model implementations can handle short missing segments, long missing segments cause a performance drop. In these cases, a more traditional architecture where predictions and losses are evaluated only on the final element of a training segment may be preferred. In this study, we work around this issue by restarting the series whenever a gap occurs.

3.3 Validating probabilistic models

In probabilistic machine learning models, the forecast quality is usually assessed via a scoring rule (Gneiting and Raftery [87]). The rule is proper if the forecaster maximizes the expected score for an observation drawn from a distribution F if they issue the probabilistic forecast F rather than some other arbitrary distribution $G \neq F$. One well-known proper scoring rule is the negative log-likelihood (NLL):

$$\mathrm{NLL}(f, x) = -\log(f(x)),$$

where f is the density function. The NLL is a consistent standard throughout statistical literature; it is proper, but it lacks robustness (Selten [88] and [87]).

Moreover, the density function cannot be computed without intermediate calculations for quantile models, which

makes the NLL inconvenient. In these cases, one can minimize a particular surrogate loss function in hopes that the performance metric is optimized [71]. However, the choice for a surrogate function related to NLL is not immediately apparent.

Instead, the continuous ranked probability score (CRPS) (Matheson and Winkler [89] and Hersbach [90]):

$$\mathrm{CRPS}(F, x) = -\int_{-\infty}^{\infty} \left(F(y) - \mathbb{1}(y \geq x)\right)^2 dy,$$

where F is the cumulative density function, is a particularly popular alternative. The CRPS integrates the squared differences between the predicted CDF and the Heaviside function (a step function that represents the observed outcome) over the entire range of possible outcomes. This results in a score that captures both the calibration and sharpness of the probabilistic forecast, essentially quantifying the "distance" between the forecast distribution and the underlying distribution of the observations. Calibration refers to how often the true outcomes are within the range predicted by the model. A well-calibrated model will have its predicted probabilities match the true frequencies of outcomes over many instances. On the other hand, sharpness gauges the specificity of predictive distributions. A sharper distribution produces tighter prediction intervals, indicating more precise predictions as long as the model remains well-calibrated.

Note that CRPS is a negative function, thus, it is typically used in negative orientation, i.e., $\mathrm{CRPS}^*(F, x) = -\mathrm{CRPS}(F, x)$. Hence, the smaller CRPS^* indicates better probabilistic forecasts. The optimal solution to CRPS^* is the same as for NLL (Morris et al. [91]). This indicates that a forecast that minimizes the CRPS^* will also minimize the NLL, thus underscoring the efficacy of CRPS^* as a scoring rule. Moreover, we argue that the CRPS^*, unlike NLL, has a related surrogate loss function and then explain how this loss function relates to the scoring rule. In the following section, we discuss how the pinball loss function approximates CRPS^*.

3.3.1 Pinball/Tilted Loss Let Γ be a finite set of target percentiles, y_i be the true value, and $f_\theta(\mathbf{x}_i)$ be the predicted value for each data point $i = 1, \ldots, n$. We know that minimizing the ℓ_1-norm loss function produces a median estimator. The symmetry of the median implies that minimizing the ℓ_1-norm loss function produces the same number of positive and negative estimation errors, $(y_i - f_\theta(\mathbf{x}_i))$. By weighing the magnitude of positive and negative errors, Koenker and Bassett Jr [63] propose a loss function for each quantile q_γ, $\gamma \in \Gamma$, that serves as a metric of the accuracy of a quantile forecast. This is known as the pinball (or tilted or quantile) loss function,

$$L^\gamma(y, \hat{y}) \coloneqq \begin{cases} \gamma \cdot (y - \hat{y}) & \text{if } y \geq \hat{y}; \\ (1 - \gamma) \cdot (\hat{y} - y) & \text{if } \hat{y} > y, \end{cases}$$

which can also be written as

$$\begin{aligned} L^\gamma(y, \hat{y}) =& (1 - \gamma) \cdot \max\{0, (\hat{y} - y)\} \\ &+ \gamma \cdot \max\{0, -(\hat{y} - y)\}. \end{aligned}$$

The link between this loss function and quantile regression occurs because it can be shown that

$$q_\gamma = \arg\min_{\hat{y} \in \mathbb{R}} \mathbb{E}[L^\gamma(y, \hat{y})].$$

That is, the empirical risk minimizer for $L^\gamma(y, \hat{y})$ is the $\gamma-$quantile of the response distribution. Furthermore, Grushka-Cockayne et al. [92] show that as Γ (the set of percentiles) grows,

$$c \sum_{\gamma \in \Gamma} L^\gamma(y, F^{-1}(\gamma)) \to \mathrm{CRPS}^*(F, x), \tag{1}$$

with some positive scalar constant c. Hence, for a finite set of quantiles, simply summing the separate loss functions is a sensible approximation to CRPS. For each training step, we add the losses of each quantile to compute the overall loss

$$\ell_{\text{pinb}}(\mathbf{y}, \mathbf{x}) = \sum_{\gamma \in \Gamma} \sum_{i=1}^{n} L^\gamma(y_i, f_\theta(x_i)).$$

A remaining technical issue is that the function ℓ_{pinb} is not smooth, since its gradient has discontinuities at the minimizer of ℓ_{pinb}. Theoretically, this could result in higher variance in weight updates as the model approaches a local minimum, leading to slower training time and potentially worse performance. Empirically, this issue does not seem significant – the median loss $L_{0.5} = \ell_1$-norm is frequently used by practitioners with little trouble. Modern deep learning frameworks provide utilities for learning rate scheduling and model averaging [93], which lessen the burden of a non-smooth loss function. However, some authors [66, 94] choose to use the smoothed version presented by Zheng [95]:

$$\tilde{L}^\gamma(y, f_\theta) = \gamma + \alpha \log\left(1 + \exp\left(-x/\alpha\right)\right),$$

where $\alpha > 0$ is a smoothing parameter. In preliminary experiments, we found that these loss functions showed no significant difference in performance. So, since non-smooth stochastic gradient descent with averaging provides reasonable theoretical and practical performance [96], we opt for the simplicity of the pinball loss function, ℓ_{pinb}.

4 Enforcing monotonicity in probabilistic forecasting

All cumulative distribution functions are non-decreasing. Yet, the probabilistic forecasting framework described above occasionally produces CDFs with decreasing intervals. This problem of *quantile crossings* is seen throughout quantile forecasting literature, with numerous proposed solutions.

The significance of this problem differs by application. Quantile crossing occurrences become more frequent as the number of quantiles increases, making it a challenge when estimating the complete distribution from quantiles (probabilistic forecasting). For applications that require a true CDF (e.g., simulating the conditional distribution using the inverse CDF method), we must meet this non-decreasing requirement.

Solutions often involve placing constraints on the model for problems with parameters selected according to some mathematical program [97, 98]. Unfortunately, enforcing strict constraints on a neural network is typically impractical. Schnabel and Eilers [99] and Rodrigues and Pereira [66] have shown that treating multi-quantile forecasting as a multi-task problem where the neural network outputs each quantile simultaneously will lessen this problem but not do away with it.

