Vehicle Location Algorithm Based on Federated Learning and Smart Phone in GNSS Low Sampling Rate Scene

Wenbiao Guo1,2,3, Hang Li3, Feng Yin3,4 and Bo Ai1,2,*

1School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China
2State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China
3Shenzhen Research Institute of Big Data, Shenzhen, China
4School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, China

*Corresponding author e-mail: boai@bjtu.edu.cn

Abstract. In the existing vehicle positioning system based on Global Navigation Satellite System/Inertial Navigation System (GNSS/INS), when the GNSS signal is lost, the error accumulated by using only the INS will damage the positioning accuracy. In order to improve the accuracy, this paper proposed a positioning method based on data-driven and learning models, which utilized distributed data sets to collaboratively construct accurate positioning models through federated fusion algorithms without sacrificing user privacy. In the field scenarios of IID and Non-IID types, this paper compared the performance of INS and the existing two typical methods, DeepSense and PVAUA, it is verified that the two federated learning algorithm models constructed had higher positioning accuracy in different scenarios and different GNSS signal loss durations. The results were analyzed.

Keywords: Vehicle Localization, Data-driven, Federated Learning, Privacy Protection.

1. Introduction

Currently, the most widely used vehicle positioning system is the GNSS/INS combined system. In open areas, vehicles can obtain stable and high-accuracy positioning services through GNSS (such as GPS in the United States, GLONASS in Russia, GALILEO in the European Union and BeiDou in China). When the vehicle enters the "urban forest" such as tunnel, garage, etc., it is difficult for it to capture GNSS satellite signals, INS will replace GNSS to provide real-time position information for the vehicle [1]. The positioning accuracy of the INS, however, depends on the accuracy of the hardware sensor and the length of use time. As time increases, INS errors will be gradually accumulated, leading to a gradual decline in positioning performance.

The performance of inertial navigation system in the unlocked state has declined rapidly over time. To solve this problem, scholars at home and abroad have proposed many improvement methods. The
commonly used method is to enhance the performance of the unlocked GNSS/INS combined system through BF (Bayesian filter), which includes the Kalman filter (KF) [2], the extended Kalman filter (EKF) [3] and particle filter (PF) [4]. This Kalman filter-based method continuously updates the measured noise variance by collecting estimated values and measured values. At the same time, it can process the noise variance of the system, which can effectively suppress white noise [5]. However, this type of system requires prior knowledge of the INS error model, which greatly increases the complexity of the system [6]. When these methods are used in combination with low-cost inertial sensors, they have poor stability and poor anti-noise effect [7].

With the explosive growth of data and the ever-increasing computing power of devices, artificial intelligence (AI) algorithms-based methods to enhance positioning accuracy are proposed. Position, Velocity and Azimuth Update Architecture (PVAVA) is a typical learning-based method [8]. It consists of three two-layer parallel neural networks, which provide the vehicle with position, speed and azimuth information during GPS interruption intervals. Artificial intelligence-based algorithms make full use of big data and powerful computing capabilities on the basis of traditional GNSS/INS combined systems, and have shown good advantages in research. But the above method is designed based on high-quality Inertial Measurement Unit (IMU) sensor data, which requires large-scale high-precision training data. And high-quality IMUs are generally large in size and costly.

The smart phone integrates IMU and many other sensors, which featured with high integration and low price, but its data accuracy is also low [9]. Based on the low-precision sensor data of smartphones, the DeepSense framework integrates convolutional neural networks (CNN) and recurrent neural network (RNN). One CNN is used to extract features of acceleration, angular velocity and magnetic field strength, and the other CNN is used to intelligently fuse these three types of data. Then input the output data of the second CNN into the RNN to explore the hidden time pattern in the measurement [10]. Due to the nature of RNN, the prediction accuracy decreases with time, and this method requires the training party to collect a large amount of data.

The biggest problem with the above smart phone-based solutions is how to collect and label accurate training data in large quantities. A common solution for the developer is to collect the route data of the driving area to complete the route coverage, which will take a lot of time and effort; another reliable solution is to aggregate a small amount of data collected by a large number of mobile users through crowdsourcing. However, the data aggregation process will cause serious data privacy leakage, especially when the data involves location information. In response to the above problems, this paper proposes a method, which uses federated learning to construct an accurate vehicle positioning model based on the smart phone's low-precision sensor data while protecting data privacy.

