information

MDPI

Article

Prediction and Privacy Scheme for Traffic Flow Estimation on the Highway Road Network

Mohammed Akallouch [1,*], Oussama Akallouch [1], Khalid Fardousse [1], Afaf Bouhoute [1] and Ismail Berrada [2]

[1] Faculty of Sciences Dhar El Mahraz, Sidi Mohammed Ben Abdellah University, Fez 30050, Morocco
[2] School of Computer Sciences, Mohammed VI Polytechnic University, Benguerir 43150, Morocco
* Correspondence: mohammed.akallouch@usmba.ac.ma

Abstract: Accurate and timely traffic information is a vital element in intelligent transportation systems and urban management, which is vitally important for road users and government agencies. However, existing traffic prediction approaches are primarily based on standard machine learning which requires sharing direct raw information to the global server for model training. Further, user information may contain sensitive personal information, and sharing of direct raw data may lead to leakage of user private data and risks of exposure. In the face of the above challenges, in this work, we introduce a new hybrid framework that leverages Federated Learning with Local Differential Privacy to share model updates rather than directly sharing raw data among users. Our FL-LDP approach is designed to coordinate users to train the model collaboratively without compromising data privacy. We evaluate our scheme using a real-world public dataset and we implement different deep neural networks. We perform a comprehensive evaluation of our approach with state-of-the-art models. The prediction results of the experiment confirm that the proposed scheme is capable of building performance accurate traffic predictions, improving privacy preservation, and preventing data recovery attacks.

Keywords: traffic flow forecasting; federated learning; privacy-preserving

check for updates

Citation: Akallouch, M.; Akallouch, O.; Fardousse, K.; Bouhoute, A.; Berrada, I. Prediction and Privacy Scheme for Traffic Flow Estimation on the Highway Road Network. *Information* **2022**, *13*, 381. https://doi.org/10.3390/info13080381

Academic Editors: Antonio Comi, Jianbo Li and Junjie Pang

Received: 6 June 2022
Accepted: 27 July 2022
Published: 9 August 2022

1. Introduction

Traffic flow forecasting has long been considered a core and crucial element of Intelligent Transportation Systems. By providing an accurate and timely estimation of the traffic states, traffic predictive systems bring increased flexibility and efficiency to transportation systems. They strongly contribute to improving traffic information [1] and efficiency, leading consequently to smarter mobility. Accurate and timely traffic information is vitally important for both personal travelers and government companies, as it has the potential to provide real-time traffic reports to predict future states, which can help government authorities to control traffic, to make better-informed decisions, and estimate the traffic congestion system by predicting the upcoming traffic flow in a smart city environment [2].

Empowered by technological advances and the wide availability of traffic data, research on Traffic Flow Forecasting has been continually advancing. In fact, TFP has always been a challenging task due to the stochasticity of traffic and dynamic conditions such as weather conditions, calendar (i.e., time of day, day of week), accidents, events, etc. Considerable efforts have been made by researchers from different fields to tackle the TFP problem, trying to provide accurate and more reliable predictions. From mathematical and statistical modeling to more recent data-driven methods, different approaches have been proposed. The data-driven approach formulates TFP as a time series forecasting problem, which aims to predict the traffic states based on historical data, collected by using multiple sensors (e.g., radars, cameras, mobile devices, etc.). More recent TFP approaches rely on deep learning models (e.g., Recurrent Neural Networks, Convolutional Neural Networks) to automatically learn the deep features of traffic data automatically and solve the prediction problem.

Training deep learning approaches for TFP requires a large volume of traffic information that may come from multiple and heterogeneous sources. In the traditional centralized machine learning approach, the traffic data from different sources are uploaded and stored in a central cloud server. The learned model is then distributed to client devices to be applied for prediction. However, centralized learning presents many important problems, especially in terms of data privacy and data protection. In fact, exchanging data with the server raises serious privacy issues. For instance, traffic data collected from users' devices may be sensitive and, if shared on the network, may leak user private information such as captures of people's faces, vehicle license plate information, GPS information, etc.

One recent approach to address the data privacy issue is Federated Learning (FL) [3]. FL is a promising approach enabling collaborative training of machine learning models among different users without sharing private data. When training a model using FL, each user device (participating in the learning) trains the model locally on its data and only shares the learned model parameters (e.g., weights, gradients) with the cloud-server. Once receiving the locally trained parameters, the server performs aggregation and updates the new global model. Though its promising privacy benefits, FL still poses some risks as FL does not guarantee the privacy of all user information. In fact, data recovery from gradient parameters of neural networks has been first addressed in [4,5], which proves the feasibility of the attack from a single neuron or linear layer. Using these weaknesses, attackers can infer original user data, by reversing the shared gradients.

To address these concerns, this paper proposes `FL-LDP`, a hybrid approach including FL and Local Differential Privacy (LDP) [6]. The main idea is to add LDP noises locally to gradients before sending them to the server for aggregation. By adding LDP noises, `FL-LDP` helps prevent inference attacks even when gradients information is publicly available. Thus, `FL-LDP` allows training a prediction model, across multiple devices, while protecting their privacy – formally according to their locally-defined privacy settings. The proposed approach is applied for traffic flow prediction (TFP). Since we are dealing with a time series problem, we implemented `FL-LDP` to train an LSTM model, one of the strong models for time series forecasting.

The main contributions of this word are as follows:

- Introduction of the privacy-preserving approach for traffic flow prediction. This method relies on FL to use the LSTM as a prediction model. As a consequence, the proposed method presents all the advantages offered by FL (cost-saving, privacy benefits, etc.)
- Proposition of the Local Differential Privacy mechanism to strengthen protections in the proposed method, by perturbing the shared model gradients to avoid the privacy threats during the communication phase.
- Evaluation of the proposed framework on a public traffic dataset, and comparison of the results with other centralized machine learning methods. Our proposed mechanism achieves good performance compared to other approaches.

