

## Article

# Why Uncertainty in Deep Learning for Traffic Flow Prediction Is Needed

Mingyu Kim<sup>1</sup> and Donghyun Lee<sup>2,\*</sup> 

<sup>1</sup> Smart Factory Convergence Department, Tech University of Korea, Siheung-si 15073, Republic of Korea; alsrb34840@tukorea.ac.kr

<sup>2</sup> Department of Business Administration, Tech University of Korea, Siheung-si 15073, Republic of Korea

\* Correspondence: madeby2@gmail.com; Tel.: +82-31-8041-0671

**Abstract:** Recently, traffic flow prediction has gained popularity in the implementation of intelligent transportation systems. Most of the existing models for traffic flow prediction focus on increasing the prediction performance and providing fast predictions for real-time applications. In addition, they can reveal the integrity of a prediction when an actual value is provided. However, they cannot explain prediction uncertainty. Uncertainty has recently emerged as an important problem to be solved in deep learning. To address this issue, a Monte Carlo dropout method was proposed. This method estimates the uncertainty of a traffic prediction model. Using 5,729,640 traffic data points from Seoul, the model was designed to predict both the uncertainty and measurements. Notably, it performed better than the LSTM model. Experiments were conducted to show that the values predicted by the model and their uncertainty can be estimated together without significantly decreasing the performance of the model. In addition, a normality test was performed on the traffic flow uncertainty to confirm the normality, through which a benchmark for uncertainty was presented. Following these findings, the inclusion of uncertainty provides additional insights into our model, setting a new benchmark for traffic predictions, and enhancing the capabilities of intelligent transportation systems.

**Keywords:** traffic flow prediction; Monte Carlo dropout; temporal convolutional network; uncertainty



**Citation:** Kim, M.; Lee, D. Why Uncertainty in Deep Learning for Traffic Flow Prediction Is Needed. *Sustainability* **2023**, *15*, 16204. <https://doi.org/10.3390/su152316204>

Academic Editor: Juneyoung Park

Received: 13 September 2023

Revised: 7 November 2023

Accepted: 14 November 2023

Published: 22 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

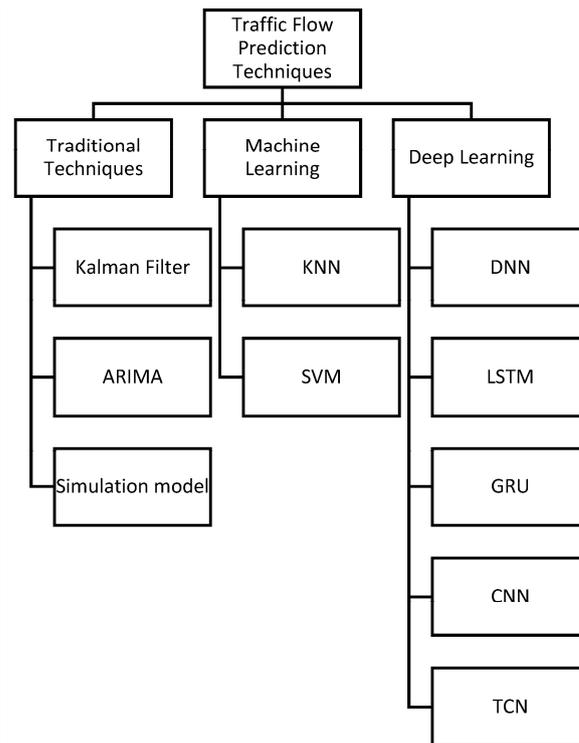
## 1. Introduction

With the development of intelligent traffic systems, traffic flow prediction has become an important and challenging topic of research in both academia and industry. The objective of intelligent traffic systems is to increase the efficiency of traffic operations and alleviate traffic congestion [1]. It is also used as an important reference for vehicle route planning [2] and an important topic of discussion for intelligent transportation systems [3]. Traffic flow prediction serves as a crucial foundation for traffic demand policies in intelligent transportation systems. It enables the prediction of traffic flow in congested areas, facilitating the efficient utilization of traffic resources and contributing to sustainable traffic development at an economic level [4]. Furthermore, the deployment of ITS technology for traffic flow prediction can aid in reducing pollution emissions that result from traffic energy consumption and congestion. This is achieved by promoting the use of public transportation and providing transportation services to those who are transportation disadvantaged.

However, traffic flow prediction is not an easy task. Traffic flow must reflect both temporal and spatial features in the presence of various variables and many uncertain factors [5]. Therefore, it has numerous inflection points. In addition, traffic flow values may differ from the expected values because of special events (e.g., traffic accidents). An inaccurate prediction can lead to the mismanagement of traffic, potentially resulting in safety hazards [6]. Particularly in scenarios such as traffic congestion or special events, a slight misjudgment can escalate into significant safety concerns [7]. As such, extensive research on the prediction of traffic flow using deep learning has been conducted because it is difficult to express the values using specific formulas [8–10]. With rapid urbanization and

an increase in vehicular traffic, ensuring the safety of road users has become paramount [11]. Accurate traffic flow prediction not only aids in efficient traffic management, but also plays a crucial role in enhancing road safety by anticipating and mitigating potential traffic hazards [12].

Recent studies have highlighted approaches that fall into three predominant categories based on the techniques, as depicted in Figure 1: traditional techniques, machine learning, and deep learning. These techniques are representative of the conventional strategies that are deeply anchored in knowledge-driven processes.



**Figure 1.** Categorized overview of traffic flow prediction techniques.

Before the advent of machine and deep learning, traffic flow was primarily predicted using traditional methodologies. These primarily encompass time series models, such as autoregressive integrated moving average (ARIMA) models [13], simulation models [14], and the Kalman filtering technique [15]. Head [14] used a simulation model to predict traffic, highlighting that proactive methods can be used to generate conventional real-time responses [14], and Necula [16] also used a simulation model based on GPS data [16]. These traditional methodologies are the basis for traffic flow prediction; however, they are less accurate than other current methods [17]. Although they can extract temporal features from a time series, they have difficulty expressing spatial relationships in traffic flow prediction tasks [18]. Several researchers have proposed a combination of these methods and artificial intelligence (AI) models to address these problems.

Moreover, with advancements in machine learning and the growth in computing power, researchers have attempted to predict traffic flow using various machine learning-based models [19], including support vector regression [20] and K-nearest neighbor (KNN) [21]. Tempelmeier et al. [21] utilized the KNN algorithm to introduce a map model that captures the spatial and temporal dependencies in traffic flow data, considering the frequency of events [21].