2. Federated Learning Positioning Model

2.1 Problem Description

Time interval \( T_s \) was assumed by navigation system. This system consists of a smartphone reference system, a vehicle reference system and a geodetic reference system. In this system, we can calculate the location of the vehicle through cell phone sensors. A variety of motion sensors are integrated in the smart phone. The linear acceleration \( \dot{a} \) can be obtained by the linear acceleration sensor, the \( \ddot{\omega} \) can be obtained by the gyroscope, and the \( \dot{q} \) can be obtained by the rotation vector sensor. Through coordinate transformation and four conversion functions \( F_1(\cdot) \sim F_4(\cdot) \) [11], the acceleration \( \ddot{a}_g \) of the geodetic reference frame can be obtained from \( \dot{a}, \dot{q} \) and \( \ddot{\omega} \). Then the vehicle position can be calculated by using the speed formula and the displacement formula.

\[
P_i^e = P_{i-1}^e + v_i^e T_s + a_i^e T_s^2 / 2 \tag{1}
\]

\[
v_i^e = v_{i-1}^e + a_i^e T_s \tag{2}
\]
However, the low accuracy of the smart phone sensor data will lead to a large acceleration error, resulting in insufficient accuracy of the speed and position. We adopted a data-driven method to construct a joint learning model to obtain a more accurate velocity vector to complete the vehicle position calculation.

2.2 Federated Learning
Data sets are currently an important part of AI. However, as AI is implemented in various application scenarios, the problem of data privacy leakage is becoming more and more serious. In this context, federated learning was proposed. The goal of federated learning is to achieve joint modeling and improve the effect of AI models on the basis of ensuring data privacy, security and legal compliance.

At present, federal learning is promoted to be applied to finance, medical and other industries, such as the WeBank FATE framework. In practical applications, island data has different data characteristics. Federated learning can be subdivided into three programs: horizontal federated learning, vertical federated learning, and federated transfer learning. The data features of participants in horizontal federated learning are aligned; the training samples of participants in vertical federated learning are aligned; while in federated transfer learning, the sample ID are not completely aligned with data features. In the framework of this article, the data characteristics of each participant are the same, so we choose the horizontal federated learning [12].

2.3 The structure of the Federated Learning Positioning Model
The vehicle positioning system based on federated learning proposed in this paper is shown in Figure 1. The system mainly includes a central node server and N edge nodes:

![Figure 1. The structure of the federated learning positioning model](image)

Edge nodes (such as smart phones) are located on the vehicle and have data collection, training, and forward inference functions. The moving range is a sub-range of the target area. By installing our data collection software, the node can obtain a variety of sensor data information and GNSS information during the movement. This data will be sent to the initial learning unit as the local data set of the edge node to participate in the local model training. After the system has completed training and obtained the federation model, the edge node will deploy the model. When the GNSS is in the unlocked state, the sensor data collected in real time is processed and input into the model to obtain the precise position of the vehicle.

The central node is assumed by the server in the network. It establishes a stable information transmission link with each edge node. During the training process, the central node uses the federated fusion algorithm (see Section 3.2 for details) to aggregate and update the parameters from each edge node to build a joint model.
Federated Training Process

Assume that the central node $S$ has the initialized global model parameter $w_0$, and at the beginning, it delivers the network model and $w_0$ to each edge node $C_i$. Then $C_i$ will train and update the parameter based on the local data set $D_i$ [13].

After a certain round training of $C_i$, the local model parameter is updated to $w_i$. At this time, the prediction loss of the model at the sample point $(x_j, y_j)$ is $f_j(w_i) = l(w_i, x_j, y_j)$. On node $C_i$, the loss function of the sample point is defined as:

$$f_i(w_i) = \frac{1}{|D_i|} \sum_{j \in D_i} f_j(w_i)$$  \hspace{1cm} (3)

Then each edge node uploads the updated parameter $w_i$ to the central node $S$. The central node selects the fusion algorithm according to different scenarios for processing, and obtains the model global parameter $w$. At this time, the sample point loss function of the global model can be expressed as [14]:

$$f(w) = \sum_{i=1}^{N} \left| D_i \right| \frac{1}{|D_i|} f_i(w_i) = E_i [f_i(w_i)]$$  \hspace{1cm} (4)