The rest of this paper is organized as follows. Section 2 presents the related literature on TFP and privacy research for Intelligent Transportation Systems. Section 3 is a preliminary section, in which the terminology and basic definitions of FL and LDP are introduced. Section 4 presents the proposed system. Section 5 details the empirical studies and discusses the results. Finally, some future works and conclusions are given in Section 6.

2. Related Work

In this section, we compared recent relevant works on traffic flow prediction and the privacy-preserving for ITS.

2.1. Traffic Flow Forecasting

Traffic flow forecasting approaches are broadly divided in two types: *parametric* and *non-parametric* approaches. The parametric methods usually apply time-series approaches to solve traffic flow prediction problems. One of the first proposed parametric methods is Auto-Regressive Integrated Moving Average (ARIMA) [6], which has been applied to predict the short-term freeway flow. With the accumulation of data, many other approaches start improving the accuracy of TFP using Kalman filtering methods [7,8]. Other parametric approaches for time-series analysis are also used for traffic prediction [9,10]. On the other hand, non-parametric approaches and non-parametric models, have achieved good performance in traffic flow prediction. In [11], the authors proposed a KNN model for short-term traffic flow prediction.

Artificial Neural Networks (ANNs) are a type of non-parametric method that can automatically extract the temporal features from raw data without the data pre-processing phase. Thanks to their numerous advantages (such as the good forecasting performance, capability to work with multi-dimensional data, and high flexibility), ANN has shown great success for traffic prediction problems.

Kumar et al. [12] apply ANN to estimate traffic based on time information and past traffic data such as volume, speed, and density. With the progress and the development of artificial intelligence, many deep learning based approaches have been proposed to improve TFP [1,10,13–16]. Yang et al. [13] introduces an optimized structure of TFP, which is based on a deep learning approach known as stacked auto-encoder. In [17], the authors propose a deep Convolutional Neural Network (CNN) approach to select spatial-temporal traffic features based on speed images in a large-scale transportation network. In addition, a deep 3D convolutional neural network was proposed by [18] in traffic flow prediction by the Spatio-temporal correlation of each segment in the subnetwork. In their paper, the author combines historical data's spatial-temporal properties with three-time intervals: closeness, daily, and weekly. Then they use the attention-based LSTM to embed features.

Deep Recurrent Neural Networks (RNNs) have received increasing interest in time series forecasting. Many works introduce RNN approaches, e.g., LSTM and GRU for traffic flow prediction [19–23]. Finally, many researchers have discussed multimodal deep learning, which combines various deep learning models (CNN and RNN), to improve the prediction performance [24,25].

2.2. Privacy and Security for Intelligent Transportation Systems

In ITS, generally, methods and models are trained directly based on users' data which is stored in a central server. With increasing attention to privacy concerns, one can notice that direct data exchanges can disclose private user information.

Chen et al. [26] introduces a differential privacy approach for trajectory data. The authors propose a data-dependent sanitization algorithm by applying a noisy prefix tree using a Laplace mechanism. In order to secure the location of trajectory data, reference [27] describe a technique to confuse the attacker by adding a path confusion in a centralized-trusted server. Rass et al. [28] introduces an anonymization technique by deriving pseudonyms to provide privacy in trips and samples. To avoid leaking vehicle location data, Hoh et al. [29] introduces a privacy-preserving and distributed traffic monitoring that uses virtual trip lines.

In recent years, many approaches used FL or collaborative learning to analyze private data. Lu et al. [30] introduce a collaborative learning system on the edges for connected vehicles. Moreover, their framework decreases the training time while guaranteeing prediction precision. Besides, Fantacci et al. [31] introduced a federated learning system to protect privacy in many wireless networks. In addition, Saputra et al. [32] uses FL to predict to manage energy resources for electric vehicle networks. The collaborative learning approach was also introduced to provide robust privacy-preserving traffic speed forecasting and protection of topological information [33]. The author proposed an FL framework named FASTGNN that combines a GNN-based predictor utilizing advanced spatial-temporal methods.

Furthermore, using FL and differential privacy have been introduced in many papers such as [34,35]. Truex et al. [36] propose a secure noise-reduction centralized differential privacy to reduce the noise required at each client during the training phase. The authors of [37] provide a privacy-preserving FL approach for learning effective personalized models.

The authors of [37] provide a differential privacy-based personalized FL. They apply the Gaussian mechanism noise for the protection of the model. To protect privacy in smart cities and ITS, recently, many efforts have been proposed with various studies. Triastcyn et al. [38] propose Bayesian differential privacy with FL. In [39], the authors design an FL system, for many applications in smart cities such as energy demand prediction. Y. Liu et al. [40] proposes a small-scale FL for traffic flow forecasting using an aggregation mechanism rather than directly sharing data among clients. In addition, the author in [41] proposed a multi-task FL framework to predict traffic flow. In their work, they use the spatial-temporal dependence of traffic data and design a divisive hierarchical clustering to divide the data at each station into groups. Then the FL approach is collaboratively trained among all stations without direct data sharing.

Although several research works have studied some privacy-preserving approaches for ITS application, there are a few methods that focus on traffic flow prediction. In this work, we introduce a new framework for privacy-preserving traffic prediction, by combining FL and Local Differential Privacy mechanism.

3. Preliminaries

This section gives an overview of Federated Learning and Local Differential Privacy mechanism for traffic flow prediction. The notations of frequently used symbols are explained in Table 1.

Table 1. Summary of symbol notations.