We provide a simple modification that works for any quantile regression neural network to enforce monotonicity. Our solution somewhat resembles Gasthaus et al. [100], although without the complexity of constructing a full spline function.

Let f^L be the final hidden layer of the neural network, and consider a vector of the form

$$\Delta = \begin{bmatrix} q_0 \\ \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{n-1} \end{bmatrix} = \begin{bmatrix} \phi\left(\mathbf{w}_0^L f^L + b_0^L\right) \\ \phi\left(\mathbf{w}_1^L f^L + b_1^L\right) \\ \phi\left(\mathbf{w}_2^L f^L + b_2^L\right) \\ \vdots \\ \phi\left(\mathbf{w}_{n-1}^L f^L + b_{n-1}^L\right) \end{bmatrix} = \phi\left(\mathbf{W}^L f^L + \mathbf{b}^L\right).$$

with $\phi : \mathbb{R}^n \to (\mathbb{R}^+)^n$, $\forall\, n \geq 1$ an activation function, and $\mathbf{w}_i^L$ and b_i^L the vectors of weights and the biases, respectively, corresponding to the transition from the last hidden layer to output i. It is important that ϕ be non-negative and unbounded, so reasonable candidates are the ReLU function,

$$\phi(x) = \max\{x, 0\} \equiv x^+,$$

or the softplus function,

$$\phi(x) = \log(1 + \exp(x)).$$

We interpret q_0 to be the lowest quantile and Δ_i to be $q_i - q_{i-1}$ for all $i > 0$. Since $\phi(x) \in [0, +\infty)$, it is naturally enforced that $q_{i-1} \leq q_i$. To reproduce the desired quantiles, we apply the cumulative sum operator

$$q = \text{cumsum}\left(\Delta\right) \equiv \begin{bmatrix} q_0 \\ q_0 + \Delta_1 \\ q_0 + \Delta_1 + \Delta_2 \\ \vdots \\ q_0 + \sum_{j=1}^{n-1} \Delta_j \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_{n-1} \end{bmatrix}.$$

The resulting predictions are non-decreasing and require no additional parameters or constraints. We proceed as before, evaluating q as a quantile forecast using one of the quantile loss functions described above.

It is worth noting that enforcing monotonicity does not necessarily improve the performance metric used to compare the models because most metrics do not verify this constraint. For example, quantile regression is formulated as a quadratic program in the setting of Takeuchi et al. [97]. In their experiments, the global optimum does not necessarily have monotonicity. Moreover, their modification with monotonicity constraints does not consistently outperform the non-monotonic global optimum. This phenomenon of *quantile crossings* occurs because each quantile is estimated separately and is a limitation of the loss function.

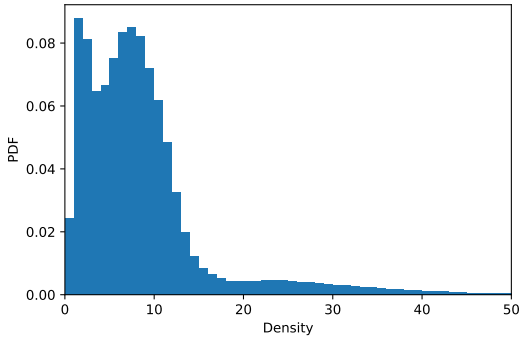
The same basic intuition holds with deep learning models as well. Suppose the model trained to minimize the tilted loss function does not possess a monotonic output. In that case, there is no particular reason to assume that enforcing monotonicity will improve the performance metric. Since we cannot rely on optimality arguments in a deep learning environment, we test this logic empirically using our two datasets.

Regardless of performance, a non-monotonic output that does not have the inherent requirements of a CDF is not sensible. With the inclusion of a monotonicity constraint, the model is thus forced to produce valid distributions.

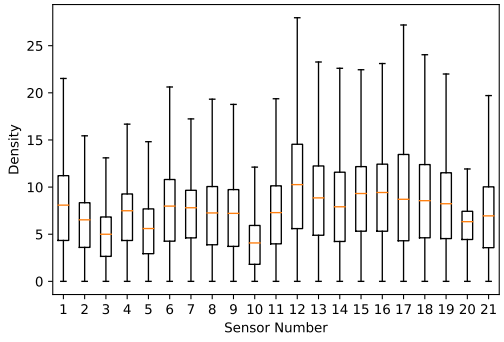
5 Data Description

We compare two data sets describing traffic density. The first covers a section of Interstate 894, located in Milwaukee, WI, and the second covers a section of Interstate I-55, located in Chicago. The data is measured by loop-detector sensors installed on interstate highways. A loop-detector sensor is a simple presence sensor that measures when a vehicle is present and generates an on/off signal. The data contains the timestamp and either traffic density or traffic occupancy. *Occupancy* is defined as the percent of space occupied in a section of the road during a period, and *density* is the number of vehicles occupying the section during a period. Traffic density can be computed by dividing the occupancy by a constant to accommodate the average vehicle length and sensor sensitivity. This constant is called the average field length, or the mean effective vehicle length [101, 102]. Hence, because traffic density is occupancy scaled by a constant factor, the forecasting models can be used in this case for either density or occupancy without the need for any modification. For consistency, we will forecast occupancy for both data sets in this paper.

Interstate I-55 in Chicago has 21 loop-detector sensors that recorded data between 2008 and 2014. The data is displayed in 5-minute intervals, but it contains long periods of missing data and is used in [44]. Figure 4 provides evidence that traffic density is heavy-tailed and right-skewed with a spike on small values. The years 2008–2012 provide data for training, and the remaining years are left as out-of-sample data for testing.

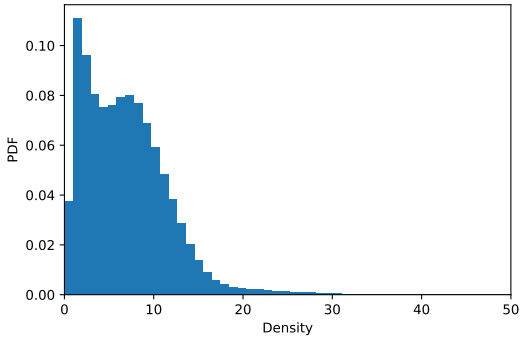


(a) Normalized Histogram of Pooled Traffic Density

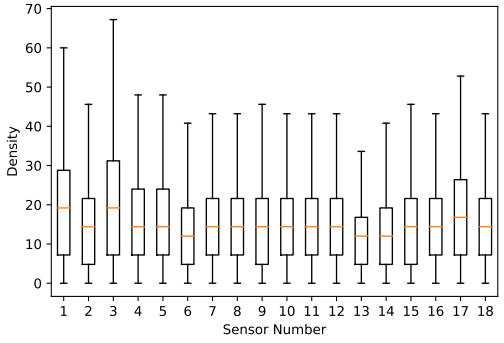


(b) Box-Plots of All Sensors

Figure 4. Traffic Density in Chicago, IL



(a) Normalized Histogram of Pooled Traffic Density



(b) Box-Plots of All Sensors

Figure 5. Traffic Density in Milwaukee, WI

Interstate 894 in Milwaukee contains 18 loop-detector sensors that recorded data in one-minute intervals for 14 months between January 2008 and February 2009. Figure 5 indicates that the density is right-skewed with a spike between zero and two. The figure suggests that heavy congestion is also observed, though more rarely than in Chicago. The recordings are provided in a rounded format because of sensor limitations. We split this dataset into training and testing sets, where 2008 serves as the training set and 2009 as the testing set.

