Precision in 3D positional forecasting with machine learning and deep temporal architectures

P. Sirish Kumar[1,2], V. B. S. Srilatha Indira Dutt[3], Sai Kiran Oruganti[1]
[1]Lincoln University College, Petaling Jaya, Malaysia
[2]Department of Electronics and Communication Engineering, Aditya Institute of Technology and Management, Tekkali, India
[3]Department of Electrical, Electronics and Communication, Engineering, GITAM School of Technology
GITAM Deemed to be University, Visakhapatnam, India

Article Info

Article history:

Received Oct 31, 2024
Revised Aug 15, 2025
Accepted Oct 15, 2025

Keywords:

3D positional forecasting
Bidirectional long short-term memory
Prediction uncertainty
Spatiotemporal modeling
Trajectory prediction

ABSTRACT

We present a comparative analysis of traditional machine learning (ML) models, random forest (RF), support vector machine (SVM), and extreme gradient boosting (XGB), and deep learning (DL) architectures, convolutional neural networks (CNN), and bidirectional long short-term memory (BiLSTM) for high-precision 3D positional forecasting. Conventional approaches often underperform when modeling complex spatiotemporal dependencies, limiting their use in dynamic systems such as robotics and autonomous vehicles. This study highlights BiLSTM's advantage in learning bidirectional temporal features, achieving superior R^2 scores and stable prediction intervals compared to both classical ML and spatially-focused CNN models. Uncertainty metrics, prediction interval coverage probability (PICP), and mean prediction interval width (MPIW) provide additional insight into model reliability. Experiments on a 22-hour GPS dataset confirm that BiLSTM achieves both high accuracy and predictive confidence, underscoring its suitability for real-world trajectory forecasting.

Corresponding Author:

P. Sirish Kumar
Lincoln University College
Petaling Jaya, Selangor Darul Ehsan, 47301, Malaysia
Email: pdf.sirish@lincoln.edu.my and sirishdg@gmail.com

1. INTRODUCTION

Accurate 3D positional forecasting is essential for autonomous navigation, robotics, and geospatial systems. These applications require real-time predictions of spatial coordinates governed by both spatial and temporal dependencies. Traditional machine learning (ML) methods often fall short in modeling high-dimensional sequential data, where complex, non-linear relationships evolve over time. Classical models such as random forest (RF), support vector machine (SVM), and extreme gradient boosting (XGB) perform well on static datasets but lack internal mechanisms to capture sequence patterns [1]-[3]. This shortcoming motivates the use of deep learning (DL) models, particularly those suited to time-series data. convolutional neural networks (CNNs), though effective for spatial pattern recognition, do not inherently capture temporal progression. In contrast, LSTM and its bidirectional variant, bidirectional long short-term memory (BiLSTM) offer temporal modeling through memory mechanisms and have shown promising results in prior sequence-learning tasks [4]-[6]. Despite this, few studies directly compare these architectures—especially using uncertainty metrics - on real-world positional datasets. This paper addresses that gap through a structured evaluation of five models on GPS data collected over a continuous 22-hour period. The core objective is to assess whether BiLSTM outperforms classical and CNN models in both accuracy and uncertainty reliability [7], [8]. Our key contributions are:

- A comparative study across five ML and DL models on real GPS-based 3D trajectory data.
- Integration of uncertainty quantification using PICP and MPIW alongside standard metrics.
- Empirical validation that BiLSTM yields the most accurate and reliable forecasts across all dimensions.

The remainder of this paper outlines the methodology in section 2, presents results and interpretation in section 3, and concludes with implications and future directions in section 4.

2. METHOD

2.1. Data source and preprocessing

We use a real-world 22-hour GPS dataset recorded at the Indian Institute of Science (IISc), Bangalore. Each record includes three-dimensional coordinates x, y, z representing longitude, latitude, and altitude [9], [10]. The position (P) at time *i* is represented as in (1).

$$\mathrm{P}_i = [\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i] \tag{1}$$

To ensure uniform scaling and enhance model training, all coordinates are normalized as in (2).

$$\widehat{\mathrm{P}}_i = \frac{\mathrm{P}_i - \mu}{\sigma}\ ,\ \mu = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P_i},\ \sigma = \sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{P_i} - \mu)^2} \tag{2}$$

Where, $\widehat{\mathrm{P}}_i$ is normalized coordinator vector, μ is the mean and σ is the standard deviation of all positions in the dataset. This step reduces data variance and prepares inputs for all learning models.

