RESEARCH ARTICLE

Parameter Adaptive Non-Model-Based State Estimation Combining Attention Mechanism and LSTM

Xuebo Jin[1,*], **Tianxiao Sun**[1], **Wei Chen**[1], **Huijun Ma**[2], **Yeqing Wang**[2] **and Yusen Zheng**[1]

[1] School of Computer Science and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China
[2] National Engineering Laboratory for Agri-product Quality Traceability, BTBU, Beijing, China

Abstract

Nowadays, state estimation is widely used in fields such as autonomous driving and drone navigation. However, in practical applications, it is difficult to obtain accurate target motion models and noise covariance. This leads to a decrease in the estimation accuracy of traditional Kalman filters. To address this issue, this paper proposes an adaptive model free state estimation method based on attention parameter learning module. This method combines Transformer's encoder with Long Short Term Memory Network (LSTM), and obtains the system's operational characteristics through offline learning of measurement data without modeling the system dynamics and measurement characteristics. In addition, based on the output of the attention learning module, the expectation maximization (EM) algorithm is used to estimate the system model parameters online, and a Kalman filter is used to obtain state estimation. This paper was validated using the GPS trajectory path dataset, and the experimental results showed that the proposed parameter adaptive model free state estimation method has better estimation accuracy than other models, providing an effective method for using deep learning networks for state estimation.

Keywords: state estimation, kalman filter, transformer, LSTM.

1 Introduction

State estimation is an important research direction in both control theory and machine learning, with wide applications in fields such as micro air vehicle navigation [1], robotics [2], and autonomous driving [3]. Its core objective is to estimate the actual state of a target using observed data. Due to the influence of random noise on system states and observations, Rudolf Emil Kalman proposed the Kalman Filter (KF), which describes the relationship between states and observations using a state model [4]. It avoids the need to store all historical data, thereby reducing the computational and storage resource requirements. However, KF cannot estimate states for nonlinear system models. To address this limitation, Kalman and Bucy introduced the Extended Kalman Filter (EKF), which linearizes nonlinear systems before utilizing a generalized Kalman filter for state estimation. EKF has since been applied to various domains including smart grid dynamic state estimation [5]. Nevertheless, EKF's

Academic Editor:
Jing Na

Submitted: 25 December 2023
Accepted: 22 May 2024
Published: 29 May 2024

Vol. 1, **No.** 1, 2024.
10.62762/TIS.2024.137329

***Corresponding author:**
✉ Xuebo Jin
jinxuebo@btbu.edu.cn

Citation
Jin, X., Sun, T., Chen, W., Ma, H., Wang, Y., & Zheng, Y. (2024). Parameter Adaptive Non-Model-Based State Estimation Combining Attention Mechanism and LSTM. *ICCK Transactions on Intelligent Systematics*, 1(1), 40–48.

© 2024 ICCK (Institute of Central Computation and Knowledge)

approximation often leads to reduced accuracy due to discarding higher-order derivative components during linearization. To tackle this issue, the Unscented Kalman Filter (UKF) approximates probability distributions using a set of determined sigma points containing mean and covariance, making it more efficient for highly nonlinear systems. Xiong et al. [6] further applied UKF to fault detection in nonlinear systems, demonstrating its effectiveness beyond standard state estimation. However, UKF may lose some statistical properties of the posterior distribution when the sampling dimension exceeds three, resulting in decreased estimation accuracy. Therefore, the Cubature Kalman Filter (CKF) is based on the third-order spherical radial volume criterion and uses cubature points to approximate the mean and covariance of nonlinear system states. Arasaratnam and Haykin [7] further extended this framework to cubature Kalman smoothers for improved retrospective state estimation. Although the filtering accuracy of the CKF is higher, the algorithm discards some approximation errors when estimating nonlinear systems, which may lead to filtering that does not meet quasi-consistency. As a result, the CKF cannot accurately estimate the true value of the state.

In order to mitigate the dependence of traditional filtering algorithms on accurate motion model parameters, Ghahramani and Hinton proposed the EM-KF algorithm, which combines Kalman filtering for linear dynamic system state estimation with the EM algorithm for parameter estimation [8]. To address the complex dynamic relationship between target states and measurement data, researchers have proposed a data-driven deep learning approach for state estimation. This method utilizes large amounts of measurement data and corresponding true state values to train recursive neural networks end-to-end, learning the intricate relationship between measurement data and states for effective state estimation. Recurrent Neural Networks (RNNs) are commonly used for analyzing time-series data, and the LSTM network, an excellent variant of RNNs, was introduced by Hochreiter and Schmidhuber in 1997 [9]. LSTM incorporates a hidden state unit to store important information from previous positions in the sequence and employs three gate control units to determine the importance of neural input information. These gate control units, utilizing different activation functions and computation methods, effectively address issues such as gradient explosion, thereby demonstrating significant advantages in long-term sequence modeling compared to traditional RNNs. While deep learning methods exhibit strong learning capabilities, neural networks often encounter challenges in training due to the receipt of large amounts of information, some of which may be irrelevant. To address this issue, attention mechanisms have been introduced to simulate human attention, allowing neural networks to focus on useful information while disregarding irrelevant data. In recent years, Transformer has emerged as a novel attention mechanism [10]. It effectively addresses long-term dependency issues in LSTM networks, thereby improving modeling capabilities for complex nonlinear data and colored noise.