Symbol	Description
$\mathcal{L}$	a specific loss function
w	the weights of a deep neural network
θ	the parameters of a deep neural network
$f(\theta)$	a deep neural network parameterized b θ
$f^*(\theta)$	perturbed f
$\mathcal{D}$	a dataset
∇	Gradient optimization
$\mathcal{M}$	randomized algorithm
$Pr[.\|.]$	conditional probability distribution
ϵ	privacy budget
S	station
T	training round

3.1. Federated Learning

Federated Learning [3] is a decentralized training strategy that enables machine learning model (e.g., RNN) training on data distributed across multiple participating devices. In FL settings, instead of uploading data to servers for centralized training, different clients (e.g., participating stations, agencies, organizations) collaboratively train a global model using their local unshared dataset. As illustrated in Figure 1, a typical FL training round involves (i) local model training on clients to generate a locally-updated model (e.g., gradients, weights), and (ii) global aggregation of these local updates by the server to create an improved global model.

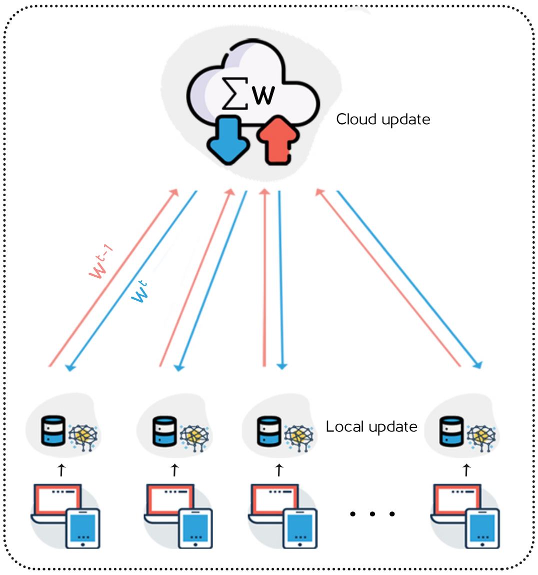


Figure 1. An overview of the federated learning system.

Formally, given a set of K participating clients, where each client k holds a local dataset $\mathcal{D}_k$ with $n_k = |\mathcal{D}_k|$, local client models are trained to minimize a local cost function $F_k(w) = \sum_{i \in \mathcal{D}_k} \frac{1}{n_k} f_i(w)$, where w is the model parameter vector and $f_i(w) = \mathcal{L}(x_i, y_i, w)$ is the loss of the i^{th} training sample. The global model training can be then formulated as the problem of minimizing the following cost function:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{1}$$

where $n = \sum n_k$.

As for most deep learning problems, FL optimization relies on Stochastic Gradient Descent (SGD). The global model can be updated in two different ways: (1) using FedAvg, in which clients perform multiple SGD iterations before sending the updated weights to the server, or (2) using FedSGD [42] which is a technique inspired by the well-known statistical optimization method of Stochastic Gradient Descent (SGD). In FedSGD, every local step is a full deterministic gradient, in which the local client transfers model gradients to a global server, which generates updated parameters from the aggregated gradient.

In this paper, we consider the FedSGD algorithm for FL updates. The training process starts with a randomly initialized (or pretrained) model, at the server level. Then, at each round $t \in \{1, 2, 3, \ldots, T\}$, the server selects clients, according to some defined criteria (e.g., strong communication, computing power) participating in the training process and broadcasts the model parameters. Upon receiving the current model parameters w_t, each client k computes the gradient on its local data $g_k = \nabla F_k(w_t)$, and uploads it to the server. The server aggregates the gradients and updates the global model as indicated by Equation (2). The aggregated global model updates w_{t+1} are sent back to the next round participants.

$$w_{t+1} \leftarrow w_t - \eta \nabla F(w_t) \tag{2}$$

where η is a fixed learning rate during training.

3.2. Local Differential Privacy (LDP)

Local Differential Privacy [43] is the state-of-the-art Differential Privacy (DP) model in distributed settings. To guarantee data privacy, LDP allows users to perturb their data locally before sending them to the cloud. Each user perturbs its data using a random perturbation algorithm $\mathcal{M}$, which transforms the local raw values to values (perturbed) in the same domain, and then sends the results to the server aggregator. In principle, LDP uses the so-called privacy budget (ϵ) to describe the level of protection granted by the LDP mechanism: the smaller ϵ, the weak the privacy. Formally, LDP can be written as follows:

Definition 1 ((ϵ, δ)-LDP [44])**.** *Let $\mathcal{M} : \mathbb{X} \rightarrow \mathbb{Y}$ be a randomized mechanism with a domain $\mathbb{X}$ and a range $\mathbb{Y}$. For a non-negative ϵ, $\mathcal{M}$ satisfies (ϵ, δ)-LDP if: $\forall x, x' \in \mathbb{X}$, and $\forall S \subseteq \mathbb{Y}$:*

$$Pr[y \in S \,|\, x] \leq e^{\epsilon} \times Pr[y \in S \,|\, x'] \tag{3}$$

where $Pr[.|.]$ denotes the conditional probability distribution depending on $\mathcal{M}$. Therefore, in LDP, a random perturbation is given by clients with a privacy budget ϵ. Instead of sharing the original parameters, the centralized server receives only the perturbed updates, which makes the server incapable recover the original data $x \in \mathbb{X}$. The case where $\delta = 0$ is called pureϵ-LDP.