Among them, $|D_i|$ is the size of the local data set of edge node $C_i$, and $|D| = \sum_{i=1}^{N} D_i$ is the total number of samples of N nodes. The goal of the federated learning model is to find the global model parameter $w^*$ that minimizes the loss function $f(w)$,

$$w^* = \arg \min f(w)$$  \hspace{1cm} (5)

Therefore, the central node continues to send the global model parameters $w$ to the edge nodes to start a new round of calculation tasks until the expected goal or the specified round is reached. At the same time, in the federated training process, because only the upload and delivery of model parameters are involved, it avoids the direct aggregation and exposure of the edge node vehicle movement trajectory data set, and protects data privacy to a great extent.

3. Learning Unit and Federated Fusion Algorithm

3.1 Learning Unit

The learning unit of each edge node is two fully connected neural networks. The first neural network has six layers, and the output is the absolute value of the combined speed. The input feature is composed of the resultant velocity calculated by the INS, the angular velocity sequence and the acceleration sequence in the vehicle reference frame. The selection of these features is related to the speed.

The second neural network is similar to the first one. The output is the vehicle yaw angle, and its input feature is composed of the linear acceleration sequence and the angular velocity sequence in the vehicle reference frame, and the compensation yaw angle. The selection of these features is related to the vehicle yaw angle.

3.2 Federated Fusion Algorithm

The federated fusion algorithm selected by the central node $S$ will directly affect the final convergence and performance of the model. When the heterogeneity of edge node data increases, setting different federated fusion algorithms can improve the robustness and stability of convergence [15]
The FedAvg algorithm was proposed by McMahan et al. in 2017. In each round of global training, the FedAvg algorithm runs the SGD (Stochastic Gradient Descent) method locally to optimize the local target, and then averages the obtained model updates,

$$w = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i$$  \hspace{1cm} (6)

The FedProx algorithm is used in a realistic statistical heterogeneous environment (the data of each node is distributed in the form of Non-IID). It provides convergence guarantee under certain conditions and solves the problems of convergence robustness and stability in FedAvg [14]. In FedProx, the local solution target $f_i(w)$ is replaced with

$$\min_w h_i(w; w') = f_i(w) + \frac{\mu}{2} \|w - w'\|^2$$  \hspace{1cm} (7)

The above expression limits the local update to be similar to the current global model by adding a near-end clause, thereby effectively limiting the impact of the local update. This will lead to the inability to accurately resolve each local target, but experiments have shown that this operation can effectively resolve the statistical heterogeneity of data in the federal environment [15].

4. Experimental Result

4.1 Experimental Environment

In order to verify the feasibility of this solution in actual scenarios, we used Xiaomi 8 mobile phone to complete data collection at the Chinese University of Hong Kong (Shenzhen). Three participants were selected to participate in the federal training process. When testing the model, in order to simulate the GNSS loss-lock phenomenon, we manually removed the GNSS information for a certain length of time, such as leaving a GNSS signal every 60s. The experiment vehicle drove 13 laps on R1 and 4 laps on R2. For convenience, the data set is numbered 1-17, of which No. 1-13 is R1, and No. 14-17 is R2. Route data is distributed to three participants in two ways to simulate the data collection and model training results of vehicle users in two scenarios (users’ data are IID and Non-IID, see 4.2).

![Mi 8 and vehicle](image1)

![R1 and R2 in CUHK(Shenzhen)](image2)

Figure 2. Mi 8 and vehicle  \hspace{1cm} Figure 3. R1 and R2 in CUHK(Shenzhen)

In the training process, two federated learning fusion algorithms, FedAvg and FedProx, are applied respectively. Finally, the RMSE (Root Mean Squared Error) with the actual path is used to evaluate the positioning performance. The actual path is subject to the data without removing the GPS signal.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N} \|\hat{P}_j - P\|^2}{N}}$$  \hspace{1cm} (8)
Where N is the number of GPS positioning points updated at a frequency of 1 Hz in the actual path. 
\( P_j = [x_j, y_j] \) is the position of the j-th GPS positioning point in the test trajectory, and 
\( \hat{P}_j = [\hat{x}_j, \hat{y}_j] \) is the predicted point of the model.