The next section compares the performance of the adapted deep-learning probabilistic model against two traditionally used methods: ARIMA and Gradient Boosting Machine. We set the models to forecast the distribution of traffic occupancy for the next five and fifteen minutes. Still, it can be altered for predictions further in the future with a potential loss in performance. Naturally, we expect models predicting further in the future to be less accurate. However, recent models have had reasonable success in this task [62, 103].

6 Validation

This paper implements a quantile deep learning model, an extension of the MQRNN proposed by Wen et al. [14]. This extension customizes the output to enforce the monotonicity characteristic of quantile forecasting, as discussed in Section 4. We call this extension *MQRNN-monotonic*, and we implement the model using the GluonTS framework [104]. We customize the loss function in the library to ensure CDF monotonicity and use the Bayesian hyperparameter optimization framework *Optuna* (described in [105]) to tune the neural network. The inputs for the models include the most recent contiguous history of traffic density up to the current period, as well as simple temporal information serving as exogenous factors (month, day of the week, and hour).

In order to verify the performance of the proposed models, we implement and evaluate two commonly used models: *naive*, *Auto-Regressive Moving-Average with eXogenous factors* (ARMAX), *Gradient-Boosted Machine* (GBM). The choice of ARMAX is grounded in the literature and in its prevalent presence in the field of traffic forecasting. Its inclusion provides insight into how traditional time series models fare against contemporary machine learning techniques. In contrast, GBMs are recognized for their predictive accuracy and are widely used in various industries for predictive models. To provide a comparison against alternative deep learning methods, we include DeepAR [62] as an additional benchmark model. DeepAR represents a deep learning parametric approach. It outputs parameters of a predefined probability distribution, unlike the MQRNN's non-parametric quantile regression.

The naive model assumes that each sensor measurement is independent and identically distributed (i.i.d.) according to the empirical quantiles. The distribution is, therefore, fixed across time. This is likely not a reasonable assumption because traffic density is not stationary; therefore, it is provided as a simple baseline performance. To generate the naive distributions, we condition on seasonal factors, separating our forecasts based on the hour of the day and the day of the week. To maintain consistency with our other predictive models, which rely on a fixed set of quantiles, we use that same set for the naive distributions. We start by calculating quantiles for each unique combination of weekday and hour. Then, through linear interpolation, we create the naive distributions conditioned on those specific time periods. Finally, we evaluate the CRPS of these distributions by comparing them to actual observations from the test set.

ARIMA models produce point estimates, but quantiles can be obtained assuming that the residuals are normally distributed with constant variance. Vlahogianni and Karlaftis [106] note that traffic occupancy data is often not a unit-root process. This indicates that using a differencing coefficient of $d = 1$ may not improve stationarity conditions for the time series. The authors suggest that fractional differencing may be a more suitable approach for traffic occupancy time series. We tested the data sets from Chicago and Milwaukee for stationarity using the Augmented Dickey-Fuller (ADF) test for daily occupancy. The p-values averaged across sensors for the non-differenced data are 0.00001 and 0.019270, respectively, suggesting that the original data is already stationary. As further validation, we verified that $d \approx 0$ produced the most evidence across the set $0 < d < 1$ for stationarity. Therefore, we consider the ARMA model instead. We include seasonality effects (month, day of the week, and hour) as exogenous regressors. Due to the lack of differencing and the presence of exogenous variables, this model is often referred to as ARMAX (ARMA with eXogenous regressors). We train the regression model using the training data described in Section 5. Then, we use the coefficients to forecast occupancy in the test dataset. We train each sensor separately in this case.

GBMs are ensemble machine learning algorithms that optimize predictive accuracy. Starting with a simple model, a GBM iteratively adds decision trees that correct prior errors. Each tree is guided by the negative gradient of a specified loss function. By combining the predictions of multiple trees, the model's accuracy improves. Over several iterations, these weak learners merge to form a powerful aggregated model. Unlike ARMAX models, GBMs can directly produce quantile results, but each model produces outputs corresponding to a single quantile. We see empirically that shallow trees should be used to prevent overfitting. In this paper, we implement GBMs using the LightGBM framework [107]. For training, we include the last 20 minutes of density. Empirically, longer periods of historical information do not seem to improve performance. We also include the weekday, the hour of the day, and the sensor number as categorical features. GBMs need to be trained for each quantile separately.

DeepAR is a parametric autoregressive model. It uses an RNN to learn temporal dependencies and outputs parameters of a chosen probability distribution for each future time step. For this implementation, we model traffic occupancy using a Gamma distribution. Gamma distributions are suitable for positive, continuous data and are flexible in handling skewed data. The model predicts shape and rate parameters, from which the full predictive distribution is constructed.

The deep learning model (MQRNN-monotonic) can capture multiple quantiles for multiple sensors within a single model, requiring a single training procedure. Hence, deep learning approaches can produce a granular distribution significantly faster than linear models or GBMs.

We validate the models by looking at the calibration and sharpness of the resulting distributions [13]. The main metric utilized for comparing the models is the approximated CRPS^* described in equation 1 because it assesses both calibration and sharpness. A lower CRPS^* indicates a better model quality. In addition, we provide coverage probabilities and a few examples of percentile intervals to illustrate the calibration and sharpness of the models' predictions, respectively. The overall goal is to obtain a sharp model, subject to calibration.

7 Results

We display the resulting CRPS^* for each model in Table 1. These CRPS^* balance the models' calibration and sharpness, and a lower value indicates a better model. The ARMAX model outperformed the Naive baseline in all scenarios but proved less effective than the machine learning approaches, likely due to the limitations of assuming normally distributed residuals for this type of data. Both the proposed MQRNN-monotonic and the parametric DeepAR model significantly outperformed the traditional benchmarks. DeepAR demonstrates competitive performance, with its score in the 5-minute Milwaukee forecast being the best. However, the MQRNN-monotonic model consistently achieves the lowest CRPS* across almost all scenarios and demonstrates particular strength in the 15-minute forecasts.

To further verify that the models with the lowest CRPS^* produce calibrated results, we provide the coverage probability

$$\mathrm{CP}(q_\alpha, q_{1-\alpha}, y) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(q_\alpha \leq y_n \leq q_{1-\alpha}),$$

which has an ideal value of $1 - 2\alpha$ for intervals $(q_\alpha, q_{1-\alpha})$. For example, we expect that 95% of the data falls between the values determined by the percentiles $\alpha = 0.025$ and $1 - \alpha =$

Model	5-minute forecast		15-minute forecast	
	Chicago	Milwaukee	Chicago	Milwaukee
Naive	0.0323	0.0370	0.0323	0.0370
ARMAX	0.0131	0.0190	0.0347	0.0315
GBM	0.0100	0.0153	0.0283	0.0194
DeepAR (Gamma)	0.0130	0.0146	0.0176	0.0195
MQRNN-monotonic	0.0097	0.0148	0.0105	0.0138

Table 1. CRPS* averages depicted here are pooled estimates in the test set from all the sensors

0.975 of the distribution. Because this metric can be assessed at any arbitrary level α, we choose the levels based on Γ: $\alpha = \{0.001, 0.005, 0.025, 0.05, 0.1, 0.25, 0.33, 0.5\}$. This metric helps visualize whether the models' calibration is good, but does not measure sharpness.