2.2. Time-series windowing

To frame the data for sequence learning, we apply a sliding window of length T=30 [11], [12]. Each input sequence $\mathrm{X_j}$ consists of 30 timesteps, and the corresponding target $\mathrm{y_j}$ is the next position as in (3).

$$\mathrm{X_j} = \left[\widehat{\mathrm{P}}_j, \widehat{\mathrm{P}}_{j+1}, \cdots\cdots \widehat{\mathrm{P}}_{j+T-1}\right]\ ,\quad \mathrm{y_j} = \widehat{\mathrm{P}}_{j+T} \tag{3}$$

Where j=1,2,…,N−T, and N is the total number of data points after normalization. This transforms the normalized dataset into supervised training samples suitable for temporal forecasting models.

2.3. BiLSTM model structure and training

2.3.1. Model architecture

Each input sequence $\mathrm{X_j}$ (30 timesteps) is processed by a BiLSTM layer with 64 units. Its output is passed through a 50-neuron dense layer (ReLU activation) and a final dense layer to predict 3D coordinates [13]-[15]. The structure of the BiLSTM network, including the bidirectional LSTM layer and dense layers, is illustrated in Figure 1.

2.3.2. LSTM computation

At timestep t, a standard LSTM unit performs as shown in (4) to (6).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)\ ,\ i_t = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c),\ c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\ ,\qquad h_t = o_t \circ tanh(c_t) \tag{6}$$

Here, f_t is Forget Gate, i_t input gate, $\tilde{c}_t$ is candidate memory, c_t is Cell state, h_t is hidden state, o_t is output gate, σ(·) is the sigmoid function, ∘ $tanh(\cdot)$ is the hyperbolic tangent, and ∘ denotes element-wise multiplication, W_* and b_* are the weight matrices and biases associated with each gate

BiLSTM combines forward and backward LSTM outputs as in (7).

$$h_t^{Bi} = [\vec{h}_t, \overleftarrow{h}_t] \tag{7}$$

The prediction is computed as in (8).

$$\hat{y}_j = W_{out} \cdot h_t^{Bi} + b_{out} \quad (8)$$

where, $\hat{y}_j \in \mathbb{R}^3$ is the predicted 3D positional vector.

2.3.4. Training setup

The model is trained using the Mean Squared Error (MSE) loss function as in (9).

$$L_{MSE} = \frac{1}{M}\sum_{j=1}^{M} \|\hat{Y}_j - Y_j\| \quad (9)$$

Where, M is the number of training samples. Training uses early stopping based on validation loss to prevent overfitting.

2.4. CNN model structure and training

To benchmark against a spatial DL approach, we implemented a one-dimensional convolutional neural network (1D-CNN). Though CNNs lack sequence memory, they can effectively learn local spatial patterns within the input window [16]-[18].

2.4.1. Model architecture

The model receives input X_j and passes it through a 1D convolutional layer with 64 filters (kernel size = 3), followed by a flattening layer and a 50-neuron dense layer (ReLU activation). The final dense layer outputs the predicted x, y, z coordinates [19]-[22]. The CNN architecture and data flow are illustrated in Figure 2.