In order to solve the problems faced by estimation methods in applications, this paper proposes a new algorithm Transformer-LSTM with EM-based Filter (TL-EF), which mainly consists of two steps: offline training and online estimation. The first step is to use the Transformer multi head self attention mechanism and LSTM to learn the complex relationships between system states and measurements, and obtain the statistical characteristics of its complex motion. The second step is to provide more accurate filtering parameters for the Kalman filter based on the network output and the EM method.

2 Methodology

2.1 Offline Training Of System Model-TL Learner

With existing filter model-based state estimation methods, the model needs to be used in conjunction with each other and the state estimation methods. The current target motion state is estimated based on the measurement information observed by the sensors, combined with the a priori knowledge of the tracked target. In order to provide an accurate description of the current state of the maneuvering target and the measurement sensors, the motion process model and measurement model of the maneuvering target need to be established. Usually, the linear system model is as follows

$$x_k = Ax_{k-1} + \omega_k \tag{1}$$
$$y_k = Cx_k + \nu_k \tag{2}$$
$$\omega_k \sim N(0, Q), \nu_k \sim N(0, R) \tag{3}$$
$$x_0 \sim N(m_0, P_0), k = 1, 2, ..., N \tag{4}$$

where x_k represents the state, y_k represents the measurement data, A represents the state transition matrix, ω_k represents the state noise, Q represents

the covariance of the state noise, C represents the measurement matrix, ν_k represents the measurement noise, R represents the covariance of the measurement noise, and m_0 and P_0 represent the mean and covariance of the initial state, respectively. Typically, these parameters are modeled based on historical knowledge or system mechanics, referred to as initial parameters in this paper.

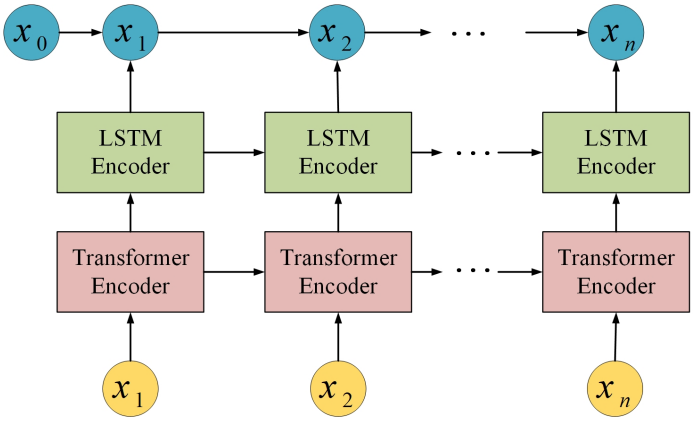


Figure 1. Model studies of the system.

Offline training involves training a system model learner based on attention and LSTM with the initial parameters of the system (as shown in Figure 1), By combining the encoder structure of Transformer and LSTM, the model learns from observation data without the need to model the dynamics and measurement characteristics of the system, The training of the neural network enables the learning of the motion characteristics of the system. The reason why Transformer can capture long-term dependencies is because it integrates multi-head self-attention and residual connections, which enhances the depth of model training and reduces the risk of overfitting. Combining LSTM with the Transformer encoder can strengthen the structural advantages of the Transformer encoder and its sequence modeling capabilities. Since Transformer can capture salient features that LSTM cannot capture, their combination can better characterize the motion sequences. This model can fully utilize the known information of the system's mechanistic model, while training the network model to capture the complex dependencies of the system from historical measurement data. Compared to relying solely on a mechanistic model, this model can better describe the dynamics and measurement relationships of real-world systems.

Drawing inspiration from the cognitive attention mechanism in humans, multi-head self-attention generates multiple attention heads in parallel within the network [11]. The inputs to this module are the observed data $y_{1:N} = [y_1, y_2, ..., y_N]$, the query vectors $Query$ related to the attention task, and the feature information Key and $Value$ represented in key-value pairs. The vectors $Query$, Key, and $Value$ are obtained through linear transformations of the input observed data $y_{1:N}$, and the attention process can be represented as

$$Query = W_Q y_{1:N} \tag{5}$$

$$Key = W_K y_{1:N} \tag{6}$$

$$Value = W_V y_{1:N} \tag{7}$$

$$\begin{aligned} &Attention(Query, Key, Value) \\ &= \text{softmax}\left(\frac{QueryKey^T}{\sqrt{d_k}}\right) Value \end{aligned} \tag{8}$$

where matrices W_Q, W_K, and W_V are trainable parameter projection matrices, while d_k represents the feature dimension of Key. Based on input data $Query$ and Key, the dot product can be obtained. Then, using the function described in $softmax$, the weights corresponding to each element in input data $Value$ can be obtained. d_k serves as a scaling factor for the dot product, preventing it from becoming too large and facilitating faster learning.