Recently, several models have been introduced to reflect the time series features of traffic, such as recurrent neural networks, long short-term memory (LSTM), and gated recurrent units (GRUs) based on deep learning, including models of graphs and convolutional neural networks (CNNs), such as temporal graph convolutional networks [22] and

diffusion convolutional recurrent neural networks [23], which have been used to reflect spatial information. The emergence of these models has improved the performance of traffic flow prediction [22–25]. In the case of CNNs, traffic flow prediction is achieved by incorporating spatial characteristics via filters. Although they demonstrated effective performance in short-term predictions, they struggled to capture time series characteristics as effectively as they did spatial characteristics, leading to limitations in long-term predictions [26]. To address these limitations, RNN-based models that reflect time series characteristics have been developed. However, traditional RNN models face a vanishing gradient problem. As a solution, LSTM and GRU models have been developed to counteract this issue [27]. Lu et al. [28] identified the shortcomings in prevailing traffic flow prediction models that fail to capture high-level temporal patterns. To address this, they integrated stacked LSTM blocks with convolutional blocks [28]. The introduction of GRUs has further mitigated the vanishing gradient challenge. R. Fu et al. [19] conducted an experiment on short-term traffic flow prediction using LSTM and GRUs, revealing that the deep learning approach, based on current neural networks, such as RNN, surpassed traditional time series models such as ARIMA in capturing temporal nuances [19]. Nonetheless, because traffic flow prediction encompasses both spatial and temporal properties, there is a demand for models that can effectively represent both dimensions. To this end, a temporal convolution network was designed to reflect both time series and spatial features. This model efficiently addresses the issues of computational time and the vanishing gradient problem inherent to traditional RNNs, and its performance can be optimized by adjusting the layer count and tweaking the parameters. It outperformed the traditional LSTM, GRU, and RNN models [29].

Consequently, a plethora of models are continually being developed to enhance prediction accuracy and streamline traffic flow forecasting. However, most studies on traffic flow prediction have focused on prediction accuracy. However, they did not provide information regarding the reliability of the prediction. The performance of the model is evaluated using various evaluation metrics, such as the root mean square error (RMSE) and R-squared; however, they only indicate the performance of the average for past data. They did not provide information on the reliability of the predicted value through the model at the prediction stage through the actual model. This can lead to misjudgment, thus acting as an obstacle to better decision making and deviating from the ultimate goal of deep learning models. In previous studies, this problem was solved using uncertainty [30–32]. Uncertainty is the reliability of the model in regard to the predicted result value. The higher the uncertainty, the lower the reliability of the predicted value, and the higher the uncertainty, the higher the reliability of the predicted value [30]. By addressing the uncertainty in traffic flow predictions, the aim is to provide more accurate predictions and safer road conditions [11]. A reliable model that can provide both accurate and trustworthy predictions is a step forward in ensuring road safety using intelligent traffic systems [7]. Recent advancements in various domains have highlighted the significance of deep learning for uncertainty estimation. For instance, in the realm of MRI imaging, Sherine Brahma et al. [33] emphasized the use of deep learning algorithms to enhance image reconstruction, while stressing the importance of obtaining a metric to identify artifacts [33]. Similarly, Choubineh et al. [34] applied Monte Carlo dropout to assess the reliability of several CNN models in the context of subterranean fluid flow modeling, emphasizing the importance of considering the uncertainty in deep learning models [34]. In the domain of EEG-based predictions, Li et al. [35] introduced a patient-specific seizure prediction framework that considers model uncertainty and proposed a modified Monte Carlo dropout strategy to enhance the reliability of DNN-based models [35]. Murad et al. [36] underscored the importance of quantifying model uncertainty in data-driven air quality forecasts by applying state-of-the-art techniques on uncertainty quantification in real-world settings [36]. These studies provide fresh insights into quantifying the uncertainty of deep learning models and their applicability in real-world scenarios. Transitioning from general deep learning applications to more specific domains, it is evident that the principles of uncertainty are crucial in fields such as traffic flow prediction.

Two recent examples demonstrate the importance of uncertainty in deep learning. Such misclassifications or misjudgments in traffic flow predictions can have dire consequences, leading to unsafe road conditions or accidents [12]. Ensuring the reliability of the predictions is not only about accuracy, but is also about ensuring the safety of all road users [6]. The use of an assisted driving system resulted in the first death in May 2016, which was caused by the system confusing the white side of a trailer with a bright sky. In addition, in a recent image classification, Africans were classified as gorillas, resulting in social issues [31]. The reason for the occurrence of the above phenomenon in deep learning is that the models derive result values through the learning process and blindly trust the corresponding result values. Current general deep learning models cannot indicate whether the value is relatively certain or unreliable based on the reliability of the result value [31], and the models always trust the result value. It is important for the models to convey that they do not know what they do not know, like the human brain.

To solve the above problem, the researchers in this study have attempted to represent uncertainty using Bayesian neural networks, which have been used for the first time in deep learning. Bayesian neural networks can model the weights of the post-probability of the network to obtain information on the standard deviation of the predicted values and estimate uncertainty [32]. However, one of the drawbacks to this network is the high learning cost. Gal [30] proposed the Monte Carlo dropout, demonstrating that sampling alone by applying dropout can approximate the Bayesian inference in the Gaussian process [30]. This allowed us to estimate uncertainty with a lower computing cost and showed better performance than traditional Bayesian neural networks. Various other methods have been proposed to estimate uncertainty [37]. Balaji Lakshminarayan and two others from the DeepMind Laboratory went beyond the existing sampling methods to obtain non-Bayesian uncertainty through deep ensembles for uncertainty estimation. Deep ensembles have been used to address the problem of the slow learning speed of existing Bayesian neural networks and facilitate parallel processing [38]. Extensive research has been conducted in various fields to solve the problems associated with the existing Bayesian neural networks. Gregory Kahn conducted collision prediction based on uncertainty in their study through reinforcement learning. When a collision is predicted, the reward is lowered, and the uncertainty is added to this predictive model [39].

Research on the uncertainty and reliability of this model is ongoing [30,32,37,38]. Even when predicting traffic flow through AI, it is necessary to determine the uncertainty of the predicted value. In addition, there were limited resources for decision making; therefore, it is important to distribute resources wisely. Recently, traffic control systems have changed from real-time to proactive responses. Accurate predictions are required for proactive responses. To distribute resources to events that have not yet occurred, it is necessary to determine the reliability of the predicted values through uncertainty estimation. That is, the accuracy of the predicted values is important; however, if an uncertain value is obtained, the model must be capable of showing the resulting value that can be reflected in the decision [31]. Hence, a benchmark for the size of the uncertainty is essential. Researchers can represent uncertainty through various uncertainty estimation methodologies; however, the benchmark for how large a value can be, used as an indicator that can be reflected in decision making, is still insufficient.

This study reflects the spatiotemporal characteristics of traffic through temporal convolutional networks (TCNs) reflecting the spatiotemporal characteristics, showing better performance than existing time series, deep learning models, LSTM and GRU, and spatial CNN models. In addition, the uncertainty predicted through Monte Carlo dropout can express the reliability of the predicted traffic value, which will enable decision-making support for real-time predictions. Therefore, the Monte Carlo dropout TCN (M-TCN) is proposed, which combines the Monte Carlo dropout (MCDO) and the TCN to express both the predicted values for multiple traffic measurement points in Seoul and the uncertainty of the prediction value. The M-TCN uses the TCN to predict multiple points by considering

complex temporal factors and samples, using the MCDO to extract the uncertainty of each point prediction value and confirm the reliability of the model prediction value.

Second, this study proposes a benchmark that can help in decision making through the correlation between uncertainty and error. Through this, it is possible to grasp the predicted value and the degree of uncertainty for multiple points, and make safer and more reliable decisions by reflecting them in the decision-making process.