Generally, (ϵ, δ)-LDP can be achieved by adding Gaussian noise on true values. The local perturbation noise-based Gaussian mechanism is defined as follows:

Definition 2 (Gaussian Mechanism [44])**.** *Assume that a user wants to release a function $f : \mathbb{X} \rightarrow \mathbb{R}$, with an input subject to (ϵ, δ)-LDP. The Gaussian perturbation mechanism is as follows:*

$$\mathcal{M}(\mathbb{X}) \hat{=} f(\mathbb{X}) + \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \tag{4}$$

If the bound of the sensitivity function is Δ_f:

$$||f(x) - f(x')||_2 \leq \Delta_f, \; \forall x, x' \tag{5}$$

then, a $\delta \in (0, 1]$ Gaussian mechanism satisfies (ϵ, δ)-LDP, if :

$$\epsilon = \frac{\Delta_f}{\sigma} \sqrt{2 \times log \frac{1.25}{\delta}} \tag{6}$$

4. System Model and FL-LDP Protocol

4.1. System Model

In this work, we select N stations connected with a centralized server. We aim to estimate the flow of vehicles based on the previous data from many stations. Each participating station has the same structured dataset (historical traffic flow) and collaborates to train an LSTM model. Thus, the historical data is a dataset of the loop vehicle radar stations that are already installed on most freeways. The prediction task is to estimate the flow of traffic on each road segment. In each local training round, the global server computes the average of received updates from stations and updates the model with this aggregation. We suppose that the Federated Learning cloud server is honest-but-curious, so the server may want to learn some extra information from gradient data [36,45]. An illustration of the considered scenario is presented in Figure 2.

4.2. Federated Learning Based LDP: `FL-LDP` *Protocol*

The `FL-LDP` protocol is detailed in Algorithms 1 and 2. `Fl-LDP` protocol consists of five main phases: (1) initialization, (2) training, (3) noise update, (4) aggregation, and (5) distribution. The protocol steps that are performed at client and server sides, for one training round, are presented below:

Server side. The global server first randomly generates an initial model weights θ_{init} then:

1. Selects a subset of stations to participate in the current training round.
2. Distributes the initialized parameters to all participating stations.
3. Waits to receive gradients computed by the participating stations.
4. Aggregates the updated gradients sent by all stations using the aggregation protocol.
5. Shares the new global model to all stations.

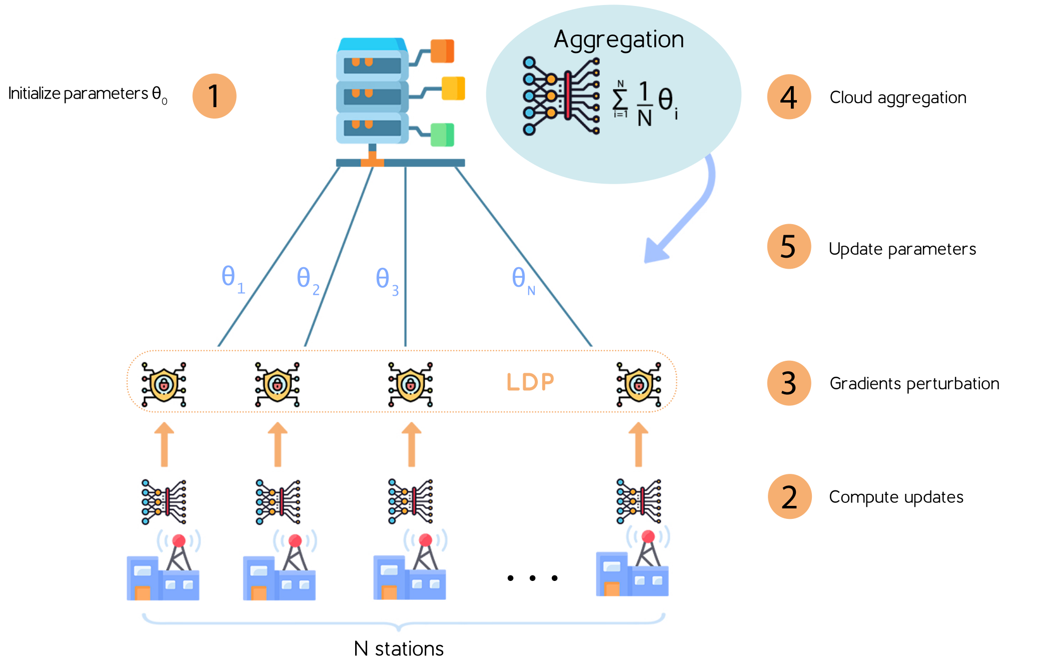


Figure 2. An overview of FL-LDP system: (**1**) Server initializes parameters θ_0, (**2**) Clients compute gradients updates on local stored data, (**3**) Clients perturb the gradient parameters, (**4**) The server computes the client gradients to get the new model, (**5**) The aggregated model is return to clients, and then the start the next iteration.

Algorithm 1: FL-LDP protocol: Server side

```
Input: A set of stations $S = \{s_1, s_2, \ldots, s_N\}$, Gradient optimization $\nabla\mathcal{L}$, Number of rounds $T$
Output: Parameter $\theta$
Initialize the global model parameters $\theta \leftarrow \theta_{init}$
foreach round $t \in \{1, 2, 3, \ldots, T\}$ do
    $\{S_t\} \leftarrow$ choose stations participating in the current round from $S$
    if $S_t$ is empty then
        break
    end
    Send $\theta$ to stations in group $\{S_t\}$
    foreach Station $s \in S_t$ do
        Receive the noisy gradients $\nabla\mathcal{L}(\theta_s, x_s)$
    end
    Aggregate the gradients from the current stations and update the current parameters $\theta$:
        $\theta \leftarrow \theta - \eta \times \frac{1}{|St|} \times \sum_{s \in S_t} \nabla\mathcal{L}(\theta_s, x_s)$
    where $\eta$ denotes the learning rate
    if $\theta$ meets the stopping conditions then
        break
    end
end
```