The essence of the multi-head self-attention mechanism is a linear transformation of the concatenated results of multiple attention computations. This mechanism allows the model to utilize different feature information obtained at different positions, thereby increasing feature diversity. The calculation method of multi-head self-attention is shown below.

$$\begin{aligned} &MHA(Query, Key, Value) = \\ &Concat(head_1, ..., head_t)W^o \end{aligned} \tag{9}$$

$$\text{head}_i = \text{softmax}\left(\frac{Query_i Key_i^T}{\sqrt{d_k}}\right) Value_i \tag{10}$$

where t represents the total number of heads, W^o denotes the weight matrix used to ensure alignment with the target dimension, $Concat$ indicates the vector concatenation operation, and $head_i$ represents the features of the i head.

This paper adopts the Transformer encoder structure to encode the latent dynamical properties of the observed data. Utilizing the multi-head attention

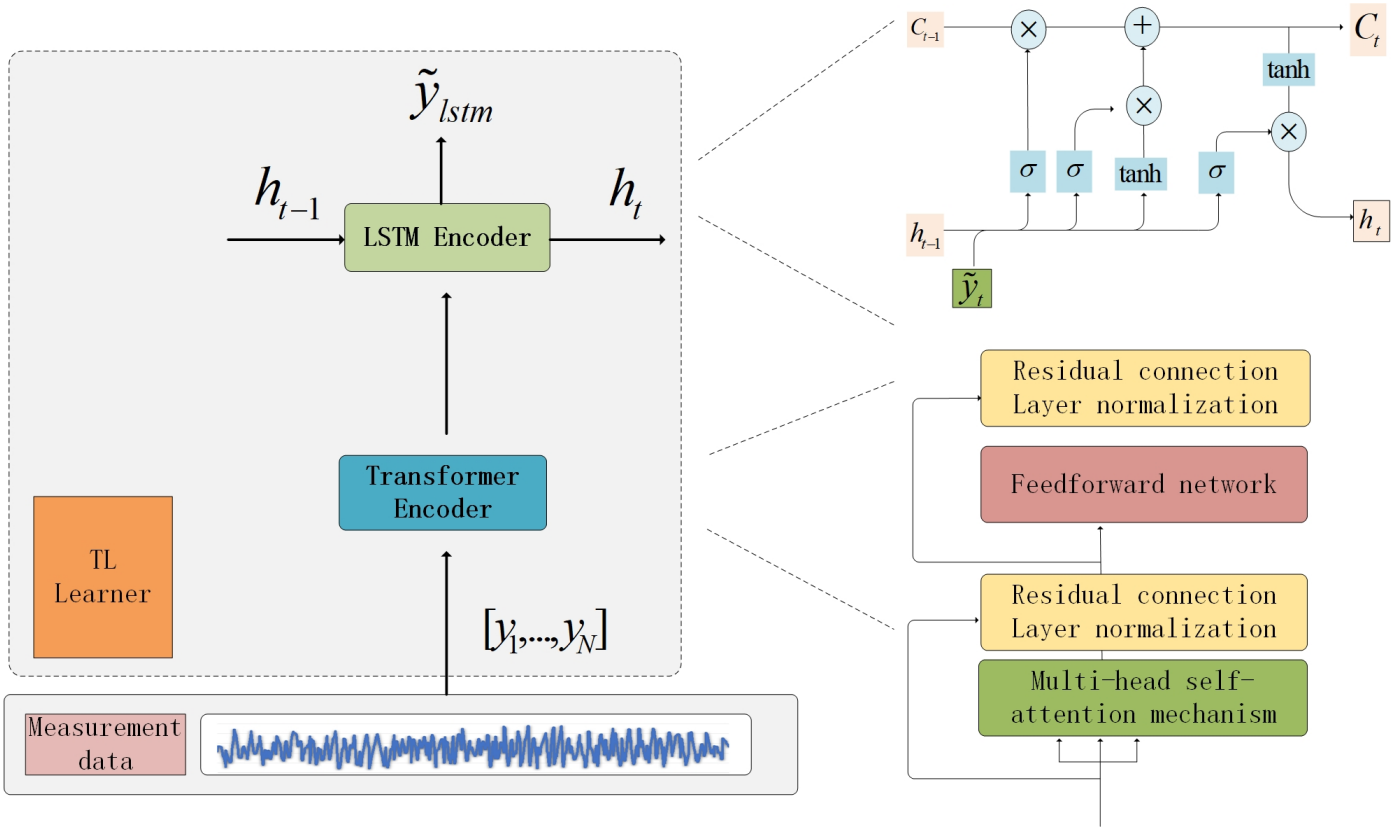


Figure 2. A system model learner based on attention and LSTM.