We note that coverage probabilities should not be used as a performance metric but simply to verify whether the outputs are reasonable. By definition, the naive model will provide perfect coverage of the training set. For the naive model, any performance discrepancy stems from the discrepancy between train and test performance. Since the data sets are not i.i.d., but time-dependent, the naive model will be over-dispersed and only correct when controlled over time. Controlling over time would ensure that seasonality factors and autocorrelation effects are not present. The naive model can be thought of as accurate but not precise. Proper scoring methods, such as CPRS, penalize this behavior. Overly or underly-dispersed conditional distributions are sub-optimal, and a more flexible model may provide sharper forecasts when conditioned on the process history. Hence, models with better coverage probability may not necessarily indicate better performance.

In contrast, it is possible that optimizing over the distributions conditioned on time may come at the cost of systemic under- or over-dispersal of the marginal distribution (the distribution obtained with the naive method). This is particularly evident in the Milwaukee data set, where the source data has been rounded. In this dataset, the deep learning models overfit to capture the resulting discrete distributions, resulting in poor coverage probabilities. This problem can be potentially remedied through further hyperparameter optimization; however, it presents a noteworthy practical gap in probabilistic forecasting with deep learning models. In summary, as long as the coverage probabilities yield sensible results, the CRPS provides a more accurate picture of which models have the highest performance.

Table 2 displays a subset of the coverage probabilities for each implemented model. Because of sensor limitations, Milwaukee data is recorded in discrete increments. This discretization affects the deep learning models differently. The MQRNN-monotonic model appears to overfit to these discrete steps, resulting in extremely sharp (narrow) prediction intervals that lead to undercoverage, particularly for the 80% interval. The DeepAR model, which assumes a continuous Gamma distribution, struggles to align this parametric form with the discrete, rounded nature of the Milwaukee data. This mismatch likely contributes to its poor calibration, especially the under-coverage seen in the 95% prediction interval for the 15-minute forecast.

For the Chicago data, which contains continuous measurements, DeepAR tends to produce overly-dispersed distributions, with coverage probabilities often far exceeding the target levels (e.g., 97.1% coverage for an 80% target interval), suggesting its distributions are not sharp. MQRNN-monotonic shows fewer calibration issues, and its superior CRPS* scores indicate that its sharper predictions provide a better overall balance of calibration and sharpness compared to the other models.

7.1 Out-of-the-ordinary Events

The literature has notably struggled to forecast traffic accurately when roadway behavior differs from expected. When traffic faces external factors that are not seasonal, either expected (e.g., sports matches, festival parades, or roadworks) or unexpected (e.g., snowstorms or accidents), traditional forecasting models decrease significantly in accuracy. To measure the proposed model's performance in such situations, we also tested it against the benchmarks when the experienced traffic occupancy was at least three standard deviations from the median within the same day of the week and the same hour of the day.

The results are shown in Table 3. As expected, we see that the CRPS^* values are overall greater than the ones in Table 1. The ARMAX model's performance dropped significantly, exemplifying its common struggle to capture sudden behavioral shifts. In contrast, the deep learning models demonstrate superior performance, achieving a consistently lower CRPS^*.

For the Chicago data, MQRNN-monotonic is the top-performing model. In the Milwaukee dataset, however, the DeepAR model achieves a lower CRPS^* during these outlier events. This suggests that while the Gamma distribution assumed by DeepAR may be a suboptimal fit for the rounded Milwaukee data overall, its parametric shape may be more robust for capturing the specific tail behavior of these non-recurrent events. Nevertheless,

Model	Chicago (%)	Milwaukee (%)	Chicago (%)	Milwaukee (%)
	5-minute forecast		15-minute forecast	
Target	80.0 / 95.0	80.0 / 95.0	80.0 / 95.0	80.0 / 95.0
Naive	79.6 / 94.3	83.1 / 92.4	79.6 / 94.3	83.1 / 92.4
ARMAX	92.2 / 96.1	81.6 / 90.6	88.6 / 94.4	81.8 / 90.7
GBM	79.6 / 94.7	78.9 / 95.5	77.7 / 93.4	77.5 / 94.7
Deep AR (Gamma)	97.3 / 98.3	89.7 / 92.9	97.1 / 98.3	90.6 / 92.9
MQRNN-monotonic	79.7 / 89.1	67.3 / 98.0	73.4 / 85.2	73.2 / 98.0

Table 2. 80% / 95% coverage probabilities averages depicted here are pooled estimates from all the sensors

considering its strong performance across both datasets and its superior accuracy in typical conditions, the MQRNN-monotonic model demonstrates the most consistent and reliable performance overall.

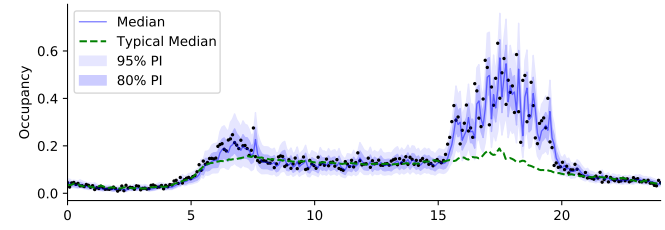
To visualize these performance differences, Figure 6 illustrates the predictive distributions of different models during the two non-recurrent events introduced in Section 1. The shaded areas represent the 80% and 95% prediction intervals.

During the Chicago Bears game, the MQRNN-monotonic model (Figure 6a) produces a sharp 95% prediction interval that accurately contains the observed traffic surge. In contrast, the GBM model (Figure 6b) generates much wider intervals, indicating greater uncertainty during this unexpected event. A similar pattern emerges for the Milwaukee snow day. The MQRNN-monotonic model (Figure 6c) remains sharp and well-calibrated, while the ARMAX model (Figure 6d) produces overly wide prediction intervals that reflect its struggle to adapt. These examples provide visual evidence that the proposed model achieves superior sharpness without sacrificing calibration, allowing it to precisely capture traffic dynamics even during significant, non-recurrent disruptions.

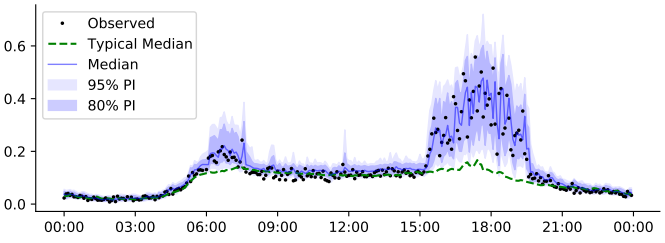
8 Decision-Making Example

Building on the predictive accuracy demonstrated in the previous sections, we now explore a practical application of the probabilistic prediction model. By contextualizing the decision-making process within a real-world scenario, we aim to demonstrate its utility in informing cost-benefit analyses for traffic management strategies.