Algorithm 2: `FL-LDP` protocol: Client side

```
Input: Data $x_s$, Gradient optimization $\nabla\mathcal{L}$, Random perturbation algorithm $\mathcal{M}$,
    Number of epochs $E$
Output: Parameter $\theta_s$
Receive $\theta$ from the server
Initialize the model with $\theta_s \leftarrow \theta$
foreach each local epoch $i \in \{1, 2, 3, \dots, E\}$ do
    Compute the gradient $\nabla\mathcal{L}(\theta_s, x_s)$
end
Apply the LDP mechanism $\mathcal{M}$ to calculate the noisy gradient
    $\theta_s \leftarrow \mathcal{M}(\nabla\mathcal{L}(\theta_s, x_s))$
Send $\theta_s$ to the server
```

Client side. Each participating station initializes (i) the local model with the received parameters θ, and (ii) the local LDP mechanism $\mathcal{M}$ with privacy secure parameters according to the private preferences. Then, in each training round, a station:

1. Initializes the local model.
2. Trains the local model by locally computing the gradients on its private local dataset.
3. Uses the LDP algorithm $\mathcal{M}$ to compute the noisy gradient.
4. Sends the noisy gradient to the cloud server.
5. Waits to receive the aggregated gradient updates from the server.

The training process continues, for both the global server and $N-$stations, until a predetermined condition is reached (the maximum number of iterations, model convergence, or there's no station found to participate in the computation round).

5. Experiments

In this section, we present and discuss the empirical results related to the performance of `FL-LDP`. The proposed protocol has been implemented by using a freeway dataset collected from the Caltrans Performance Measurement System (PeMS) [46]. PeMS traffic flow dataset provides real-time traffic data at five-minute intervals on freeways, collected by multiple sensors across the major areas of all California.

The PeMS data is a set of time series that are presented as intervals of time sequences. Therefore, predicting the traffic data at time t is based on historical traffic flow sequence of m length, i.e., $X = \{x_{t-1}, x_{t-2}, \dots, x_{t-m}\}$ and $Y = \{y_t\}$. In our experiments, to predict the traffic for the next 5 min, we select 60-minutes traffic data, corresponding to a series of 12 sequences of 5 min. The traffic flow data used for the experiments is collected during the first 28 weeks of 2017 from 31 stations. The first five weeks are chosen for the training data and the next week in the testing phase. PeMS data is equally divided and distributed to 31 stations. For FL setting, we set the number of participating stations, in each round, the learning rate at $\eta = 0.001$, $|S_v| = 10$, mini-batch SGD size at $B = 128$, rounds at $T = 500$, and local training epochs at $E = 10$.

As measures of performance, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root-Mean Square Error (RMSE), and Mean Square Error (MSE) are used for the prediction accuracy. These metrics are formulated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_p}| \tag{7}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_p})^2 \tag{8}$$

$$\mathrm{RMSE} = [\frac{1}{n}\sum_{i=1}^{n} (|y_i - \hat{y_p}|)^2]^{\frac{1}{2}} \tag{9}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y_p}}{y_i}| \tag{10}$$

where y_i denotes the real traffic flow, and $\hat{y_p}$ is the predicted or forecasted traffic flow.

For performance measures, we implemented different approaches by using TensorFlow [47] and/or Pytorch [48] frameworks. All pre-trained backbones are publicly available on Github. The different experiments and simulations are trained on one server: Fujitsu Primergy rx2540 M4, with CPU Intel Xeon Gold 6152, 512GB RAM, and Nvidia Tesla P100 16GB GPU.

5.1. System Performance

To evaluate the proposed FL-LDP protocol, we compare its performance with the following centralized baseline models: GRU, SAE, LSTM, and Support Vector Machines (SVM).

The performance is reported on the same PeMS dataset. Table 2 summarizes the scores obtained from the evaluated models. According to these results, the GRU model has a good performance, as shown by its low error scores.

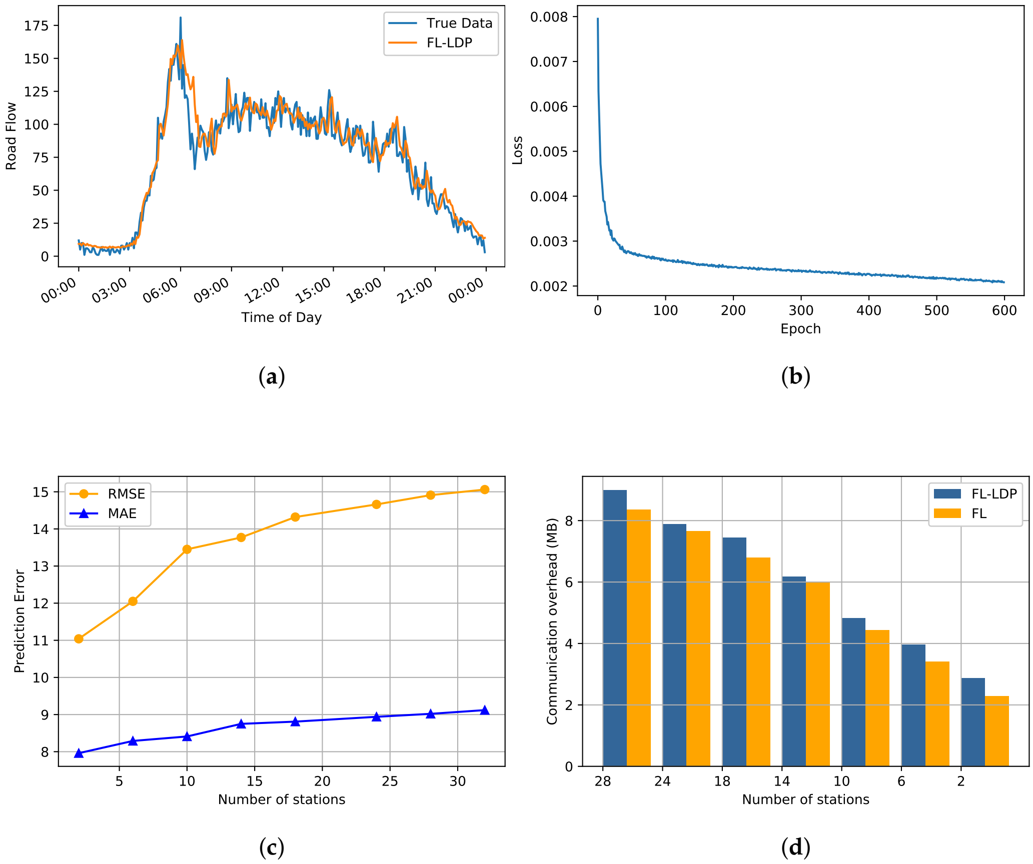
Table 2. Performance comparison of FL-LDP, LSTM, SAE, and SVM models.