mechanism, it constructs attention information with higher dimensionality and multiple channels, thereby exploring rich information. The attention-based learning module takes the observed data as input into the Transformer encoder module. Within the encoder, positional encoding is applied to the observed data before being fed into the multi-head self-attention layer. To prevent network degradation and accelerate convergence, the encoder's structure includes residual connections, layer normalization, and feedforward neural networks. The latent encoded features are obtained through the Transformer encoder module, After the Transformer, we use LSTM to learn longer dependencies among the measurement data as follows.

$$\tilde{y}_{\text{transformerEncoder}} = \text{TransformerEncoder}(y_{1:N}) \quad (11)$$

$$\tilde{y}_{\text{lstm}} = \text{LSTM}(\tilde{y}_{\text{TransformerEncoder}}, h_{t-1}) \quad (12)$$

where h_{t-1} represents the hidden state from the previous time step.

The system model learner (as shown in Figure 2) achieves the learning of system dynamics through offline training. The input-output relationship pairs consist of observation data and the reference state of the system. Based on the Transformer encoder structure with multi-head self-attention and LSTM, the model can learn the latent features of the data, eliminating the need for modeling system dynamics and measurement characteristics.

2.2 Online Estimation Based On The EM Algorithm

The network model trained offline still exhibits biases when applied to the actual system. During the estimation process, online estimation is performed using the EM algorithm to update model parameters in real-time. Specifically, measurement data is input into the pre-trained attention-LSTM learning module to obtain dynamically characterized data reflecting the motion sequence. Subsequently, Kalman filtering is employed for recursive estimation online. Furthermore, the EM algorithm is utilized to update current parameters such as the state transition matrix, measurement matrix, state noise, and measurement noise variances.

In the Algorithm 1, $z_{0:N}$ represents the latent variable, θ represents the unknown parameters to be estimated, and θ^n represents the parameters at the nth iteration, when there are latent variables in the probability density function (PDF), the expectation maximization (EM) algorithm iterates E-steps and M-steps to calculate the maximum likelihood estimate (MLE)

Algorithm 1: Expectation-Maximization (EM) Algorithm

Input: Initial parameter θ^0, tolerance ε, maximum iteration n_m
Output: Estimated parameter θ^*
repeat
  /* E-step */
  Compute $\theta^{(n)} = \mathbb{E}[\log p(z_{0:N}, y_{1:N}|\theta) \mid y_{1:N}, \theta^{(n)}]$
  $Q(\theta, \theta^{(n)}) \leftarrow \theta^{(n)}$
  /* M-step */
  $\theta^{(n+1)} \leftarrow \arg\max_\theta Q(\theta, \theta^{(n)})$
until $|\theta^{(n+1)} - \theta^{(n)}| < \varepsilon$ ***or*** $n \geq n_m$
return $\theta^* \leftarrow \theta^{(n+1)}$

of the parameters, for state estimation, maximizing the logarithm of the probability density function of observation $log\ p(z_{0:N}, y_{1:N}|\theta)$ is equivalent to maximizing parameter Q.

The system's state to be estimated is denoted by x_k, and the measurement is denoted by y_k, with the probability distribution as follows:

$$p(x_k|x_{k-1}) = \frac{\exp\left\{-\frac{1}{2}[x_k - Ax_{k-1}]^T Q^{-1}[x_k - Ax_{k-1}]\right\}}{(2\pi)^{-u/2}|Q|^{-1/2}} \tag{13}$$

$$p(y_k|x_k) = \frac{\exp\left\{-\frac{1}{2}[y_k - Cx_k]^T R^{-1}[y_k - Cx_k]\right\}}{(2\pi)^{-v/2}|R|^{-1/2}} \tag{14}$$

$$p(x_0) = \frac{\exp\left\{-\frac{1}{2}[x_0 - m_0]^T {P_0}^{-1}[x_0 - m_0]\right\}}{(2\pi)^{-u/2}|P_0|^{-1/2}} \tag{15}$$

where u and v are the number of state dimensions and observation dimension in the state. Under the assumption of conditional independence, we can obtain

$$\begin{aligned}\mathcal{Q}(\theta, \theta^{(n)}) = &-\frac{1}{2}\ln|2\pi P_0| - \frac{N}{2}\ln|2\pi Q| - \frac{N}{2}\ln|2\pi R| \\ &- \frac{1}{2}tr\{{P_0}^{-1}[P_{0|N} + (m_{0|N} - m_0)(m_{0|N} - m_0)^T]\} \\ &- \frac{1}{2}\sum_{k=1}^{N} tr\{Q^{-1}\mathbb{E}[(x_k - Ax_{k-1})(x_k - Ax_{k-1})^T|y_{1:N}]\} \\ &- \frac{1}{2}\sum_{k=1}^{N} tr\{R^{-1}\mathbb{E}[(y_k - Cx_k)(y_k - Cx_k)^T|y_{1:N}]\}\end{aligned} \tag{16}$$

$$\frac{\partial \mathcal{Q}(\theta, \theta^{(n)})}{\partial \theta} = 0 \tag{17}$$