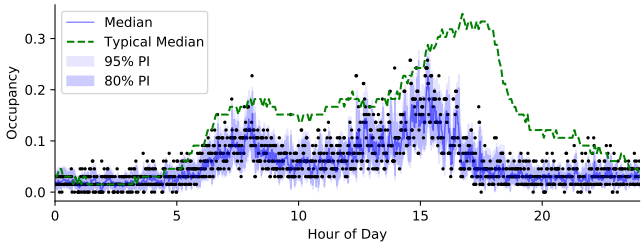
Let us assume a highway authority is responsible for a stretch of highway prone to congestion issues represented in Figure 6a. They need to decide when to ease congestion by implementing variable speed limits, ramp metering, or using the hard shoulder as a regular lane in response to the Bears' game. Implementing the suite of congestion mitigation measures costs the highway authority $2,000. They estimate that moderate congestion (110 to 150 cars/0.5 mile) results in a cost of $2,000 (due to delays, fuel, etc.). Severe congestion (above 150 cars/0.5 mile) costs $10,000 (higher delays, increased chance of accidents).



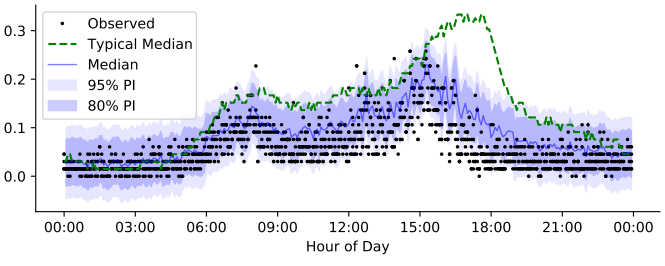
(a) Bear's Game (MQRNN - monotonic)



(b) Bear's Game (GBM)



(c) Snow Day (MQRNN - monotonic)



(d) Snow Day (ARMAX)

Figure 6. Predictions during out-of-the-ordinary events

Model	5-minute forecast		15-minute forecast	
	Chicago	Milwaukee	Chicago	Milwaukee
Naive	0.0540	0.0451	0.0540	0.0451
ARMAX	0.0407	0.0845	0.0951	0.0850
GBM	0.0333	0.0274	0.0355	0.0338
Deep AR (Gamma)	0.0149	0.0155	0.0199	0.0200
MQRNN-monotonic	0.0103	0.0227	0.0099	0.0221

Table 3. Outlier (> 3 standard deviations from the median) CRPS^* test averages. They are pooled estimates from all the sensors.

Suppose we are in the scenario displayed in Figure 6a and the deep-learning model predicts quantiles for traffic density (cars/0.5-mile stretch) at 18:00 pm ahead of the game. Because the model outputs occupancy, we must transform the predictions to traffic density. We multiply the predicted occupancy by the sensor length segment (0.5 miles) and divide by the average car length plus headway of 22 ft, obtained through the procedure described in [108, 35]. The predictions are in table 4.

Quantiles	Cars / 0.5-mi	Quantiles	Cars / 0.5-mi
0.001	5	0.670	118
0.005	22	0.750	133
0.025	55	0.900	150
0.050	80	0.950	164
0.100	103	0.975	190
0.250	105	0.995	206
0.330	111	0.999	229
0.500	115	-	-

Table 4. Quantile distribution of cars per 0.5 mile.

In this scenario, there is a 90% chance the traffic density will fall between 83 and 177 cars/0.5 mile in the next 15 minutes. There is a 33% chance of very light traffic (fewer than 111 cars/0.5 mile) and a 10% chance of severe congestion (more than 149 cars/0.5 mile).

In considering the deployment of congestion mitigation measures, the authority must weigh the model's predictive uncertainty against potential societal costs. For instance, a false alarm—predicting high congestion that does not materialize—could lead to unnecessary expenditure and public inconvenience. Conversely, inaction in the face of a true high congestion prediction could result in significant economic losses and safety risks. The highway authority can then weigh its decisions by comparing the approximated expected cost of doing nothing versus the cost of the congestion reduction actions.

$$\begin{aligned}\mathbb{E}[\text{cost}] &= 2000 \cdot P(110 \leq X < 150) + 10000 \cdot P(X \geq 150) \\ &= 2000 \cdot (0.9 - 0.33) + 10000 \cdot (1 - 0.9) = 2,140.\end{aligned}$$

As the expected cost ($2,140) of congestion is higher than the cost of implementing the congestion mitigation actions ($2,000), the highway authority decides to implement them in the next 15 minutes.

This decision-making example underscores the practical value of integrating advanced predictive models into traffic management systems. By translating probabilistic predictions into actionable insights, authorities can make more informed decisions that strike a balance between efficiency, cost, and public welfare.

8.1 The Problem of Quantile Crossings

We would like to provide some clarity on why enforcing the monotonicity of the distribution and preventing quantile crossing is necessary to ensure valid distributions and lead to the correct decisions. Suppose the quantile predictions from table 4 remain the same except for this crossing:

$$P(X \leq 110) = 0.5 \text{ and } P(X \leq 115) = 0.33.$$

This crossing indicates an inconsistency. There cannot be a lower chance of seeing 110 cars or fewer compared to 115 cars or fewer. The probability of seeing fewer than 110 cars must always be less than the probability of seeing fewer than 115. Moreover, this error highlights how misinterpreting quantile crossings can lead to incorrect predictions and potentially miscalculated congestion costs. Note, for example, the newly calculated expected cost:

$$\begin{aligned}\mathbb{E}[\text{cost}] &= 2000 \cdot P(110 \leq X < 150) + 10000 \cdot P(X \geq 150) \\ &= 2000 \cdot (0.9 - 0.5) + 10000 \cdot (1 - 0.9) = 1800.\end{aligned}$$

In this case, the expected cost of congestion is $1,800, lower than the cost of implementing the mitigation actions. Therefore, the highway authority would wrongly decide not to intervene.

9 Conclusion

This study implements and compares several methods for estimating the transient distribution of traffic density using two distinct data sets. It implements an adapted sequence-to-sequence encoder-decoder deep learning model that enforces monotonicity constraints for quantile forecasts. Hence, the proposed MQRNN-monotonic model avoids the quantile-crossing issue and produces cumulative distributions guaranteed to be monotonically non-decreasing.

We observe that a deep learning model can considerably outperform traditional models when predicting traffic density

distributions directly. Specifically, our proposed MQRNN-monotonic model showed a superior balance of calibration and sharpness, surpassing not only traditional models but also DeepAR, our benchmark parametric deep learning forecaster.

We show that the deep learning models' structure successfully forecasts the distribution of traffic congestion and can capture non-recurrent events efficiently. Notably, the resulting distributions remain well-calibrated while being much sharper, thus producing tighter percentile intervals for out-of-the-ordinary predictions than the benchmarks. Furthermore, unlike many current models, we verify that solid performance can be achieved with minimal data preprocessing, even on datasets with distinct data problems.