Model	MAE	MSE	RMSE	MAPE
FL-LDP	8.03	103.24	11.16	18.76%
GRU Model [21]	7.20	99.32	9.97	17.78%
SAE Model [1]	8.26	99.82	11.60	19.80%
LSTM model [19]	8.28	107.16	11.45	20.32%
SVM model [49]	8.68	115.52	13.24	22.73%

Comparing the different model experiments, we can conclude that our FL-LDP has achieved state-of-the-art results, since its score values remain not very far from the other model scores. Moreover, we can see that the performance of LSTM has been improved by using the advanced FL architecture: MAE is improved from 8.28 to 8.03. These experimental results prove the significant importance of the proposed approach in combining accurate traffic flow prediction and data privacy protection.

To give more insights about FL-LDP prediction, Figure 3a shows an example of a 5-min traffic data forecast task in a single day. We notice that the obtained experiments of FL-LDP system are close to those of the real information. Furthermore, the stability and the good convergence of the model are clearly visible in Figure 3b which illustrates the local model loss of a client station. In Figure 3c, we compare the impact of the number of participating stations (i.e., $S = 2, 6, 10, \ldots$) on the performance of FL-LDP. The number of stations has an unfavorable impact on the model performance i.e., MAE and RMSE vary from (8 to 9) and (11 to 15) respectively. This impact may be due to many reasons, such as connection failures in some stations which prevent them from uploading their gradient information update in a specific round. Furthermore, unconnected stations cannot receive server updates making the local performance different and the cloud aggregation difficult.

An evaluation of the communication overhead in the case of FL and FL-LDP is performed. Figure 3d shows the communication overhead of the two algorithms with different participation ratios. The plotted results show that the perturbation mechanism has small communication costs and is unaffected by the number of participants.



(**a**) (**b**)

(**c**) (**d**)

Figure 3. System performance.(**a**) Traffic flow prediction. (**b**) Loss of local model. (**c**) Prediction error with different client stations. (**d**) Communication overhead.

5.2. Privacy Budget ϵ Impact

In order to study the impact of the privacy-budget parameter ϵ on the stability of the proposed approach, we evaluate FL-LDP with three values of ϵ ($\epsilon \in \{1,3,5\}$). The obtained results are reported in Figure 4a. By comparing the loss for the different privacy parameters, we can see a similar loss for the two higher privacy $\epsilon = (1,3)$ with a noticeable difference for the lowest privacy ($\epsilon = 5$). We remind that setting a smaller value of the privacy budget ϵ means adding more noise which implies stronger privacy protection. The curve in Figure 4a shows that our approach maintains better resilience even with a high level of privacy ($\epsilon = 1$).

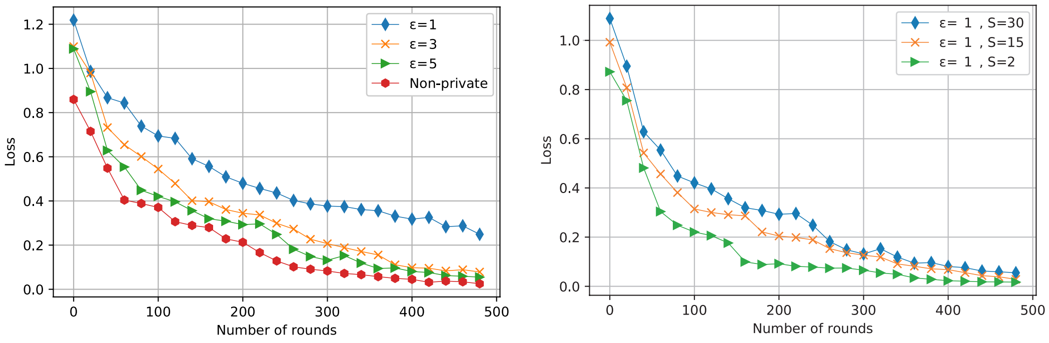
Finally, we display in Figure 4b the FL-LDP performance for three different numbers of participants ($S \in \{2,15,30\}$), with a fixed privacy budget ($\epsilon = 1$). Although the increase in the number of participating stations has previously been reported to have an unfavorable impact on performance, the curves plotted in Figure 4b show that FL-LDP has similar convergence rates toward the last training round, even when the number of participants increases.

5.3. Performance of Secure FL-LDP Model Training

This subsection studies the FL-LDP resilience against poisoning attacks. This is assessed by measuring the Attack Success Rate (*ASR*) with/out the application of local differential privacy. The *ASR* is calculated as:

$$ASR(\mathrm{X}) = \frac{1}{n}\sum_{i=0}^{n} F(x_i) \neq F^*(x_i) \tag{11}$$

where F is the trained model of neural network (e.g., LSTM), and F^* is the perturbed F.