For the parameter estimation of the model, we utilize both the original measurement data $y_{[1,N]}$ and the data $\tilde{y}_{lstm}$ estimated by the attention learning module trained offline in the EM algorithm. Through the multi-head self-attention and long short-term memory (LSTM) network, we can capture long-term dependencies among the data and the underlying correlations between them. This enables a better estimation of the parameters of the filter model.

$$\begin{aligned}A = &(\sum_{k=1}^{N-1}\mathbb{E}[\hat{x}_{k|k-1}\hat{x}_{k-1|k-1}^T])/ \\ &(\sum_{k=1}^{N-1}\mathbb{E}[\hat{x}_{k-1|k-1}\hat{x}_{k-1|k-1}^T])\end{aligned} \tag{18}$$

$$C = (\sum_{k=0}^{N-1} y_{\text{lstm},k}\mathbb{E}[\hat{x}_{k|k}]^T)/(\sum_{k=0}^{N-1}\mathbb{E}[\hat{x}_{k|k}\hat{x}_{k|k}^T]) \tag{19}$$

$$\begin{aligned}Q = &\frac{1}{N-1}\sum_{k=0}^{N-2}(\mathbb{E}[\hat{x}_{k+1|k+1}] - A\mathbb{E}[\hat{x}_{k|k}])(\mathbb{E}[\hat{x}_{k+1|k+1}] \\ &- A\mathbb{E}[\hat{x}_{k|k}])^T + AVar[\hat{x}_{k|k}]A^T + Var[\hat{x}_{k+1|k+1}] \\ &- Cov(\hat{x}_{k+1|k+1}, \hat{x}_{k|k})A^T - ACov(\hat{x}_{k+1|k+1}, \hat{x}_{k|k})\end{aligned} \tag{20}$$

$$\begin{aligned}R = &\frac{1}{N}\sum_{k=0}^{N-1}\left[y_{\text{lstm},k} - C\mathbb{E}[\hat{x}_{k|k}]\right]\left[z_k - C\mathbb{E}[\hat{x}_{k|k}]\right]^\top \\ &+ C\text{Var}[\hat{x}_{k|k}]C^\top\end{aligned} \tag{21}$$

where $\hat{x}_{k|k}$ represents the estimation of the filter at step k, by recursively applying the Kalman filter, the estimated trajectory of the output state $\hat{x}_{k|k}$ is obtained. When using the Kalman filter, A represents the state transition matrix calculated from Equation (18), ω_t represents the state noise, Q represents the state noise covariance calculated from Equation (20), C represents the measurement matrix calculated from Equation (19), v_k represents the measurement noise, and R represents the measurement noise covariance calculated from Equation (21).

3 Experiments

3.1 Experimental Setup and Evaluation Metrics

In this study, we utilized the open-source deep learning library PyTorch to build neural network models. All experiments were conducted on a PC machine with an Intel Core i5-7300HQ @2.50 GHz CPU processor and 8GB RAM. The neural network models were designed with default parameters in PyTorch for initialization, including deterministic weight initialization. Additionally, a fixed random seed was set for all experiments. The experiments involved comparing different filter parameters and various neural network models. We conducted the following cases:

(1) Case 1: Comparison based on publicly available trajectory datasets.

(2) Case 2: Comparison based on GPS trajectory datasets of vehicles in Beijing.

To evaluate the predictive performance of different models, four evaluation metrics were used to assess their performance comprehensively: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), and Pearson Correlation Coefficient (R). The four evaluation indicators have the following meanings: for (RMSE), (MAE), and (SMAPE), smaller values indicate higher accuracy of model estimation. On the contrary, for the evaluation metric (R), the larger the value, the better the fit between the reference and estimated trajectory.

3.2 Results and Discussion

3.2.1 Comparison based on public trajectory datasets

In this context, we used a publicly available trajectory dataset from an online source[1]. The simulated dataset consists of 4200 trajectory paths containing 201 data points. The simulated data range is in a two-dimensional plane from 0 to 100, and pink noise signals are added. The proposed estimation method is compared with traditional KF [4], EM-KF [8], and process models such as CA, CV, Singer, Current Statistical. Traditional Kalman filters are primarily designed for linear systems, assuming Gaussian models for both system and measurement noise. The CV and CA models treat the acceleration of the maneuvering target and its derivatives as zero-mean Gaussian noise, while the Singer model considers the maneuvering target and acceleration as exponentially correlated zero-mean colored noise. However, assuming zero mean is not suitable for describing the motion state of actual maneuvering targets. The current statistical models assume that the target motion follows a constant velocity process, and observations are linear combinations of state variables, enabling effective state estimation through minimum mean square estimation. On the other hand, the EM-Kalman filter combines the Expectation-Maximization (EM) algorithm to update system model parameters and estimate hidden state variables iteratively. When the underlying system model is inaccurate or uncertain, the EM-Kalman filter can outperform traditional Kalman filters, making it a popular method in practice. By comparing our proposed estimation method with these widely used approaches, we aim to demonstrate its superior performance in tracking highly maneuverable targets.