The MQRNN-monotonic model is semi-parametric, and its forecasts do not have immediately explainable features. In the future, we suggest further investigating which features play the most critical roles in determining the forecast.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: all authors; data collection: Baykal-Gursoy, M and Lopes Gerum, P C; analysis and interpretation of results: Benton, A and Lopes Gerum, P C; draft manuscript preparation: all authors. All authors reviewed the results and approved the final version of the manuscript.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data Accessibility Statement

Part of the data and the code utilized in this manuscript are available at `https://github.com/pedrogerum/probabilistic-traffic-forecasting.git`.

Funding

The authors disclosed no financial support for the research, authorship, and/or publication of this article.

Acknowledgements

LLM usage: Claude (Anthropic) and Gemini (Google) were used solely for proofreading. All code was developed prior to the widespread availability of LLMs. No conceptual or methodological content involved LLM assistance.

References

[1] European Commission. *European Urban Mobility: policy Context.* 2017.

[2] Kim, J., Mahmassani, H., and Dong, J. "Likelihood and Duration of Flow Breakdown". *Transportation Research Record: Journal of the Transportation Research Board* 2188.-1, 2010, pp. 19–28. ISSN: 0361-1981.

[3] Miller-Hooks, E. and Sorrel, G. "Maximal dynamic expected flows problem for emergency evacuation planning". *Transportation Research Record: Journal of the Transportation Research Board* 2089.-1, 2008, pp. 26–34. ISSN: 0361-1981.

[4] Bharadwaj, N., Kumar, P., Mane, A. S., Arkatkar, S. S., Bhaskar, A., and Joshi, G. J. "Comparative evaluation of density estimation methods on different uninterrupted roadway facilities: Few case studies in India". *Transportation in developing economies* 3.1, 2017, p. 3.

[5] Daganzo, C. F. "The Cell Transmission Model: A dynamic representation of highway traffic consistent with the hydrodynamic theory". *Transportation Research Part B, Methodological* 28.4, 1994, pp. 269–287.

[6] Nagel, K. and Schreckenberg, M. "A cellular automaton model for freeway traffic". *Journal de Physique I* 2.12, 1992, pp. 2221–2229.

[7] Zechin, D. and Cybis, H. B. B. "Probabilistic traffic breakdown forecasting through Bayesian approximation using variational LSTMs". *Transportmetrica B: Transport Dynamics* 11.1, 2023, pp. 1026–1044.

[8] Arnesen, P. and Hjelkrem, O. A. "An Estimator for Traffic Breakdown Probability Based on Classification of Transitional Breakdown Events". *Transportation Science* 52.3, 2018, pp. 593–602. DOI: 10.1287/trsc.2017.0776. URL: https://doi.org/10.1287/trsc.2017.0776.

[9] Han, Y. and Ahn, S. "Stochastic modeling of breakdown at freeway merge bottleneck and traffic control method using connected automated vehicle". *Transportation Research Part B: Methodological* 107, 2018, pp. 146–166.

[10] Lombardi, C., Picado-Santos, L., and Annaswamy, A. M. "Model-based dynamic toll pricing: An overview". *Applied Sciences* 11.11, 2021, p. 4778.

[11] Ye, F.-F., Yang, L.-H., Wang, Y.-M., and Lu, H. "A data-driven rule-based system for China's traffic accident prediction by considering the improvement of safety efficiency". *Computers & Industrial Engineering* 176, 2023, p. 108924.

[12] Petelin, G., Hribar, R., and Papa, G. "Models for Forecasting the Traffic Density within the City of Ljubljana". *Available at SSRN 4233939*, 2022.

[13] Gneiting, T. and Katzfuss, M. "Probabilistic forecasting". *Annual Review of Statistics and its Application* 1, 2014, pp. 125–151.

[14] Wen, R., Torkkola, K., Narayanaswamy, B., and Madeka, D. "A multi-horizon quantile recurrent forecaster". *arXiv preprint arXiv:1711.11053*, 2017.

[15] Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., and Dubrawski, A. "Nhits: Neural hierarchical interpolation for time series forecasting". *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 6, 2023, pp. 6989–6997.

[16] Zeng, A., Chen, M., Zhang, L., and Xu, Q. "Are transformers effective for time series forecasting?" *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 9, 2023, pp. 11121–11128.

[17] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. "Short-term traffic forecasting: Where we are and where we're going". *Transportation Research Part C: Emerging Technologies* 43, 2014, pp. 3–19.

[18] Kurzhanskiy, A. A. "Set-valued estimation of freeway traffic density". *IFAC Proceedings Volumes* 42.15, 2009, pp. 271–277.

[19] Chrobok, R., Kaumann, O., Wahle, J., and Schreckenberg, M. "Different methods of traffic forecast based on real data". *European Journal of Operational Research* 155.3, 2004, pp. 558–568.

[20] Zhang, J., Yu, Y., and Lei, Y. "The study on an optimized model of traffic congestion problem caused by traffic accidents". *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 2016, pp. 688–692.

[21] Zhang, X., Onieva, E., Perallos, A., Osaba, E., and Lee, V. C. "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction". *Transportation Research Part C: Emerging Technologies* 43, 2014, pp. 127–142.

[22] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[23] Chan, W. and Tong, H. "On tests for non-linearity in time series analysis". *Journal of Forecasting* 5.4, 1986, pp. 217–228.

[24] Bollerslev, T. "Generalized autoregressive conditional heteroskedasticity". *Journal of Econometrics* 31.3, 1986, pp. 307–327.

[25] Ghosh, B., Basu, B., and O'Mahony, M. "Bayesian time-series model for short-term traffic flow forecasting". *Journal of transportation engineering* 133.3, 2007, pp. 180–189.

[26] Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, A, Kiartzis, S, Panagiotou, H, and Maslaris, N. "A Bayesian multiple models combination method for time series prediction". *Journal of intelligent and robotic systems* 31.1, 2001, pp. 69–89.

[27] Wang, J., Deng, W., and Guo, Y. "New Bayesian combination method for short-term traffic flow forecasting". *Transportation Research Part C: Emerging Technologies* 43, 2014, pp. 79–94.

[28] Gu, Y., Lu, W., Xu, X., Qin, L., Shao, Z., and Zhang, H. "An improved Bayesian combination model for short-term traffic prediction with deep learning". *IEEE Transactions on Intelligent Transportation Systems* 21.3, 2019, pp. 1332–1342.

[29] Vlahogianni, E. I. "Enhancing predictions in signalized arterials with information on short-term traffic flow dynamics". *Journal of Intelligent Transportation Systems* 13.2, 2009, pp. 73–84.

[30] Kamarianakis, Y., Gao, H. O., and Prastacos, P. "Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions". *Transportation Research Part C: Emerging Technologies* 18.5, 2010, pp. 821–840.

[31] Kwon, J., Barkley, T., Hranac, R., Petty, K., and Compin, N. "Decomposition of travel time reliability into various sources: incidents, weather, work zones, special events, and base capacity". *Transportation Research Record* 2229.1, 2011, pp. 28–33.

[32] Xia, J., Chen, M., and Qian, Z. "Predicting Freeway Travel Time Under Incident Conditions". *Transportation Research Record: Journal of the Transportation Research Board* 2178.-1, 2010, pp. 58–66. ISSN: 0361-1981.