To solve the above-mentioned problems, this study proposes a new traffic forecasting model, the M-TCN, to check the uncertainty in traffic forecasting. The contributions from this study are as follows:

- (1) In this study, the MCDO was applied to the TCN models to effectively predict the traffic flow. Through this, it is possible to express the uncertainty of the model that cannot be expressed in the existing traffic model; thus, it is effective not only in predicting traffic, but also in various other places.
- (2) In this study, the effect of the model's uncertainty on the error was confirmed, and a normal distribution was used to present a benchmark for the uncertainty value.
- (3) In this study, the MCDO was applied to various traffic prediction models, and the results were compared to identify differences in the resulting values and to propose an appropriate uncertainty estimation model for traffic flow prediction.

## 2. Methodology

### 2.1. Monte Carlo Dropout

In this study, the MCDO was applied to a traffic prediction model to estimate the traffic prediction uncertainty. The Bayesian model used for the initial uncertainty measurement computes the posterior distribution using a Gaussian distribution and measures the uncertainty. In the existing general deep learning model, the weight is not fixed through learning; however, the probability distribution of the weight is determined. This is equivalent to implementing multiple neural networks because various values may emerge instead of one [40]. However, one of the disadvantages of the Bayesian model is the increased computing costs. To solve this problem, Gal [30] attempted to measure uncertainty with a lower computing cost by applying dropout. Dropout is commonly used in neural networks as a regularization method to prevent model overfitting [41]. This method is generally applied in the training process. However, in the method proposed by Gal [30], dropout was applied before the weight layer. This allows neural networks to learn by randomly omitting some neurons in the input or hidden layer according to the dropout rate and expressing neural networks with different weights. This is the effect of implementing multiple neural networks, and has the same features as a Bayesian neural network. It was shown that deep learning models eventually approximate Bayesian inferences in Gaussian processes and are available in various networks [30].

Moreover,  $\hat{y}$  is defined as the output value of the neural network with  $L$  layers, the weight parameters are set to  $w = \{W_1, \dots, W_L\}$ ,  $x^*$  is defined as the input vector, and  $y^*$  is an observation of  $x^*$ . When a given dataset is defined as  $X = \{x_1, \dots, x_N\}$  and  $Y = \{y_1, \dots, y_N\}$ , the predicted distribution may be expressed as follows:

$$p(y^* | x^*, X, Y) = \int p(y^* | x^*, w) p(w | X, Y) dw, \quad (1)$$

where  $p(y^* | x^*, w)$  is the likelihood and  $p(w | X, Y)$  is the posterior probability. To calculate the posterior probability, the prior probability and likelihood must be defined and multiplied; however, the process for calculating the likelihood through the prior probability is intractable. In this study,  $q(w)$  is arbitrarily defined as having a distribution similar to the posterior probability. The above distribution is called variational distribution, and to approximate it to posterior probability, the posterior probability and Kullback–Leiber divergence (KL-d) of  $q(w)$  are minimized. Minimizing the KL-d minimizes the difference between the two distributions, and if the difference between the two distributions decreases, eventually  $q(w)$  can be used instead because it is close to the posterior probability; that is,

it becomes a problem that minimizes the corresponding KL-d. The predicted distribution is estimated using the variational reference process, as follows:

$$q(y^*|x^*) = \int p(y^*|x^*, w)q(w)dw. \quad (2)$$

Referring to [29], the approximation of the posterior probability  $q(w)$  is expressed as the distribution for a matrix randomly set to 0, according to the Bernoulli distributions in (4) and (5), where  $p_i$  is the dropout rate and  $M_i$  represents the variational parameters.

$$W_i = M_i \cdot \text{diag}\left([z_{i,j}]_{j=1}^{K_i}\right) \quad (3)$$

$$z_{i,j} \sim \text{Bernoulli}(p_i) \text{ for } i = 1, \dots, L, j = 1, \dots, K_{i-1} \quad (4)$$

Therefore, in the Bernoulli distribution,  $T$ , which is the vector set of the sample, is defined as  $\{W_1^t, \dots, W_L^t\}_{t=1}^T$ . Therefore, the predicted mean and uncertainty can be expressed as:

$$E_q(y^*|x^*)(y^*) \approx \frac{1}{T} \sum_{t=1}^T \hat{y}^*(x^*, W_1^t, \dots, W_L^t) = p_{MC}(y^*|x^*), \quad (5)$$

$$\text{uncertainty}(y^*, x, \{W_1, \dots, W_T\}) \approx \sqrt{\frac{1}{T} \sum_{t=1}^T \hat{y}^*(x^*, W_t)^2 - \left(\frac{1}{T} \sum_{t=1}^T \hat{y}^*(x^*, W_t)\right)^2}. \quad (6)$$

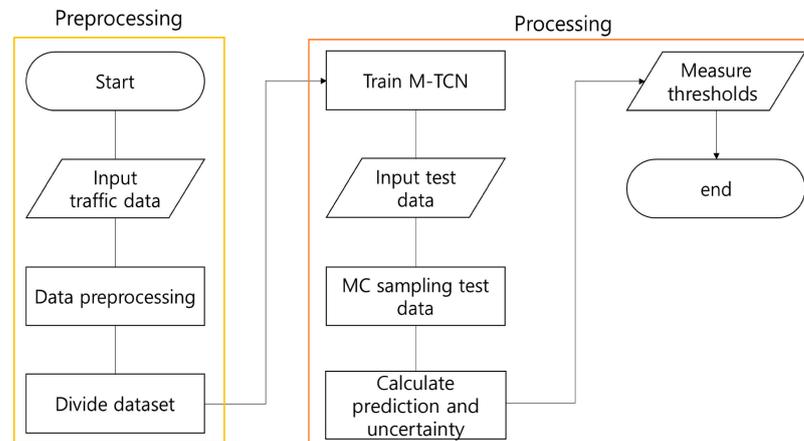
The predicted values and uncertainties shown through the MCDO eventually indicate the average and standard deviation of the resulting values predicted, by providing input into many neural networks created through the dropout. The sum of the predicted values of various neural networks and their average becomes the predicted value of the model to which the MCDO is applied, and the standard deviation of the corresponding values represents the uncertainty. A large uncertainty means that the standard deviation of several neural networks from the dropout is large, and the reliability of the predicted value is small.

## 2.2. Temporal Convolutional Networks

To understand the method implemented in this study, the mechanisms of the TCN used are briefly introduced below. Time series data, from 1 January 2014 to 30 November 2018, were used as the traffic flow data in this study. The time series data consist of input variables for each time step, along with the extracted time series features through RNN-based models specialized in sequential data processing. However, RNN-based models experience a vanishing gradient problem in backpropagation when the distance between the relevant information and the perceptron is significantly reduced in regard to the learning ability. To overcome this problem, a TCN-based model study was conducted [24]. The TCN obtains the output value using a casual convolution filter for the input value, which limits the convolution operation to the region of the time step before the current time point. Unlike conventional CNNs, the TCN receives information from a previous time zone and passes it back to the next time zone. However, computations using casual convolution filters require numerous layers to widen the receptive field. This is problematic for many computations using time series data that require sequential data processing. To compensate for this, Oord et al. [42] proposed a dilated convolutional filter. The filter was adjusted to simplify the calculation and secure the accommodation area covering each input by prioritizing the casual filter as much as the dilation. Thus, it is possible to stack layers effectively to obtain time series features.