We compared the trajectory tracking performance of different filter models, and the results are shown in Table 1. The proposed models have RMSE, MAE, SMAPE, and R of 1.71, 1.35, 4.36, and 0.99, respectively. Compared with the other six models, our proposed TL-EF model reduced RMSE by 78.2%, 20.5%, 34.5%, 30.5%, 70.2%, and 72.1%, MAE by 66.5%, 10.0%, 11.8%, 10.6%, 59.8%, and 62.0%, and SMAPE by 63.0%, 27.1%, 33.6%, 30.7%, 68.3%, and 57.6%, respectively. From the index of R, the R value of the proposed model represents the best fit between the estimated value and the true value. The analysis of the above experimental data shows that our proposed model eliminates the complex manual parameter adjustment process and outperforms other filter models in estimation accuracy and result fitting, further proving its applicability in the field of state estimation.

3.2.2 Based on GPS track data sets in Beijing

We utilize the Geolife Beijing vehicle GPS trajectory dataset[2]. Among these are 13987 GPS points. In which, 80% was selected as the training set for the model, while the remaining 20% was allocated for testing. Figure 3 illustrates an instance of trajectory within this dataset.

Neural network models have significant advantages in estimation applications, especially when dealing with complex nonlinear problems and large-scale data. However, different types of neural network models exhibit obvious differences in estimation accuracy, computational efficiency, and model parameters. Therefore, we applied the

[1] https://github.com/btbuIntelliSense

[2] https://www.gpsvisualizer.com/

Table 1. Estimation results of different filtering models.

Model	Model parameters	RMSE	MAE	SMAPE	R
KF		7.88	4.04	11.79	0.97
CA	T=0.5 R=400 Qq=100	2.83	1.64	7.13	0.98
CA	T=1 R=1600 Qq=200	2.15	1.50	5.98	0.99
CV	T=0.5 R=400 Qq=100	3.18	1.62	7.00	0.97
CV	T=1 R=1600 Qq=200	2.61	1.53	6.57	0.98
Singer	T=0.5 R=400 Qq=100 A=2	3.24	1.64	7.10	0.98
Singer	T=1 R=1600 Qq=200 A=1	2.46	1.51	6.29	0.98
Current Statistical	T=0.2 R=50 a=0.5 xamax=50	5.76	4.30	15.00	0.93
Current Statistical	T=0.1 R=25 a=0.2 xamax=30	5.74	3.36	13.76	0.93
EM-KF		6.14	3.55	10.28	0.98
TL-EF		1.71	1.35	4.36	0.99

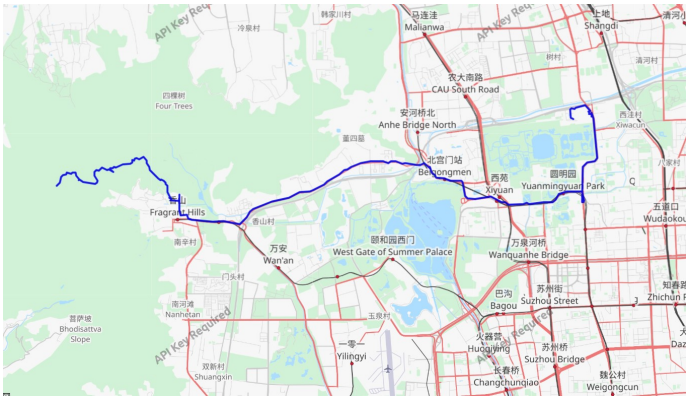


Figure 3. Example GPS trajectories from the Geolife Beijing vehicle dataset.

Table 2. Evaluation of estimated performance evaluation of GPS track data for vehicles in Beijing.

Model	RMSE	MAE	SMAPE	R
LSTM	3.84	3.97	3.75	0.99
GRU	6.83	5.52	4.29	0.99
CNN-LSTM	5.92	4.47	3.87	0.99
ConvLSTM	5.76	4.69	3.15	0.99
PFVAE	1.82	1.32	0.97	0.99
TL-EF	1.32	0.89	0.50	0.99

TL-EF model to the Beijing vehicle trajectory dataset and conducted comparative experiments with five other models, including LSTM [9], GRU [12], CNN-LSTM [13], ConvLSTM [14], and PFVAE [15]. CNN-LSTM combines convolutional layers for local feature extraction with LSTM for sequential modeling [13], and is here adapted for trajectory state estimation. For each comparison model, we set corresponding parameters. Taking GRU as an example, we used 50 hidden neural units, a learning rate of 0.0001, 1000 iterations, and a batch size of 30. It consisted of two network layers, each containing 50 hidden units. For the sliding window strategy of comparison models, we adopted the same recursive estimation strategy as the filter model. Table 2 presents the estimation results of the six models used in the experiments.