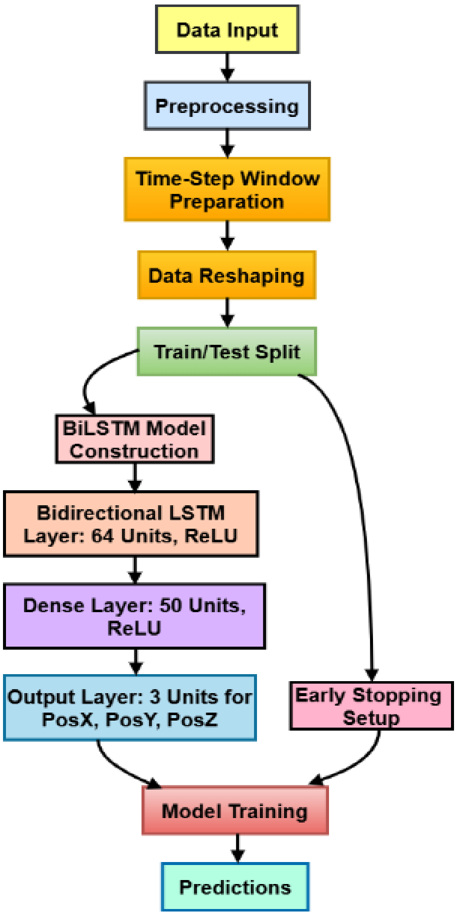


Figure 1. Workflow for BiLSTM model for 3D positional prediction

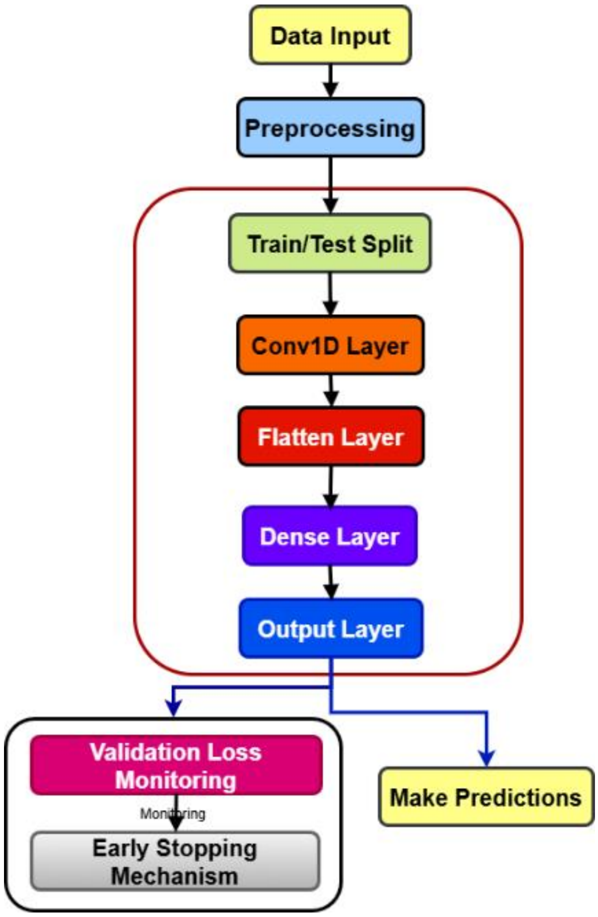


Figure 2. Workflow for CNN model for 3D positional prediction

2.4.2. Convolution operation

The convolution operation at position t for filter m in layer *l* is defined as in (10).

$$h_t^{(l,m)} = f\left(\sum_{k=0}^{K-1} w_k^{(l,m)} \cdot x_{t+k}^{(l-1)} + b^{(l,m)}\right) \tag{10}$$

Where, $f(\cdot)$ is the ReLU activation function, $w_k^{(l,m)}$ are the filter weights, $b^{(l,m)}$ is the bias term, $x_{t+k}^{(l-1)}$ is the input from the previous layer, K is the kernel size (set to 3).

2.4.3. Output layers

The feature map is flattened and transformed is presented in (11).

$$\boldsymbol{z} = Flatten\ \left(\left\{h_t^{(l,m)}\right\}\right) \tag{11}$$

This is then passed through a dense layer as in (12).

$$\mathrm{h}_{\mathrm{dense}} = ReLU(\mathrm{W}_{\mathrm{dense}} \cdot z + \mathrm{b}_{\mathrm{dense}}) \tag{12}$$

And finally mapped to the predicted position as in (13).

$$\hat{\mathrm{y}}_{\mathrm{j}} = \mathrm{h}_{\mathrm{out}} \cdot \mathrm{h}_{\mathrm{dense}} + \mathrm{b}_{\mathrm{out}} \tag{13}$$

2.4.4. Training setup

The CNN model is trained using the same MSE loss function as the BiLSTM. Early stopping is also applied with a patience of 10 epochs, restoring the model weights to the best-performing validation epoch.

2.5. Uncertainty quantification with prediction intervals

In addition to point prediction accuracy, we assess model reliability using prediction intervals (PIs), which estimate the confidence range around each forecast [23]-[25]. Two metrics are used: prediction interval coverage probability (PICP) and mean prediction interval width (MPIW).

2.5.1. Prediction interval estimation

For each prediction $\hat{y}_j$, a confidence interval is computed as in (14).

$$\mathrm{PI}_j = \left[\hat{y}_j - z_\alpha \cdot \sigma_\mathrm{j}, \hat{y}_j + z_\alpha \cdot \sigma_\mathrm{j}\right] \tag{14}$$

Where, σ_j is the estimated standard deviation estimate, z_α is the z-score corresponding to the desired confidence level (e.g., 1.96 for 95% confidence).