### 2.3. Monte Carlo Temporal Convolutional Networks

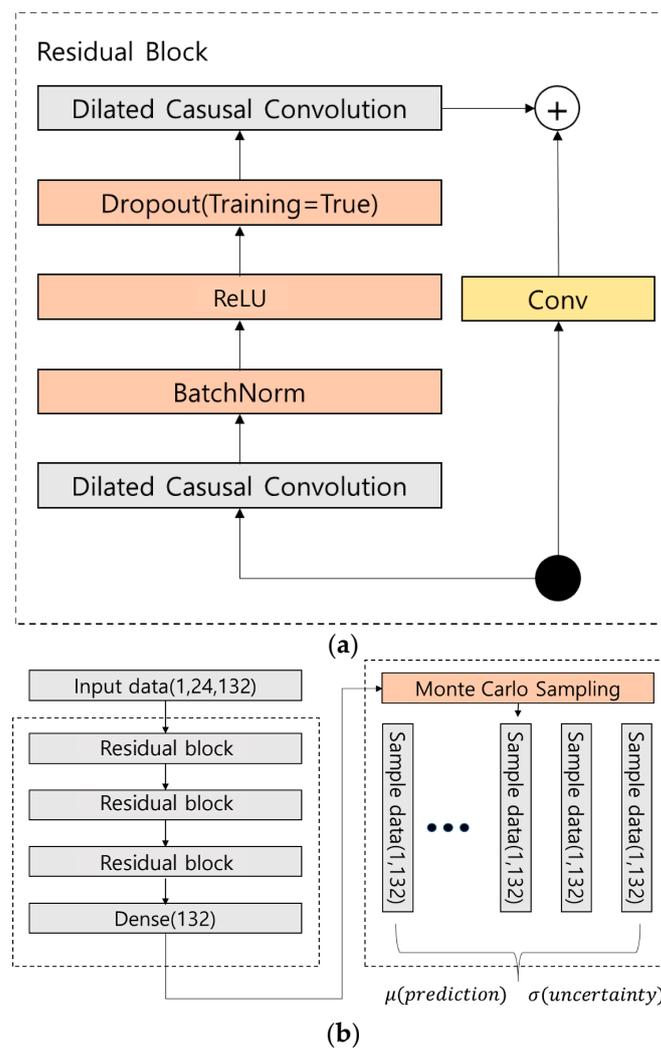
There are two methods for achieving the goal of predicting traffic flow. First, the traffic flow was predicted for various future points based on past traffic flow information. Second, the uncertainty of the model was extracted for the predicted values using Monte Carlo sampling. The overall process is illustrated in Figure 2.



**Figure 2.** Diagram of research flow (the yellow part to the left represents preprocessing and the orange part to the right represents processing).

In this study, the experimental model was divided into preprocessing and processing stages. In the preprocessing stage, the dataset was divided into training, validation, and testing after identifying the missing values in the collected traffic data and performing interpolation and formatting operations. Then, in the processing stage, the model was trained based on the training data, verified using the validation dataset, and tested with the test dataset. The traffic flow was considered as the input to obtain the prediction and uncertainty through the TCN and Monte Carlo sampling. In the test process, the model does not immediately proceed to the prediction stage, unlike the other models, but predicts various values sampled through the Monte Carlo sampling process and calculates the predicted value and uncertainty by calculating the average and standard deviation of the values. Finally, the threshold was measured based on the calculated uncertainty, and all the processes were terminated. The M-TCN model, which is important for executing this process, is shown in Figure 3.

The model consists of three residual blocks, each with a dilation of 1, 2, and 4, to expand the receive field. As shown in Figure 3a, the structure of each residual block consists of dilated casual convolution, batch normalization, and activation rectified linear unit (ReLU) functions, dropout, and dilated casual convolution, which are connected through a skip connection and are finally derived from the dense layer. The result value is a prediction for 132 points for one time point and is in the form (1,132). For the learning model, the distribution of the result value is obtained through 500 Monte Carlo samplings, through which the mean and standard deviation of the sampling values are obtained. Here, the  $\mu(\text{prediction})$  is the mean of sampling values, and the  $\sigma(\text{uncertainty})$  is standard deviation of sampling values. The larger the uncertainty value of the model, the less confident and more unreliable the model is regarding the resulting value. The proposed model can provide an additional predicted value of uncertainty, compared with existing models that represent only the predicted value.



**Figure 3.** (a) Residual block of M-TCN and (b) overall process for M-TCN prediction.

#### 2.4. Models for Comparison

The LSTM is commonly used for sequential data processing, such as speech or text. The LSTM model was used for predictions based on recent time series data. In this study, the researchers combined several other methods to develop a novel prediction model [43]. The LSTM model was used as a comparative model for the M-TCN prediction model by applying the MCDO. CNNs are hierarchical neural network-based algorithms designed to handle multidimensional array data. When a CNN receives multidimensional array data, a weighted array known as a “convolutional filter” operates on the input array and generates the final output through a nonlinear function [44].

#### 2.5. Hypothesis Test Metrics

Correlation analysis and Kolmogorov–Smirnov tests were conducted to verify the additional hypotheses in the experiment. Correlation analysis is a numerical value that quantifies the linear correlation between two variables, by dividing the covariance between the variables by the product of the standard deviation of each variable. In this study, Pearson correlation analysis was performed to confirm the relationship between the uncertainty and the error for the predicted value of the model, and the Pearson correlation used was  $-1$  to  $+1$  using the Cozy–Schwartz inequality, with  $+1$  for a perfect positive linear relationship,  $-1$  for a perfect negative linear relationship, and  $0$  for no relationship [45].

The Kolmogorov–Smirnov test was conducted to test the normality of whether the probability distribution of the population would follow the normal distribution curve,

using estimated uncertainty as a sample group. It is a representative nonparametric test method for determining the specific distribution that a population follows [46].

### 2.6. Evaluation Metrics

The RMSE is calculated by averaging the sum of the squares of deviations, which is the difference between the predicted and actual values, and then taking the square root.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (7)$$

Here,  $N$  denotes the number of predicted values,  $Y_i$  denotes the actual value, and  $\hat{Y}_i$  denotes the predicted value. That is, the difference between the predicted and actual values was squared and averaged, and the root was applied to the RMSE. The closer the RMSE is to 0, the smaller the magnitude of the squared deviation between the predicted and actual values of the model, and it exhibits better performance. The evaluation index is the most frequently used indicator for regression problems.