In Table 2, compared with LSTM, GRU, CNN-LSTM, ConvLSTM, and PFVAE models, our proposed TL-EF model has reduced RMSE by 65.6%, 80.7%, 77.7%, 77.1%, and 27.5%, respectively, and reduced MAE by 77.6%, 83.9%, 80.1%, 81.0%, and 32.6%. SMAPE decreased by 86.7%, 88.3%, 87.1%, 84.1%, and 48.5%. In addition, the R index obtained by the TL-EF model is comparable to that of LSTM, GRU, CNN-LSTM, ConvLSTM, and PFVAE models.

In Figure 4, we observe scatter plots showing each model trajectory's estimated values and true values. The lines corresponding to these estimates closely align with the true latitude and longitude values of the respective scatter points, accurately reflecting the model's estimation results.

The analysis above demonstrates that the TL-EF model exhibits high precision and stability in estimating GPS

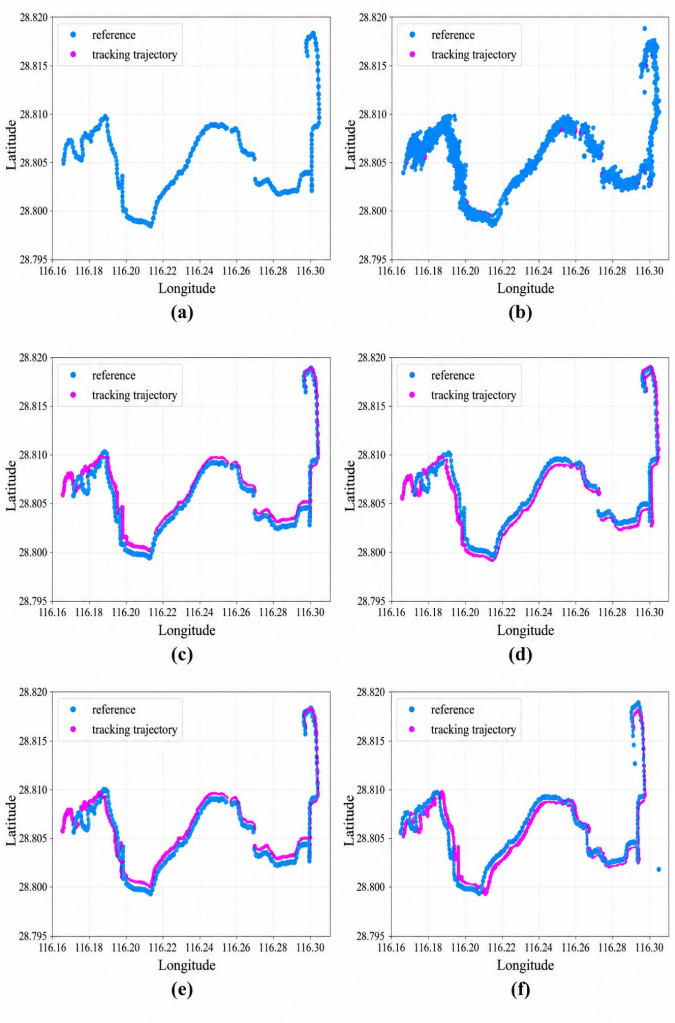


Figure 4. Comparison of vehicle GPS movement trajectories in the Beijing area estimated by different models. (a) Trajectory estimated using TL-EF, (b) Trajectory estimated using PFVAE, (c) Trajectory estimated using LSTM, (d) Trajectory estimated using GRU, (e) Trajectory estimated using CNN-LSTM, (f) Trajectory estimated using ConvLSTM.

trajectories, outperforming traditional and deep learning models. This effectiveness makes it a valuable tool for trajectory-tracking applications. The introduction of this model holds significant importance for state estimation methods and provides reliable technical support for further research endeavors.

4 Conclusion

State estimation is an important research field, although widely used, it still faces many challenges. In practical applications, GPS navigation signals are prone to signal weakness or complete occlusion, and it is difficult to model the system when dealing with highly maneuvering targets or complex systems containing colored noise. Therefore, there is an urgent need to develop novel and effective estimation algorithms to address this challenge and improve state estimation accuracy. To address this challenge, this paper proposes an adaptive Kalman algorithm for state estimation, which utilizes a learning module based on Transformer and LSTM. By combining the Transformer encoder structure with LSTM, a neural network parameter learning module was designed to learn measurement data in offline state, in order to obtain the motion characteristics of the system without modeling the system dynamics and measurement characteristics. Subsequently, based on the output of the neural network parameter learning module, the EM algorithm is used to obtain model parameters suitable for estimating the Kalman filter estimation, and the Kalman filter is used for state estimation. However, the model has two main limitations. First, the Transformer-LSTM architecture involves significantly higher computational cost than traditional Kalman filters, making real-time deployment challenging on resource-constrained platforms. Second, the offline training phase assumes that training data is representative of the deployment environment; performance may degrade for measurement noise distributions that differ substantially from those seen during training.