(**a**) (**b**)

Figure 4. (**a**) The impact of the privacy-budget parameter on the model. (**b**) The impact of the number of participants on the model.

The results obtained using different privacy budgets ($\epsilon \in \{1, 3, 5\}$) are displayed in Figure 5. According to these results, we can conclude that FL-LDP can provide an effective defense against attacks ($ASR = 2\%$). Compared to the implementation without the LDP scheme, which has a higher possibility of privacy leakage ($ASR > 50\%$), FL-LDP achieves very much lower ASRs (between 2% for $\epsilon = \{1\}$ and 11% for $\epsilon = \{5\}$). These results argue that our hybrid approach can protect user information under various privacy conditions. It can completely improve the performance even when the privacy budget is high.

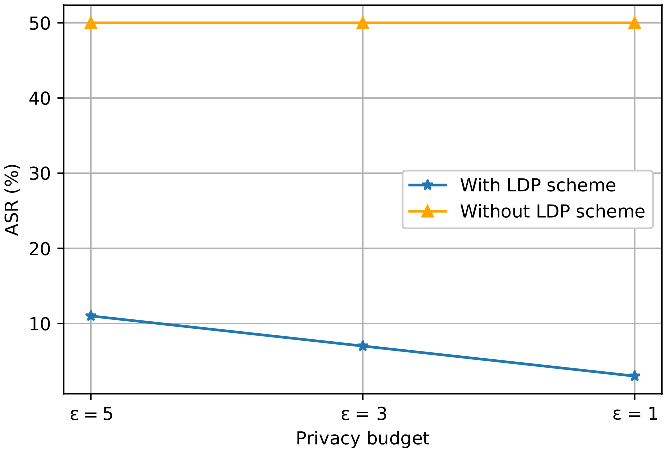


Figure 5. The impact of privacy budgets on the attack-success-rate (ASR).

5.4. Discussion

The performed experimental studies aim to provide insights on some important design intuitions of the FL-LDP system. This latter is applied for predicting traffic flow and its advantages and limitations in different scenarios are discussed. The obtained results confirm the robustness and the effectiveness of our mechanism. So, the findings of these results are summarized below:

- FL-LDP provides strong data protection and accurate traffic prediction, by combining local differential privacy (Gaussian noise) and federated learning (FedSGD). Specifically, the model achieves good performance by aggregating perturbed gradients, instead of the true gradient values, which guarantees user privacy protection.
- Increasing the number of participants in each communication round may lead to some issues such as the communication overhead and model convergence due to the failure of synchronizing some stations.

- The privacy budget ϵ affects the convergence of the model in some cases: decreasing the privacy budget can delay the convergence.

6. Conclusions

In this work, we introduce `FL-LDP`, a traffic flow forecasting system that combines federated learning with local differential privacy. This enables prediction models to be collaboratively trained on multiple stations without compromising user data privacy. More specifically, by applying the LDP mechanism to perturb the computed gradients, attackers cannot recover users' sensitive data (from their shared gradients). The performance of the proposed approach is evaluated on the PeMS traffic flow data. We perform a comprehensive and empirical study of the LSTM trained using `FL-LDP` with different centralized ML models (GRU, LSTM, SAE, and SVM). The results showed that our approach achieved a good prediction performance in terms of MAE, MSE, RMSE, and MAPE scores by 8.03, 103.24, 11.16, and 18.76%, respectively. On the other hand, extensive analyses prove that by adding the LDP mechanism, the framework achieves a good performance in terms of ensuring strong user privacy protection (the ASR score decreases from 50% to 2%). In future work, we intend to train more ML models with the proposed federated learning mechanism, focusing more on communication costs.

Author Contributions: M.A., O.A. designed the model framework and performed the experiments. M.A. and A.B. wrote the manuscript with input from all authors. Both I.B. and K.F. authors supervised the project. All authors read and approved the final manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 865–873. [CrossRef]
2. Aung, N.; Zhang, W.; Dhelim, S.; Ai, Y. T-Coin: Dynamic traffic congestion pricing system for the Internet of Vehicles in smart cities. *Information* **2020**, *11*, 149. [CrossRef]
3. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
4. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning: Revisited and enhanced. In Proceedings of the International Conference on Applications and Techniques in Information Security, Auckland, New Zealand, 6–7 July 2017; Springer: New York, NY, USA, 2017; pp. 100–110.
5. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
6. Bindschaedler, V.; Shokri, R.; Gunter, C.A. Plausible deniability for privacy-preserving data synthesis. *arXiv* **2017**, arXiv:1708.07975.
7. Ojeda, L.L.; Kibangou, A.Y.; De Wit, C.C. Adaptive Kalman filtering for multi-step ahead traffic flow prediction. In Proceedings of the 2013 IEEE American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 4724–4729.
8. Kumar, S.V. Traffic flow prediction using Kalman filtering technique. *Procedia Eng.* **2017**, *187*, 582–587. [CrossRef]
9. Ghosh, B.; Basu, B.; O'Mahony, M. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 246–254. [CrossRef]
10. Olayode, I.O.; Tartibu, L.K.; Okwu, M.O.; Ukaegbu, U.F. Development of a hybrid artificial neural network-particle swarm optimization model for the modelling of traffic flow of vehicles at signalized road intersections. *Appl. Sci.* **2021**, *11*, 8387. [CrossRef]
11. Wu, Y.; Tan, H.; Peter, J.; Shen, B.; Ran, B. Short-term traffic flow prediction based on multilinear analysis and k-nearest neighbor regression. In Proceedings of the COTA International Conference of Transportation Professionals (CICTP), Beijing, China, 24–27 July 2015; pp. 556–569.
12. Kumar, K.; Parida, M.; Katiyar, V.K. Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport* **2015**, *30*, 397–405. [CrossRef]