## 3. Experiments

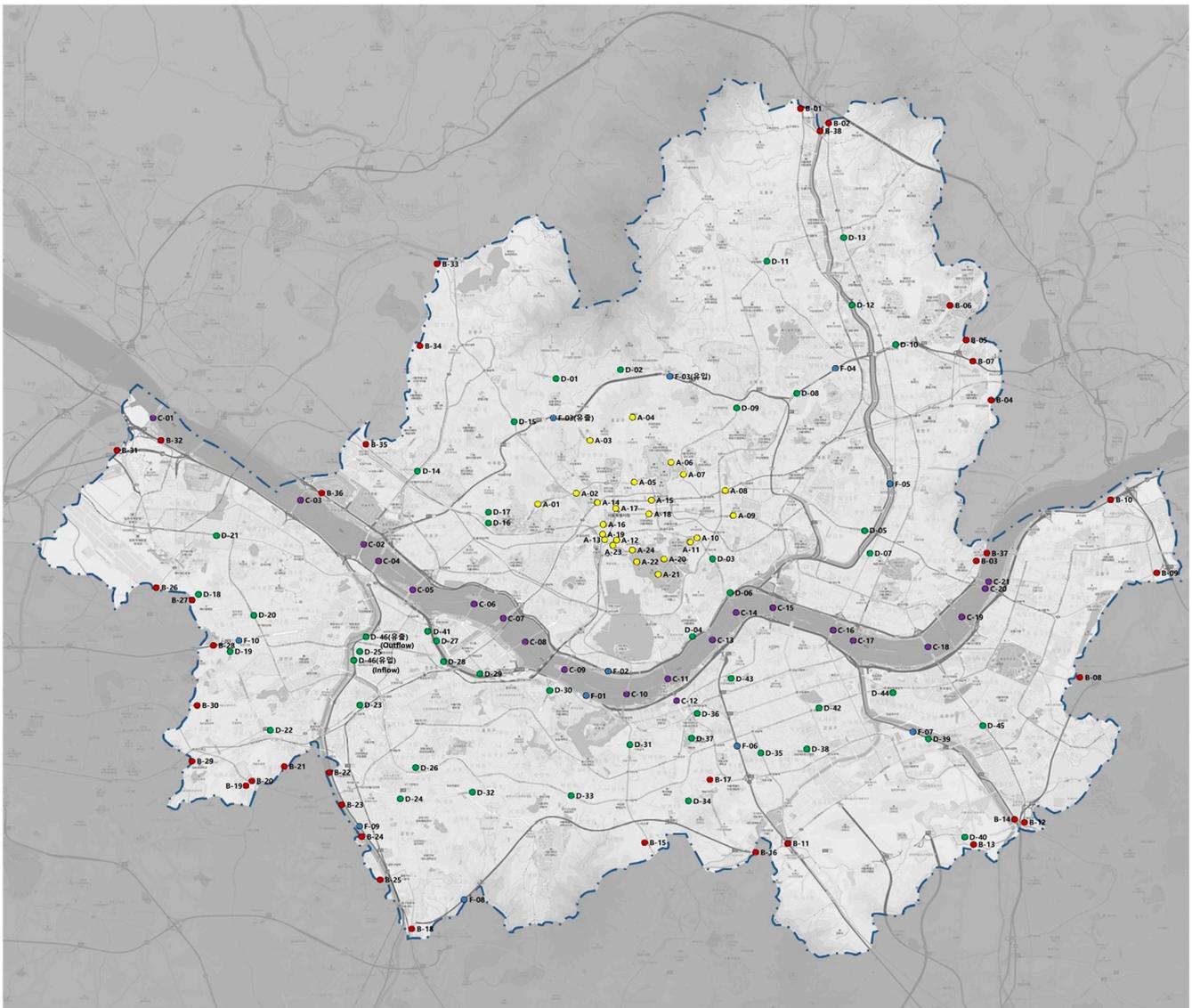
The data used in this study were traffic flow data from Seoul, the capital of Korea, from 1 January 2014 to 30 November 2018, provided by the Seoul traffic information system. The traffic flow dataset for Seoul is described as Table 1 below. The major highways and urban highways were divided into 132 nodes, and the number of vehicles passing through them per hour was measured. Loop, image, and geomagnetic detectors were used for the measurement.

**Table 1.** Traffic flow descriptive statistics.

Observed	Min	Median	Mean	Max	Standard Deviation
5,729,640	1.25	34.62	39.89	229.58	19.58

As shown in Figure 4, the data were collected from 132 traffic measurement nodes in Seoul, which provided only single traffic data on an hourly basis. The data had a total of 5,729,640 observations, with an average of 39.89 and a standard deviation of 19.58. A total of 23,864 values were missing, accounting for approximately 0.5% of the total. For the missing values, the corresponding values were interpolated using a linear interpolation technique. The interpolated data were divided into train, validation, and test data at a ratio of 6:2:2. The training data were used to teach the model, the validation data to verify the model, and the test data to predict the traffic and evaluate its performance.

In this study, a traffic flow prediction model using the MCDO, which reflects the time series features of the traffic and can estimate the uncertainty, was proposed. The experiment was conducted by applying a timestep of lag 1,12,24 and 6,12,24 to check and compare the short- and long-term prediction performance of the models, with and without the MCDO. Based on previous studies, two models, the LSTM and GRU, were used to predict traffic flow and compare the uncertainty patterns exhibited when models other than those with the MCDO were applied. The timestep and lag were calculated in the same environment as in the above experiment. After the learning stage was completed, Monte Carlo sampling was performed 500 times. In this experiment, four Titan RTX graphics processing units (GPUs) were used, and a total of 850 learning epochs were conducted.



**Figure 4.** Traffic nodes in the Seoul dataset (132 traffic measurement nodes).

The MCDO was combined in three models that are frequently used in traffic management. For the 6,12,24 timestep and 1,12,24 lag, the result values are summarized as (a), (b), and (c) in Table 2. (a) The RMSE of the nine result values obtained by applying the MCDO to the TCN was at least 4.68, or a maximum of 6.51, with an average RMSE value of 5.77, and the uncertainty was at least 0.15, or a maximum of 2.42, with an average of 1.21. (b) When the MCDO was applied to the LSTM, the RMSE was at least 4.76, or a maximum of 8.07, with an average RMSE value of 6.51, and the uncertainty was at least 1.94, or a maximum of 3.35, with an average of 2.77. (c) In the case of the CNN, when the MCDO was applied, the RMSE was at least 6.37, or a maximum of 9.03, with an average value of 7.42, and the uncertainty was at least 6.29, or a maximum of 7.33, with an average of 6.82. Overall, the larger the RMSE of the model, the greater the tendency for uncertainty to appear. To verify this in detail, the relationship between the error for the entire test data, as well as the predicted uncertainty must be understood.

**Table 2.** Comparison of the mean of the uncertainty and RMSE using the Seoul traffic flow dataset for various models. (a) Traffic flow predicted using the M-TCN, (b) Traffic flow predicted using the M-LSTM, (c) Traffic flow predicted using the M-GRU.