[33] Fei, X., Lu, C.-C., and Liu, K. "A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction". *Transportation Research Part C: Emerging Technologies* 19.6, 2011, pp. 1306–1318.

[34] Min, W. and Wynter, L. "Real-time road traffic prediction with spatio-temporal correlations". *Transportation Research Part C: Emerging Technologies* 19.4, 2011, pp. 606–616.

[35] Gerum, P. C. L., Benton, A. R., and Baykal-Gürsoy, M. "Traffic density on corridors subject to incidents: models for long-term congestion management". *EURO Journal on Transportation and Logistics* 8.5, 2019, pp. 795–831.

[36] Gerum, P. C. L. and Baykal-Gürsoy, M. "How incidents impact congestion on roadways: A queuing network approach". *EURO Journal on Transportation and Logistics* 11, 2022, p. 100067.

[37] Baykal-Gürsoy, M. and Xiao, W. "Stochastic decomposition in M/M/∞ queues with Markov modulated service rates". *Queueing Systems* 48.1-2, 2004, pp. 75–88. ISSN: 02570130. DOI: 10.1023/B:QUES.0000039888.52119.1d.

[38] Baykal-Gürsoy, M., Xiao, W., and Ozbay, K. M. A. "Modeling Traffic Flow Interrupted by Incidents". *European Journal of Operational Research* 195, 2009, pp. 127–138.

[39] Baykal-Gürsoy, M., Benton, A. R., Gerum, P. C. L., and Candia, M. F. "How Random Incidents Affect Travel-Time Distributions". *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[40] Gao, J., Zuo, F., Ozbay, K., Hammami, O., and Barlas, M. L. "A new curb lane monitoring and illegal parking impact estimation approach based on queueing theory and computer vision for cameras with low resolution and low frame rate". *Transportation Research Part A: Policy and Practice* 162, 2022, pp. 137–154.

[41] Dougherty, M. S., Kirby, H. R., and Boyle, R. D. "The use of neural networks to recognise and predict traffic congestion". *Traffic Engineering & Control* 34.6, 1993.

[42] Dia, H. "An object-oriented neural network approach to short-term traffic forecasting". *European Journal of Operational Research* 131.2, 2001, pp. 253–261.

[43] Zhu, J., Sun, K., Jia, S., Li, Q., Hou, X., Lin, W., Liu, B., and Qiu, G. "Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network". *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.12, 2018, pp. 4968–4981.

[44] Polson, N. and Sokolov, V. "Deep learning for short-term traffic flow prediction". *Transportation Research Part C: Emerging Technologies* 79, 2017, pp. 1–17.

[45] Zhao, Z., Chen, W., Wu, X., Chen, P. C., and Liu, J. "LSTM network: a deep learning approach for short-term traffic forecast". *IET Intelligent Transport Systems* 11.2, 2017, pp. 68–75.

[46] Zhong, Y., Xie, X., Guo, J., Wang, Q., and Ge, S. "A new method for short-term traffic congestion forecasting based on LSTM". *IOP Conference Series: Materials Science and Engineering*. Vol. 383. 1. IOP Publishing, 2018, p. 012043.

[47] Chen, M., Yu, X., and Liu, Y. "PCNN: deep convolutional networks for short-term traffic congestion prediction". *IEEE Transactions on Intelligent Transportation Systems* 19.11, 2018, pp. 3550–3559.

[48] Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., Albeshri, A., and Center, H. P. C. "A deep learning model to predict vehicles occupancy on freeways for traffic management". *IJCSNS* 18.12, 2018, p. 1.

[49] Yao, R., Zhang, W., and Zhang, L. "Hybrid methods for short-term traffic flow prediction based on ARIMA-GARCH model and wavelet neural network". *Journal of Transportation Engineering, Part A: Systems* 146.8, 2020, p. 04020086.

[50] Chinthakunta, S., Sunkavalli, J. P., and Koduru, P. "Deep Learning–Based Traffic Density Estimation And Analysis". *2025 Global Conference in Emerging Technology (GINOTECH)*. IEEE, 2025, pp. 1–6.

[51] Ismaeel, A. G., Janardhanan, K., Sankar, M., Natarajan, Y., Mahmood, S. N., Alani, S., and Shather, A. H. "Traffic pattern classification in smart cities using deep recurrent neural network". *Sustainability* 15.19, 2023, p. 14522.

[52] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. "Spatial-temporal transformer networks for traffic flow forecasting". *arXiv preprint arXiv:2001.02908*, 2020.

[53] Sharma, A., Sharma, A., Nikashina, P., Gavrilenko, V., Tselykh, A., Bozhenyuk, A., Masud, M., and Meshref, H. "A graph neural network (GNN)-based approach for real-time estimation of traffic speed in sustainable smart cities". *Sustainability* 15.15, 2023, p. 11893.

[54] Wilkman, D., Morozovska, K., Johansson, K. H., and Barreau, M. "Online traffic density estimation using physics-informed neural networks". *arXiv preprint arXiv:2504.03483*, 2025.

[55] Barros, J., Araujo, M., and Rossetti, R. J. "Short-term real-time traffic prediction methods: A survey". *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2015, pp. 132–139.

[56] Do, L. N., Taherifar, N., and Vu, H. L. "Survey of neural network-based models for short-term traffic state prediction". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.1, 2019, e1285.

[57] Lee, K., Eo, M., Jung, E., Yoon, Y., and Rhee, W. "Short-term traffic prediction with deep neural networks: A survey". *IEEE Access* 9, 2021, pp. 54739–54756.

[58] Hou, Y., Zheng, X., Han, C., Wei, W., Scherer, R., and Połap, D. "Deep Learning Methods in Short-Term Traffic Prediction: A Survey". *Information Technology and Control* 51.1, 2022, pp. 139–157.

[59] Soon, K. L., Chan, R. K. C., Lim, J. M.-Y., and Parthiban, R. "Short-term traffic forecasting model: prevailing trends and guidelines". *Transportation safety and environment* 5.3, 2023, tdac058.

[60] Dong, J. and Mahmassani, H. S. "Stochastic modeling of traffic flow breakdown phenomenon: Application to predicting travel time reliability". *IEEE Transactions on Intelligent Transportation Systems* 13.4, 2012, pp. 1803–1809.

[61] Wang, Z., Tian, J., Jiang, R., Li, X., and Ma, S. F. "Car following model simulating traffic breakdown and concave growth pattern of oscillations in traffic flow". *arXiv preprint arXiv:1703.10378*, 2017.

[62] Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. "DeepAR: probabilistic forecasting with autoregressive recurrent networks". *International Journal of Forecasting* 36.3, 2020, pp. 1181–1191.

[63] Koenker, R. and Bassett Jr, G. "Regression quantiles". *Econometrica: Journal of the Econometric Society*, 1978, pp. 33–50.

[64] Meinshausen, N. and Ridgeway, G. "Quantile regression forests." *Journal of Machine Learning Research* 7.6, 2006.

[65] Landry, M., Erlinger, T. P., Patschke, D., and Varrichio, C. "Probabilistic gradient boosting machines for GEFCom2014 wind forecasting". *International Journal of Forecasting* 32.3, 2016, pp. 1061–1066.