Future research can explore using different deep learning networks or adjust the trade-off between estimation accuracy and computational complexity to further improve the practicality of state estimation systems. By addressing these challenges, we can open up new possibilities for developing and deploying advanced state estimation systems in various applications.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

Xuebo Jin served as an Editor-in-Chief of *ICCK Transactions on Intelligent Systematics* at the time of manuscript submission. To ensure the integrity of the peer-review process, Xuebo Jin was not involved in the editorial handling, peer review, or decision-making process for this manuscript, which was handled independently by another editor. The remaining authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Webb, T. P., Prazenica, R. J., Kurdila, A. J., & Lind, R. (2007). Vision-based state estimation for autonomous micro air vehicles. *Journal of guidance, control, and dynamics*, *30*(3), 816-826. [CrossRef]

[2] Rotella, N., Bloesch, M., Righetti, L., & Schaal, S. (2014, September). State estimation for a humanoid robot. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 952-958). IEEE. [CrossRef]

[3] Li, X., Guvenc, L., & Aksun-Guvenc, B. (2023). Vehicle state estimation and prediction for autonomous driving in a round intersection. *Vehicles, 5*(4), 1328-1352. [CrossRef]

[4] Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82D, 35-45. [CrossRef]

[5] Karimipour, H., & Dinavahi, V. (2015). Extended Kalman filter-based parallel dynamic state estimation. *IEEE transactions on smart grid, 6*(3), 1539-1549. [CrossRef]

[6] Xiong, K., Chan, C. W., & Zhang, H. Y. (2005). Unscented Kalman filter for fault detection. *IFAC Proceedings Volumes, 38*(1), 113-118. [CrossRef]

[7] Arasaratnam, I., & Haykin, S. (2011). Cubature kalman smoothers. *Automatica, 47*(10), 2245-2250. [CrossRef]

[8] Ghahramani, Z., & Hinton, G. E. (1996). *Parameter estimation for linear dynamical systems* (Technical Report No. CRG-TR-96-2). University of Toronto. https://www.cs.toronto.edu/~hinton/absps/tr-96-2.pdf

[9] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780. [CrossRef]

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

[11] Wang, H., & Tu, M. (2020, December). Enhancing attention models via multi-head collaboration. In *2020 International Conference on Asian Language Processing (IALP)* (pp. 19-23). IEEE. [CrossRef]

[12] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*. [CrossRef]

[13] Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015, April). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4580-4584). Ieee. [CrossRef]

[14] Huang, H., Zeng, Z., Yao, D., Pei, X., & Zhang, Y. (2021). Spatial-temporal ConvLSTM for vehicle driving intention prediction. *Tsinghua Science and Technology, 27*(3), 599-609.[CrossRef]

[15] Jin, X. B., Gong, W. T., Kong, J. L., Bai, Y. T., & Su, T. L. (2022). PFVAE: a planar flow-based variational auto-encoder prediction model for time series data. *Mathematics*, 10(4), 610. [CrossRef]



Xuebo Jin received the B.S. and M.S. degrees in control theory and control engineering from Jilin University, Changchun, China, in 1994 and 1997, and the Ph.D. degree in control theory and control engineering from the University of Zhejiang, Zhejiang, China, in 2004. From 2009 to 2012, she was an Assistant Professor with Zhejiang Sci-tech University. Since 2012, she has been a Professor with Beijing Technology and Business University, Beijing, China. Her research includes a variety of areas in information fusion, big data analysis, condition estimation, and video tracking.(Email: jinxuebo@btbu.edu.cn)



Tianxiao Sun received the B.S. Automation from Beijing Technology and Business University, Beijing, China, in 2018. Currently pursuing a Master's degree in Control Engineering at Beijing Technology and Business University, Main research interests focus on navigation and control of autonomous driving, unmanned aerial vehicles, and robots.(Email: suntianxiao@st.btbu.edu.cn)



Wei Chen received the M.S. Control engineering from Beijing Technology and Business University, Beijing, China, in 2024. His research focuses on pattern recognition and information fusion, unmanned vehicles, machine learning, and other related fields.(Email: chenwei@st.btbu.edu.cn)



Huijun Ma received the M.S. Atomic and Molecular Physics from Changchun Institute of Optics and Mechanics , Changchun, China, in 2010. She is currently pursuing a Ph.D. in Systems Science from Beijing University of Technology and Business, with a research focus on complexity System modeling, pattern recognition and information fusion, machine learning, etc.(Email: mahuijun@th.btbu.edu.cn)



Yeqing Wang graduated from Beijing Technology and Business University in 2022 with a Bachelor's degree in Automation. He is currently a graduate student in Control Theory and Control Engineering at the same university. His research interests include pattern recognition and prediction, deep Learning, and related fields. (Email: wangyeqing@st.btbu.edu.cn)