(a)			
Timestep	Lag	Model	
		M-TCN	
		Uncertainty	RMSE
6	1	0.37	4.79
	12	1.29	6.51
	24	1.25	6.22
12	1	0.15	4.75
	12	1.33	6.01
	24	1.10	6.45
24	1	1.11	4.68
	12	2.42	6.09
	24	1.83	6.41
(b)			
Timestep	Lag	Model	
		M-LSTM	
		Uncertainty	RMSE
6	1	3.35	4.76
	12	2.75	6.39
	24	3.14	6.28
12	1	3.08	6.40
	12	2.69	6.65
	24	2.66	8.07
24	1	2.82	6.00
	12	1.94	6.22
	24	2.53	6.83
(c)			
Timestep	Lag	Model	
		M-GRU	
		Uncertainty	RMSE
6	1	6.98	7.94
	12	6.36	9.03
	24	6.29	8.74
12	1	7.12	6.75
	12	6.52	7.24
	24	6.81	7.00
24	1	6.97	6.37
	12	6.89	7.04
	24	7.33	6.67

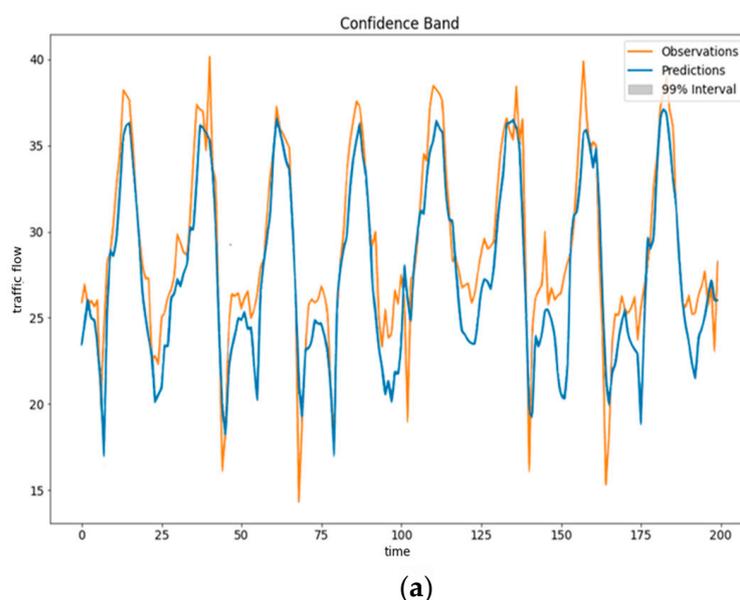
The cases with and without the MCDO have the same neural network structure, and the only difference occurs depending on whether dropout is also applied in the test phase. For the dropout model, 500 samples were sampled, and their average was set as the predicted value. According to Table 3, the short- and long-term prediction performance of the model show a difference of approximately 1% in the RMSE for Lag1, and there is both high and low performance when the MCDO is used. In addition, the difference according to the change in the timestep and lag is approximately 2% and, as in the above case, there is no one-sided performance difference according to the model. It was confirmed that even

if the MCDO is applied in the traffic flow prediction model, uncertainty can be exhibited without significantly impairing the performance.

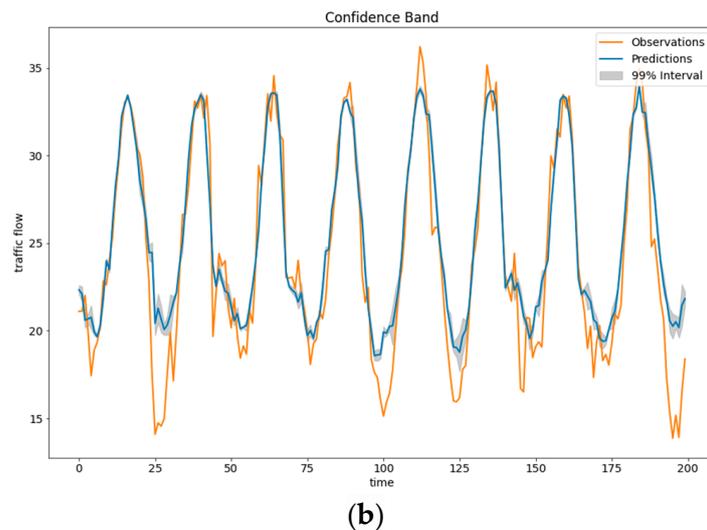
**Table 3.** Comparison of RMSE with and without MCDO using the Seoul traffic flow dataset.

Timestep	Lag	Base Model			
		With MCDO		Without MCDO	
		RMSE	Uncertainty	RMSE	Uncertainty
6	1	4.79	O	4.76	X
	12	6.51	O	6.39	X
	24	6.22	O	6.28	X
12	1	4.75	O	4.73	X
	12	6.01	O	6.37	X
	24	6.45	O	6.13	X
24	1	4.68	O	4.62	X
	12	6.09	O	6.02	X
	24	6.41	O	6.18	X

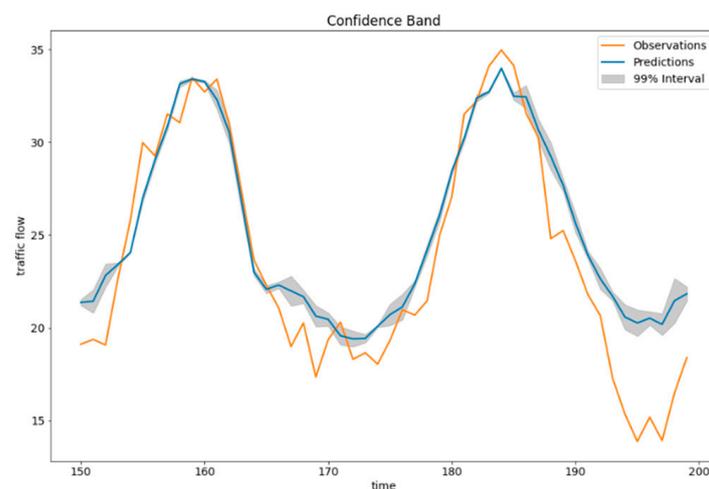
As shown in Table 3, the models without and with the MCDO show a difference in the performance of approximately 2%; however, there is a difference in the amount of information expressed. Figure 5 shows a graph of the 116th point for both models; the RMSE is 2.49 without the MCDO and 2.51 with the MCDO. (a) Without the MCDO, only the information about the predicted and actual values is shown, whereas (b) with the MCDO 99% of the uncertainty is provided using Monte Carlo sampling. The larger the corresponding interval, the more difficult it is for the model to trust the current value. The uncertainty is shown in Figure 6.



**Figure 5.** Cont.



**Figure 5.** Confidence band for one station: (a) without MCDO and (b) with MCDO.



**Figure 6.** Uncertainty and error in the inflection point.

Figure 6 shows an enlarged graph of the 175–200 time interval in Figure 5b, which shows large uncertainties and errors. From 195 to 200 h, the RMSE was 4.27, 71% larger than the RMSE without the MCDO and 70% larger than that with the MCDO. The uncertainty calculated in this section is 2.3 times larger than the average in the other section and contains information indicating that the prediction is less reliable. When using the actual model, if only the performance through the evaluation index for the existing model is considered because the actual value is unknown, the value shows more than 70% variation from the actual value when making decisions.

In addition, Figure 6 shows that the error widened significantly at the inflection point of the box section, and the uncertainty also tended to increase. The uncertainty value in the section showed a difference of 8.8 to 15.7 times, compared to the 30 h time before the section. Based on this, a hypothesis that there may be a correlation between uncertainty and error was established and verified.

To verify the new hypothesis, the mean absolute error (MAE) and mean standard error (MSE) were obtained through prediction and correlation analyses, with uncertainty additionally performed. As a result of the experiment, the MAE had a positive correlation, with an uncertainty of 34.3% and a  $p$ -value of 0.003. The MSE had a positive correlation, with an uncertainty of 31.3% and a  $p$ -value of 0.000, confirming that the MAE and the MSE

had a positive correlation with uncertainty within the set significance level of 0.05. This means that the greater the uncertainty, the higher the probability that the value predicted by the model differs from the actual value.