At the same time, in order to examine the accuracy of the method proposed in this paper, we selected two existing typical learning-based methods PVAUA, DeepSense and INS methods for comparison.

4.2 Experimental Result

The first scenario is an IID type scenario. Three participants use the four data in No. 1-12 respectively for training. The second scenario is the Non-IID scenario, which is more common than the first scenario. The data sets of the three users come from different routes. The amount of data for each user is not necessarily the same. In the testing phase, No.13 of R1 and No.14 of R2 are the test sets.

**Table 1. Data distribution of the two scenarios**

| Numbers | User a | User b | User c |
|---------|--------|--------|--------|
| Scenario 1 | R1 | 4 | 4 | 4 |
| | R2 | 0 | 0 | 0 |
| Scenario 2 | R1 | 0 | 2 | 6 |
| | R2 | 2 | 0 | 1 |

**Figure 4.** R1-No.13 the scenarios

**Figure 5.** R1-No.14 the scenarios

**Table 2.** Data distribution of the two scenarios

| Test Route | Method | IID-RMSE(m) | Non-IID-RMSE(m) |
|------------|--------|-------------|-----------------|
| R1-No.13   | INS    | 99.84       | 99.84           |
|            | FedAvg | 11.41       | 37.18           |
|            | FedProx| 12.17       | 11.04           |
|            | PVAUA  | 125.62      | 125.58          |
|            | DeepSense | 18.88      | 193.03          |
| R2-No.14   | INS    | 71.06       | 71.06           |
|            | FedAvg | 8.15        | 34.15           |
|            | FedProx| 11.84       | 19.45           |
|            | PVAUA  | 135.18      | 135.10          |
|            | DeepSense | 161.65     | 45.88           |
From Figures 4, 5 and Table 2, we can conclude that in the IID scenario, the RMSE value of the FedAvg and FedProx is about 12m in the travel of about 1.7km. Compared with the INS, PVAUA and DeepSense methods, they can get better results. Compared with the two federated fusion algorithms, FedProx sacrifices a small amount of performance in order to ensure a certain degree of convergence, which is slightly inferior to FedAvg. At the same time, according to the prediction results of R2-No.14, the model can predict the new routes very well and has good generalization performance. DeepSense directly predicts the position coordinates. In the prediction of R1-No.13, more accurate results can be obtained from the familiar trajectory during training. In the result of R2-No.14, because of the unfamiliarity with the new trajectory, the prediction error is significant. PVAUA cannot be applied well because of the low accuracy of sensor data, and the prediction results have a great of errors.

In the Non-IID scenario, due to the different distribution of the training data of each user and the different amount of data, the RMSE of the FedAvg algorithm is about 37m, although it is still better than the INS and the two existing methods, as shown in Figure 5. The FedProx prediction effect is slightly lower than that of the model trained in the IID scene, it achieves better results than FedAvg when the training parameters are the same. In the Non-IID scenario, PVAUA still has a large error due to sensor accuracy limitations, and DeepSense cannot obtain better results in the face of data sets with mixed paths.

In order to eliminate the contingency of the results and better evaluate the actual effect of the model in this paper, we have carried out a comparative evaluation of the error trend of various methods under different GPS intervals. In the other case of the same parameters, we used R1-No.13 to test the RMSE performance of the two federated fusion algorithm models under different GPS interruption duration scenarios, and the results are shown in Table 3.