[66] Rodrigues, F. and Pereira, F. C. "Beyond expectation: deep joint mean and quantile regression for spatiotemporal problems". *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[67] Tang, B. and Matteson, D. S. "Probabilistic transformer for time series analysis". *Advances in Neural Information Processing Systems* 34, 2021, pp. 23592–23608.

[68] Polson, N and Sokolov, V. "Deep learning predictors for traffic flows". *arXiv preprint arXiv:1604.04527*, 2016.

[69] Yang, H., Zhang, X., Li, Z., and Cui, J. "Region-Level Traffic Prediction Based on Temporal Multi-Spatial Dependence Graph Convolutional Network from GPS Data". *Remote Sensing* 14.2, 2022, p. 303.

[70] Yoon, T., Park, Y., Ryu, E. K., and Wang, Y. "Robust probabilistic time series forecasting". *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 1336–1358.

[71] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep Learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[72] Cybenko, G. "Approximation by superpositions of a sigmoidal function". *Mathematics of Control, Signals and Systems* 2.4, 1989, pp. 303–314.

[73] Hornik, K., Stinchcombe, M., White, H., et al. "Multilayer feedforward networks are universal approximators." *Neural Networks* 2.5, 1989, pp. 359–366.

[74] Park, S., Yun, C., Lee, J., and Shin, J. "Minimum width for universal approximation". *arXiv preprint arXiv:2006.08859*, 2020.

[75] Bottou, L., Curtis, F. E., and Nocedal, J. "Optimization methods for large-scale machine learning". *SIAM Review* 60.2, 2018, pp. 223–311.

[76] Ruder, S. "An overview of gradient descent optimization algorithms". *arXiv preprint arXiv:1609.04747*, 2016.

[77] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. "Learning representations by back-propagating errors". *nature* 323.6088, 1986, pp. 533–536.

[78] Hochreiter, S. and Schmidhuber, J. "Long short-term memory". *Neural Computation* 9.8, 1997, pp. 1735–1780.

[79] Hewamalage, H., Bergmeir, C., and Bandara, K. "Recurrent neural networks for time series forecasting: Current status and future directions". *International Journal of Forecasting* 37.1, 2021, pp. 388–427.

[80] Sutskever, I., Vinyals, O., and Le, Q. V. "Sequence to sequence learning with neural networks". *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[81] Chen, Y., Kang, Y., Chen, Y., and Wang, Z. "Probabilistic forecasting with temporal convolutional neural network". *Neurocomputing* 399, 2020, pp. 491–501.

[82] Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. "Time-series extreme event forecasting with neural networks at Uber". *International Conference on Machine Learning*. Vol. 34, 2017, pp. 1–5.

[83] Shen, Z., Zhang, Y., Lu, J., Xu, J., and Xiao, G. "A novel time series forecasting model with deep learning". *Neurocomputing* 396, 2020, pp. 302–313.

[84] Chung, J., Gülçehre, C., Cho, K., and Bengio, Y. "Empirical evaluation of gated recurrent neural networks on sequence modeling". *arXiv preprint arXiv:1412.3555*, 2014.

[85] Cho, K., Van Merriënboer, B., Gülçehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". *arXiv preprint arXiv:1406.1078*, 2014.

[86] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. "Convolutional sequence to sequence learning". *International Conference on Machine Learning*. PMLR, 2017, pp. 1243–1252.

[87] Gneiting, T. and Raftery, A. E. "Strictly proper scoring rules, prediction, and estimation". *Journal of the American statistical Association* 102.477, 2007, pp. 359–378.

[88] Selten, R. "Axiomatic characterization of the quadratic scoring rule". *Experimental Economics* 1, 1998, pp. 43–61.

[89] Matheson, J. E. and Winkler, R. L. "Scoring rules for continuous probability distributions". *Management science* 22.10, 1976, pp. 1087–1096.

[90] Hersbach, H. "Decomposition of the continuous ranked probability score for ensemble prediction systems". *Weather and Forecasting* 15.5, 2000, pp. 559–570.

[91] Morris, M., Hayes, P., Cox, I. J., and Lampos, V. "Estimating the Uncertainty of Neural Network Forecasts for Influenza Prevalence Using Web Search Activity". *arXiv preprint arXiv:2105.12433*, 2021.

[92] Grushka-Cockayne, Y., Lichtendahl Jr, K. C., Jose, V. R. R., and Winkler, R. L. "Quantile evaluation, sensitivity to bracketing, and sharing business payoffs". *Operations Research* 65.3, 2017, pp. 712–728.

[93] Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. "Averaging weights leads to wider optima and better generalization". *arXiv preprint arXiv:1803.05407*, 2018.

[94] Hatalis, K., Lamadrid, A. J., Scheinberg, K., and Kishore, S. "Smooth pinball neural network for probabilistic forecasting of wind power". *arXiv preprint arXiv:1710.01720*, 2017.

[95] Zheng, S. "Gradient descent algorithms for quantile regression with smooth approximation". *International Journal of Machine Learning and Cybernetics* 2.3, 2011, pp. 191–207.

[96] Shamir, O. and Zhang, T. "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes". *International Conference on Machine Learning*. PMLR, 2013, pp. 71–79.

[97] Takeuchi, I., Le, Q., Sears, T., Smola, A., et al. "Nonparametric quantile estimation", 2006.

[98] Sangnier, M., Fercoq, O., and Buc, F. d'Alché. "Joint quantile regression in vector-valued RKHSs". *Neural Information Processing Systems*, 2016.

[99] Schnabel, S. K. and Eilers, P. H. "Simultaneous estimation of quantile curves using quantile sheets". *AStA Advances in Statistical Analysis* 97.1, 2013, pp. 77–87.

[100] Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S. S., Salinas, D., Flunkert, V., and Januschowski, T. "Probabilistic forecasting with spline quantile function RNNs". *The*

22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019, pp. 1901–1910.

[101] Mn/DOT. *Minnesota Department of Transportation*, 2020.

[102] Papageorgiou, M. and Vigos, G. "Relating time-occupancy measurements to space-occupancy and link vehicle-count". *Transportation Research Part C: Emerging Technologies* 16.1, 2008, pp. 1–17.

[103] Makridakis, S, Spiliotis, E, and Assimakopoulos, V. "The M5 accuracy competition: Results, findings and conclusions". *International Journal of Forecasting*, 2020.

[104] Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. "Gluonts: Probabilistic time series models in python". *arXiv preprint arXiv:1906.05264*, 2019.

[105] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. "Optuna: A next-generation hyperparameter optimization framework". *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[106] Vlahogianni, E. and Karlaftis, M. "Temporal aggregation in traffic data: implications for statistical characteristics and model choice". *Transportation Letters* 3.1, 2011, pp. 37–49.

[107] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. "Lightgbm: A highly efficient gradient boosting decision tree". *Advances in Neural Information Processing Systems* 30, 2017, pp. 3146–3154.

[108] Dailey, D. J. "A statistical algorithm for estimating speed from single loop volume and occupancy measurements". *Transportation Research Part B: Methodological* 33.5, 1999, pp. 313–322.