2.5.2. Prediction interval coverage probability

PICP quantifies how often true values fall within the predicted intervals as in (15).

$$PICP = \frac{1}{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \mathbb{1}(y_j \in \mathrm{PI}_{j_i}) \tag{15}$$

where $\mathbb{1}(\cdot)$ is the indicator function returning 1 if the true value is within the interval and 0 otherwise, M is the total number of predictions.

2.5.3. Mean prediction interval width

MPIW captures the average width of all prediction intervals as in (16).

$$MPIW = \frac{1}{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \left\| \hat{y}_j^{upper} - \hat{y}_j^{lower} \right\| \tag{16}$$

Together, PICP (reliability) and MPIW (precision) offer a robust uncertainty assessment for each model.

2.6. Model evaluation metrics

We use standard regression and variance metrics to evaluate model performance across all spatial axes in (17) to (21). These include error magnitude, predictive strength, and explained variability. Mean squared error (MSE): Penalizes large deviations, highlighting average squared error.

$$\mathrm{MSE} = \frac{1}{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{M}}\left\|\hat{\boldsymbol{y}}_j - \mathbf{y}_{\mathrm{j}}\right\|^2 \quad (17)$$

Root mean squared error (RMSE): Expresses prediction error in original units.

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{M}}\left\|\hat{\boldsymbol{y}}_j - \mathbf{y}_{\mathrm{j}}\right\|^2} \quad (18)$$

Mean absolute error (MAE): Measures average absolute difference.

$$\mathrm{MAE} = \frac{1}{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{M}}\left\|\hat{\mathbf{y}}_{\mathrm{j}} - \mathbf{y}_{\mathrm{j}}\right\| \quad (19)$$

R-squared (R^2): Reflects the proportion of variance explained by the model.

$$\mathrm{R}^2 = 1 - \frac{\sum_{\mathrm{j}=1}^{\mathrm{M}}\left\|\hat{y}_j - \mathrm{y}_{\mathrm{j}}\right\|^2}{\sum_{\mathrm{j}=1}^{\mathrm{M}}\left\|\hat{y}_j - \bar{\mathrm{y}}_{\mathrm{j}}\right\|^2} \quad (20)$$

Where $\bar{y}$ is the mean of all true target vectors.
Explained variance score (EVS): Indicates how well the model captures variation in target data.

$$\mathrm{EVS} = 1 - \frac{\mathrm{Var}(\mathbf{y}_{\mathbf{j}} - \hat{\mathbf{y}}_{\mathrm{j}})}{\mathrm{Var}(\mathbf{y}_{\mathbf{j}})} \quad (21)$$

3. RESULTS AND DISCUSSION

This section presents a comprehensive evaluation of all five models, three classical ML models (RF, SVM, and XGB) and two DL architectures (CNN and BiLSTM), on the task of 3D positional forecasting. The models were assessed based on multiple performance indicators, including prediction accuracy metrics (MSE, RMSE, MAE, R^2, EVS) and uncertainty quantification metrics (PICP and MPIW), evaluated separately across the X, Y, and Z coordinate axes.

The results are presented in tabular form, followed by an in-depth discussion that interprets the key findings. We examine the strengths and limitations of each model, compare outcomes with existing literature, and discuss implications for real-world deployment. This structured analysis emphasizes the practical significance of the findings and demonstrates how the BiLSTM model effectively overcomes limitations inherent in traditional ML and spatially-focused DL approaches.

3.1. Performance of classical ML models

The predictive performance of RF, SVM, and XGB models across all three spatial dimensions is summarized in Table 1. These models were evaluated using the standardized input-output pairs generated from time-series windowing. While all three models demonstrated reasonably good performance, their ability to capture the underlying temporal structure of the data varied considerably.

Among the classical models, XGB consistently outperformed both RF and SVM, particularly in the Y and Z axes, achieving higher R^2 scores (up to 0.9958) and lower RMSE values. XGB's tree-boosting mechanism enabled it to capture complex non-linear patterns in the trajectory data, making it a strong baseline for non-sequential modeling. Additionally, it demonstrated narrower prediction intervals (MPIW ≈ 0.198–0.199) and respectable coverage (PICP ≈ 0.989), suggesting a balance between accuracy and uncertainty estimation.

RF, while less powerful than XGB, delivered robust performance, especially in the Z-axis with minimal variance in prediction error. RF's ensemble of decorrelated trees helped reduce overfitting and contributed to stable MAE values. However, its performance across all dimensions showed a slight lag in capturing subtle dynamic variations in the dataset.