13. Yang, H.F.; Dillon, T.S.; Chen, Y.P.P. Optimized structure of the traffic flow forecasting model with a deep learning approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2371–2381. [CrossRef]
14. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [CrossRef]
15. Wu, Y.; Tan, H.; Qin, L.; Ran, B.; Jiang, Z. A hybrid deep learning based traffic flow prediction method and its understanding. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 166–180. [CrossRef]
16. Fredianelli, L.; Carpita, S.; Bernardini, M.; Del Pizzo, L.G.; Brocchi, F.; Bianco, F.; Licitra, G. Traffic flow detection using camera images and machine learning methods in ITS for noise map and action plan optimization. *Sensors* **2022**, *22*, 1929. [CrossRef]
17. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [CrossRef]
18. Ul Abideen, Z.; Sun, H.; Yang, Z.; Ali, A. The Deep 3D Convolutional Multi-Branching Spatial-Temporal-Based Unit Predicting Citywide Traffic Flow. *Appl. Sci.* **2020**, *10*, 7778. [CrossRef]
19. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
20. 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; pp. 153–158.
21. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 IEEE 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 324–328.
22. Xiao, Y.; Yin, Y. Hybrid LSTM neural network for short-term traffic flow prediction. *Information* **2019**, *10*, 105. [CrossRef]
23. Karimzadeh, M.; Schwegler, S.M.; Zhao, Z.; Braun, T.; Sargento, S. MTL-LSTM: Multi-Task Learning-based LSTM for Urban Traffic Flow Forecasting. In Proceedings of the 2021 IEEE International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2021; pp. 564–569.
24. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
25. Ren, J.; Hu, Y.; Tai, Y.W.; Wang, C.; Xu, L.; Sun, W.; Yan, Q. Look, listen and learn—A multimodal LSTM for speaker identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
26. Chen, R.; Fung, B.; Desai, B.C. Differentially private trajectory data publication. *arXiv* **2011**, arXiv:1112.2020.
27. Hoh, B.; Gruteser, M. Protecting location privacy through path confusion. In Proceedings of the IEEE First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), Washington, DC, USA, 5–9 September 2005; pp. 194–205.
28. Rass, S.; Fuchs, S.; Schaffer, M.; Kyamakya, K. How to protect privacy in floating car data systems. In Proceedings of the Fifth ACM International Workshop on VehiculAr Inter-NETworking, San Francisco, CA, USA, 15 September 2008; pp. 17–22.
29. Hoh, B.; Gruteser, M.; Herring, R.; Ban, J.; Work, D.; Herrera, J.C.; Bayen, A.M.; Annavaram, M.; Jacobson, Q. Virtual trip lines for distributed privacy-preserving traffic monitoring. In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, Breckenridge, CO, USA, 17–20 June 2008; pp. 15–28.
30. Lu, S.; Yao, Y.; Shi, W. Collaborative learning on the edges: A case study on connected vehicles. In Proceedings of the 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19), Renton, WA, USA, 9 July 2019.
31. Fantacci, R.; Picano, B. Federated learning framework for mobile edge computing networks. *CAAI Trans. Intell. Technol.* **2020**, *5*, 15–21. [CrossRef]
32. Saputra, Y.M.; Hoang, D.T.; Nguyen, D.N.; Dutkiewicz, E.; Mueck, M.D.; Srikanteswara, S. Energy demand prediction with federated learning for electric vehicle networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019; pp. 1–6.
33. Zhang, C.; Zhang, S.; James, J.; Yu, S. FASTGNN: A topological information protected federated learning approach for traffic speed forecasting. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8464–8474. [CrossRef]
34. Hao, M.; Li, H.; Luo, X.; Xu, G.; Yang, H.; Liu, S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans. Ind. Inform.* **2019**, *16*, 6532–6542. [CrossRef]
35. McMahan, H.B.; Ramage, D.; Talwar, K.; Zhang, L. Learning differentially private recurrent language models. *arXiv* **2017**, arXiv:1710.06963.
36. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; pp. 1–11.
37. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized federated learning with differential privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [CrossRef]
38. Triastcyn, A.; Faltings, B. Federated learning with bayesian differential privacy. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2587–2596.
39. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]

40. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [CrossRef]
41. Zeng, T.; Guo, J.; Kim, K.J.; Parsons, K.; Orlik, P.; Di Cairano, S.; Saad, W. Multi-task federated learning for traffic prediction and its application to route planning. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–15 July 2021; pp. 451–457.
42. Chen, J.; Pan, X.; Monga, R.; Bengio, S.; Jozefowicz, R. Revisiting distributed synchronous SGD. *arXiv* **2016**, arXiv:1604.00981.
43. Dewri, R. Local differential perturbations: Location privacy under approximate knowledge attackers. *IEEE Trans. Mob. Comput.* **2012**, *12*, 2360–2372. [CrossRef]
44. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
45. Lyu, L.; Yu, H.; Yang, Q. Threats to federated learning: A survey. *arXiv* **2020**, arXiv:2003.02133.
46. Chen, C. *Freeway Performance Measurement System (PeMS)*; University of California: Berkeley, CA, USA, 2003.
47. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
48. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv* **2019**, arXiv:1912.01703.
49. Mohandes, M.A.; Halawani, T.O.; Rehman, S.; Hussain, A.A. Support vector machines for wind speed prediction. *Renew. Energy* **2004**, *29*, 939–947. [CrossRef]