Based on this experiment, the uncertainty was correlated with the error. In an actual scenario the predicted value can be inferred; however, the actual value is unknown until a corresponding event occurs. Thus, the performance and predicted value of the model must be used in decision making. However, the use of a specific model will allow uncertainty, which involves a positive correlation between the error and difference between the actual and predicted values, to be identified before the actual value, which may help in decision making.

However, to incorporate this into decision making, it is crucial to have a clear understanding of the level of uncertainty. To simplify decision making, it was imperative to ascertain whether the uncertainty approximated a normal distribution. Hence, the Kolmogorov–Smirnov test was performed. The outcome of this test indicated a test statistic of 0.365 and a  $p$ -value of 0.001, confirming that the distribution of uncertainty, with a 99% confidence level, adhered to a normal distribution.

For additional tests, the skewness and kurtosis of uncertainty for a total of 8592 h were analyzed for 132 points. As a result of the analysis, the skewness was 1.53 and the kurtosis was 5.66. Uncertainty follows a normal distribution because the criterion for skewness does not exceed an absolute value of 3, and kurtosis does not exceed an absolute value of 8–10. Based on the test, the reference point according to the magnitude of the uncertainty can be presented through the mean and standard deviation of the uncertainty. Based on the average of the uncertainties, the values of the first standard deviation range were set to be normal uncertainty, low and high for  $\pm 2$  standard deviations, and very low and very high for  $\pm 3$  standard deviations, which were determined as outliers. Therefore, it is possible to determine the reliability of the predicted values and use them as indicators to reflect uncertainty in decision making.

#### 4. Discussion

In this study, uncertainty prediction using the MCDO was employed for traffic prediction. Whereas previous studies mainly aimed to enhance predictive performance, this research introduced the M-TCN model, which infers uncertainty. This approach aligns with the growing prominence of explainable AI and increasing significance of model reliability. The uncertainty values predicted by this model serve as a tool to gauge the trustworthiness of the prediction, and a positive correlation between the uncertainty and error was confirmed through correlation analysis. Furthermore, this research proposes a benchmark that facilitates the application of predicted uncertainty values in real-world decision making, thereby enhancing the efficacy of uncertainty considerations.

In reflecting on the methodologies and results of this research, it is imperative to acknowledge several inherent limitations that might shape the interpretation and application of our findings. First, the analytical framework of our study was anchored predominantly to hourly traffic flow data. This granularity, while providing a broad overview, might not capture the intricacies and fluctuations that are evident in shorter timeframes. Delving into more detailed datasets, specifically those segmented by intervals such as 1 min or 10 min durations, could offer a richer and more nuanced understanding of traffic patterns. Second, our predictive model, in its current iteration, was heavily reliant on a singular variable: traffic flow. While this variable is central to our research focus, the multifaceted nature of traffic dynamics suggests that a broader spectrum of determinants can be considered. Factors such as weather conditions and specific days of the week may further refine our understanding and predictions of traffic flow. Finally, the data sources used in this study were primarily confined to traffic flow metrics.

## 5. Conclusions

In this study, the M-TCN model is proposed to estimate the predicted values of traffic flow measurement and the uncertainty of the predicted values in traffic flow prediction. The M-TCN is a combination of the MCDO and the TCN, which is slightly different from the existing model. The experiment conducted in this study was divided into three stages, and the results were as follows. First, even if the MCDO is applied to a traffic model, it does not necessarily degrade the performance of the existing model. For instance, cases of increased performance have also been recorded. Second, a correlation analysis between the error and uncertainty was conducted based on the results obtained by applying the MCDO to the comparison group model and analyzing the uncertainty and RMSE. Consequently, a positive correlation between error and uncertainty was confirmed. Finally, a normality test for uncertainty was conducted to present a benchmark for uncertainty, so that it can be used in decision making. As a result of the test, it was found that uncertainty followed a normal distribution, and a benchmark was set for each standard deviation from  $\pm 1$  to 3, indicating that uncertainty could be used for decision making. Based on the above results, this study has the following implications.

First, implementing the MCDO does not necessarily degrade the performance of the existing M-TCN model. According to the study results, there was no significant difference in the performance between the models with and without the MCDO and, in some cases, the performance increased. As shown in Table 3, the prediction performance of the model shows a difference of approximately 1% in the RMSE with Lag1, and the performance varies when the MCDO is implemented. This suggests that uncertainty can be inferred by adjusting only part of the existing AI model, without compromising its overall performance. Such uncertainty can be crucial in scenarios where the reliability of real-time predictions is paramount, as it indicates the trustworthiness of the model beyond simply providing an outcome.

Second, numerous studies have emphasized the importance of considering uncertainty during the decision-making process and post-prediction. There is a correlation between the error and the uncertainty value, signifying the difference between the predicted and actual values of the model. It was verified how the uncertainty derived from the M-TCN model correlates with the actual outcome, allowing its incorporation into the decision-making phase. In real-world scenarios, the immediate acquisition of the actual value is not always feasible; hence, the error in the predicted value remains undetermined. However, experiments have confirmed a positive correlation between uncertainty and error. This suggests that the reliability of the predicted value can be ascertained even when the actual data are not yet available. Decisions can be based on predicted values with high certainty. If the predicted reliability is low, decisions should be approached with caution, considering various possible outcomes, without placing undue trust in the predicted value.

Finally, to incorporate traffic flow uncertainty into the decision-making processes, reference points were introduced to determine the acceptable level of uncertainty for decision makers. Although predicting uncertainty is essential for practical applications, establishing standards for the ideal response based on the magnitude of the predicted uncertainty is equally critical. This study verified that uncertainty adheres to a normal distribution, as determined by a normality test. Subsequently, a benchmark based on standard deviation was proposed.

In conclusion, our study delved into traffic flow dynamics using deep learning, focusing on both point and interval predictions. Although the hourly traffic data is primarily used, the results highlight the potential for more detailed insights in future research. This narrow focus, while intentional for the purpose of this research, hints at the potential benefits of a more expansive analytical approach in future research. By integrating additional data sources, such as real-time feedback from traffic signalling systems and analytics from traffic CCTV systems, future research could achieve a more comprehensive understanding of traffic dynamics.

The shift from point to interval predictions in deep learning is evident, and our research provides valuable insights into this evolving area. Furthermore, our findings have implications beyond the technical aspects. They can influence broader areas, such as the social sciences and traffic policy making. The traffic flow prediction model in this study can be used to establish a traffic demand management plan for sustainable transportation through accurate and reliable prediction, and can contribute to reducing traffic congestion. It will also contribute to improving overall social mobility, reducing energy consumption, and reducing air pollution emissions. As demonstrated in this study, the use of AI can offer new ways to shape traffic policies and decisions. This will lay the groundwork for subsequent research, fostering a more profound comprehension of the nexus between AI, traffic trends, and policy formulation.