| GPS Interval (s) | IID (m) | | Non-IID (m) | |
|------------------|---------|------------------|------------------|
|                  | Fed Avg | Fed Prox | PVAUA | Deep Sense | Fed Avg | Fed Prox | PVAUA | Deep Sense | INS (m) | |
| 30               | 5.60    | 7.22    | 66.83  | 12.85    | 14.44    | 6.75    | 66.84  | 100.76    | 39.89   |
| 45               | 7.39    | 9.63    | 98.29  | 13.84    | 25.76    | 10.01   | 98.38  | 141.72    | 51.00   |
| 60               | 11.41   | 12.17   | 125.62 | 18.88    | 37.18    | 11.04   | 125.58 | 193.03    | 99.84   |
| 75               | 11.32   | 10.57   | 147.24 | 22.57    | 39.61    | 20.97   | 147.34 | 223.24    | 78.95   |
| 90               | 11.25   | 17.75   | 179.72 | 15.78    | 63.08    | 27.25   | 179.84 | 265.35    | 80.63   |
| 105              | 16.42   | 20.58   | 212.00 | 31.44    | 74.43    | 25.48   | 212.02 | 327.59    | 152.50  |
| 120              | 16.56   | 13.38   | 232.39 | 15.77    | 76.40    | 31.94   | 232.35 | 346.97    | 249.32  |
| 135              | 15.88   | 18.48   | 251.31 | 26.60    | 72.38    | 39.94   | 251.64 | 410.18    | 169.59  |
| 150              | 21.04   | 28.72   | 230.08 | 39.36    | 60.17    | 43.01   | 230.26 | 370.22    | 248.84  |

5. Conclusion
This article introduces a vehicle positioning method based on joint learning and low-precision sensor data. By selecting appropriate data features and using two fully connected neural networks for parallel processing, more accurate vehicle position information than existing methods can be obtained when the GNSS signal is out of lock. In this scheme, we chose two joint learning fusion algorithms FedAvg and FedProx. Taking three participants as an example, corresponding solution algorithms are given for IID and Non-IID scenarios. Based on a complex simulation environment, the results show that when the GNSS update interval is shortened, the accuracy of the model will increase rapidly. In the future intelligent transportation system, the GNSS calibration function can be replaced by smart devices (such as smart street lights) on both sides of the road. For new scenarios, we will further design new vehicle navigation training models and network structures (such as LSTM) to further improve system performance.
Acknowledgments
Foundation Item(s): National Natural Science Foundation of China (No.92067202, No.61701426, No.61725101, No.6196113039, No.U1834210), Royal Society Newton Advanced Fellowship (No.NA191006), State Key Lab of Rail Traffic Control and Safety(No.RCS2020ZT010, No.RCS2019ZZ007), National Key R&D Program of China(No.2016YFE0200900).

References
[1] Yang Y. Tightly coupled MEMS INS/GPS integration with INS aided receiver tracking loops . Department of Geomatics Engineering, 2008.
[2] Grewal M S, Weill L R, Andrews A P. Global positioning systems, inertial navigation, and integration. John Wiley & Sons, 2007.
[3] Särkkä S. Bayesian filtering and smoothing. Cambridge University Press, 2013.
[4] Carvalho H, Del Moral P, Monin A, et al. Optimal nonlinear filtering in GPS/INS integration . IEEE Transactions on Aerospace and Electronic Systems, 1997, 33(3): 835-850.
[5] Liu F, Sun X, Xiong Y, et al. Combination of iterated cubature Kalman filter and neural networks for GPS/INS during GPS outages . Review of Scientific Instruments, 2019, 90(12): 125005.
[6] Malleswaran M, Vaidehi V, Sivasankari N. A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques . Aerospace Science and Technology, 2014, 32(1): 169-179.
[7] Abdolkarimi E S, Mosavi M R, Abedi A A, et al. Optimization of the low-cost INS/GPS navigation system using ANFIS for high speed vehicle application//2015 Signal Processing and Intelligent Systems Conference (SPIS). IEEE, 2015: 93-98.
[8] Chiang K W. INS/GPS integration using neural networks for land vehicular navigation applications. Canada: University of Calgary, 2004.
[9] Perlmutter M, Robin L. High-performance, low cost inertial MEMS: A market in motion!//Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium. IEEE, 2012: 225-229.
[10] Yao S, Hu S, Zhao Y, et al. Deepsense: A unified deep learning framework for time-series mobile sensing data processing //Proceedings of the 26th International Conference on World Wide Web. 2017: 351-360.
[11] Qi J, Li H, Yin F, et al. Navigation with low-sampling-rate GPS and smartphone sensors: A data-driven learning-based approach . 2019.
[12] Yang Q. AI and Data Privacy Protection: The Way to Federated Learning . Journal of Information Security Research, 2019, 5(11): 961-965.
[13] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications . ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.
[14] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data //Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.
[15] Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks . arXiv preprint arXiv:1812.06127, 2018.