**Author Contributions:** D.L. initiated the project. M.K. and D.L. designed the experiments. M.K. collected the data. M.K. conducted the data processing and data analysis with the help of D.L. M.K. and D.L. wrote the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Jungseok Logistics Foundation.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **2018**, *318*, 297–305. [[CrossRef](#)]
2. Yuan, J.; Zheng, Y.; Xie, X.; Sun, G. Driving with knowledge from the physical world. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 316–324. [[CrossRef](#)]
3. Zheng, Y.; Capra, L.; Wolfson, O.; Yang, H. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2014**, *5*, 1–55. [[CrossRef](#)]
4. Afrin, T.; Yodo, N. A survey of road traffic congestion measures towards a sustainable and resilient transportation system. *Sustainability* **2020**, *12*, 4660. [[CrossRef](#)]
5. Su, B.; Zheng, W. Traffic Flow Prediction via Spatial Temporal Neural Network “ResLS-C”. In Proceedings of the 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, 5–6 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 119–124. [[CrossRef](#)]
6. Liu, B.; Zhang, T.; Hu, W. Intelligent traffic flow prediction and analysis based on internet of things and big data. *Comput. Intell. Neurosci.* **2022**, *2022*, 6420799. [[CrossRef](#)]
7. Reddy, V.C.S.; Ganji, S.; Nayak, M.M.; Yadav, M.M.; Reddy, G.D. Survey on traffic flow prediction for intelligent transportation system using machine learning. *World J. Adv. Res. Rev.* **2023**, *17*, 460–463. [[CrossRef](#)]
8. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Wang, K.; Wang, J.; Zeng, Z. Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 July 2019; pp. 485–492. [[CrossRef](#)]
9. Altché, F.; de La Fortelle, A. An LSTM network for highway trajectory prediction. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 353–359. [[CrossRef](#)]
10. An, J.; Hu, L.; Hu, M.; Chen, W.; Zhan, J. A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information. *IEEE Access* **2019**, *7*, 20708–20722. [[CrossRef](#)]
11. Momin, K.A.; Barua, S.; Jamil, S.M.; Hamim, O.F. Short duration traffic flow prediction using kalman filtering. In *AIP Conference Proceedings*; AIP Publishing: New York, NY, USA, 2023; Volume 2713. [[CrossRef](#)]
12. Ding, N.; Lu, L.; Jiao, N. Rear-End Crash Risk Analysis considering Drivers’ Visual Perception and Traffic Flow Uncertainty: A Hierarchical Hybrid Bayesian Network Approach. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 7028660. [[CrossRef](#)]
13. Moayedi, H.Z.; Masnadi-Shirazi, M.A. Arima model for network traffic prediction and anomaly detection. In Proceedings of the 2008 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 26–29 August 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–6. [[CrossRef](#)]

14. Head, K.L. Event-based Short-term Traffic Prediction Model. *Transp. Res. Board* **1995**, *1510*, 45–52. Available online: <https://onlinepubs.trb.org/Onlinepubs/trr/1995/1510/1510.pdf#page=51> (accessed on 14 November 2023).
15. Kumar, S.V. Traffic flow prediction using Kalman filtering technique. *Procedia Eng.* **2017**, *187*, 582–587. [[CrossRef](#)]
16. Necula, E. Dynamic traffic flow prediction based on GPS data. In Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 10–12 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 922–929. [[CrossRef](#)]
17. Sun, T.; Yang, C.; Han, K.; Ma, W.; Zhang, F. Bidirectional spatial–temporal network for traffic prediction with multisource data. *Transp. Res. Rec.* **2020**, *2674*, 78–89. [[CrossRef](#)]
18. Xie, P.; Li, T.; Liu, J.; Du, S.; Yang, X.; Zhang, J. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Inf. Fusion* **2020**, *59*, 1–12. [[CrossRef](#)]
19. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 324–328. [[CrossRef](#)]
20. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
21. Tempelmeier, N.; Dietze, S.; Demidova, E. Crosstown traffic-supervised prediction of impact of planned special events on urban traffic. *Geoinformatica* **2020**, *24*, 339–370. [[CrossRef](#)]
22. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [[CrossRef](#)]
23. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926. [[CrossRef](#)]
24. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078. [[CrossRef](#)]
25. Connor, J.T.; Martin, R.D.; Atlas, L.E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [[CrossRef](#)]
26. Yu, D.; Liu, Y.; Yu, X. A data grouping CNN algorithm for short-term traffic flow forecasting. In Proceedings of the Web Technologies and Applications: 18th Asia-Pacific Web Conference, APWeb 2016, Suzhou, China, 23–25 September 2016; Proceedings, Part I. Springer International Publishing: Cham, Switzerland, 2016; pp. 92–103. [[CrossRef](#)]
27. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 153–158. [[CrossRef](#)]
28. Lu, H.; Huang, D.; Song, Y.; Jiang, D.; Zhou, T.; Qin, J. St-trafficnet: A spatial-temporal deep learning network for traffic forecasting. *Electronics* **2020**, *9*, 1474. [[CrossRef](#)]
29. Lv, Z.; Li, J.; Li, H.; Xu, Z.; Wang, Y. Blind travel prediction based on obstacle avoidance in indoor scene. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5536386. [[CrossRef](#)]
30. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Volume 48, pp. 1050–1059.
31. Kendall, A.; Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
32. Gal, Y.; Ghahramani, Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv* **2015**, arXiv:1506.02158. [[CrossRef](#)]
33. Brahma, S.; Kolbitsch, C.; Martin, J.; Schaeffter, T.; Kofler, A. Data-efficient Bayesian learning for radial dynamic MR reconstruction. *Med. Phys.* **2023**, *50*, 6955–6977. [[CrossRef](#)]
34. Choubineh, A.; Chen, J.; Coenen, F.; Ma, F. Applying Monte Carlo Dropout to Quantify the Uncertainty of Skip Connection-Based Convolutional Neural Networks Optimized by Big Data. *Electronics* **2023**, *12*, 1453. [[CrossRef](#)]
35. Li, C.; Deng, Z.; Song, R.; Liu, X.; Qian, R.; Chen, X. EEG-based seizure prediction via model uncertainty learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *31*, 180–191. [[CrossRef](#)]
36. Murad, A.; Kraemer, F.A.; Bach, K.; Taylor, G. Probabilistic deep learning to quantify uncertainty in air quality forecasting. *Sensors* **2021**, *21*, 8009. [[CrossRef](#)]
37. McClure, P.; Kriegeskorte, N. Representing Inferential Uncertainty in Deep Neural Networks through Sampling. OpenReview 2016. Available online: <https://openreview.net/forum?id=HJ1JBj5gl> (accessed on 14 November 2023).
38. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
39. Kahn, G.; Villafior, A.; Pong, V.; Abbeel, P.; Levine, S. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv* **2017**, arXiv:1702.01182. [[CrossRef](#)]
40. Mackay, D.J.C. A practical Bayesian framework for backpropagation networks. *Neural Comput.* **1992**, *4*, 448–472. [[CrossRef](#)]
41. Nitish, S. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
42. van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499. [[CrossRef](#)]

43. Zhao, J.; Deng, F.; Cai, Y.; Chen, J. Long short-term memory-Fully connected (LSTM-FC) neural network for PM<sub>2.5</sub> concentration prediction. *Chemosphere* **2019**, *220*, 486–492. [[CrossRef](#)]
44. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
45. Benesty, J.; Chen, J.; Huang, Y. On the importance of the Pearson correlation coefficient in noise reduction. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 757–765. [[CrossRef](#)]
46. Chakravarti, I.M.; Laha, R.G.; Roy, J. *Handbook of Methods of Applied Statistics*; Wiley: Hoboken, NJ, USA, 1967.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.