SVM, in contrast, produced the weakest results. While it maintained moderate accuracy ($R^2 \approx 0.991$), its higher MAE and wider prediction intervals indicate difficulty in generalizing over the full sequence window. The lack of internal mechanisms to model temporal correlation limits its utility for sequence-based forecasting tasks.

These observations reinforce the notion that while classical ML models can offer baseline predictive capabilities, they struggle to generalize temporal dependencies, which are critical in trajectory prediction scenarios. This highlights the need for more specialized architectures capable of modeling time-aware patterns, an area addressed by the DL models in the next section.

Table 1. Comparative performance of ML and DL models across all metrics

Model	Coordinate	Evaluation metric						
		R^2	MSE	RMSE	MAE	Explained variance	PICP	MPIW
RF	X	0.9940	0.0059	0.0771	0.0070	0.9940	0.9914	0.1984
	Y	0.9935	0.0065	0.0807	0.0068	0.9935	0.9904	0.1991
	Z	0.9947	0.0052	0.0723	0.0070	0.9947	0.9910	0.1988
SVM	X	0.9909	0.0090	0.0949	0.0506	0.9909	0.9510	0.9909
	Y	0.9915	0.0085	0.092	0.0519	0.9915	0.9405	0.9915
	Z	0.9919	0.0081	0.0897	0.0562	0.9920	0.9516	0.9919
XGB	X	0.9936	0.0064	0.0797	0.0110	0.9936	0.9892	0.1984
	Y	0.9951	0.0049	0.0699	0.0093	0.9951	0.9894	0.1995
	Z	0.9958	0.0042	0.0647	0.0106	0.9958	0.9889	0.1989
CNN	X	0.9945	0.0055	0.0741	0.0183	0.9945	0.9933	0.1990
	Y	0.9951	0.0049	0.0699	0.0160	0.9951	0.9939	0.1987
	Z	0.9959	0.0040	0.0636	0.0164	0.9959	0.9910	0.1992
BILSTM	X	0.9948	0.0052	0.0719	0.011	0.9948	0.9965	0.1988
	Y	0.9954	0.0046	0.0678	0.0104	0.9954	0.9975	0.1999
	Z	0.9962	0.0038	0.0613	0.0098	0.9962	0.9964	0.1999

3.2. DL model performance (CNN and BiLSTM)

In contrast to the classical models, both DL architectures, CNN and BiLSTM demonstrated significantly enhanced performance in modeling the complex spatiotemporal patterns of the GPS dataset. Their ability to extract meaningful representations from sequential input data led to improvements across nearly all evaluation metrics.

The CNN achieved strong results across the X, Y, and Z dimensions, with R^2 values exceeding 0.995 in all cases and RMSE values consistently below 0.075. Its strength lies in its capability to extract localized spatial features across the 30-timestep input window. However, because CNNs lack inherent temporal recurrence, they are limited in modeling long-term dependencies. This is visible in the positional plot shown in Figure 3, where predicted values align closely with actual values but show slightly more dispersion than BiLSTM, particularly in the Y and Z axes. Residual plot in Figure 4 further illustrates these deviations, with residuals showing broader variance compared to BiLSTM.

The BiLSTM model, however, delivered the most accurate and stable performance across all models. It achieved the highest R^2 scores (0.9948 to 0.9962) and the lowest RMSE and MAE values, particularly along the Z-axis, which often presents greater variability in GPS data. As shown in Figure 5, the predicted positional values closely follow the actual values, with minimal dispersion. The residual plot in Figure 6 further confirms this: residuals are tightly clustered around zero, highlighting the model's ability to minimize prediction error across the entire sequence.

Moreover, BiLSTM produced the highest PICP (~0.9975 for Y-axis) and maintained MPIW within 0.199, indicating not only accuracy but also high confidence in its predictions. This performance affirms BiLSTM's suitability for time-series forecasting in scenarios where sequence dependencies are critical such as vehicle tracking, UAV navigation, and robotics.

The learning curves of both models are presented in Figures 7 and 8. As seen in Figure 8, the BiLSTM model converged smoothly and achieved minimal validation loss by epoch 37, while Figure 7 shows CNN plateauing slightly earlier. This difference in convergence behavior reflects BiLSTM's greater representational capacity for capturing temporal dynamics.

In summary, while CNN provides a fast and spatially-aware alternative, BiLSTM emerges as the most effective model for accurate and reliable 3D positional forecasting. Its superior performance directly addresses the limitations of classical and convolutional approaches by fully leveraging temporal patterns in the input data.

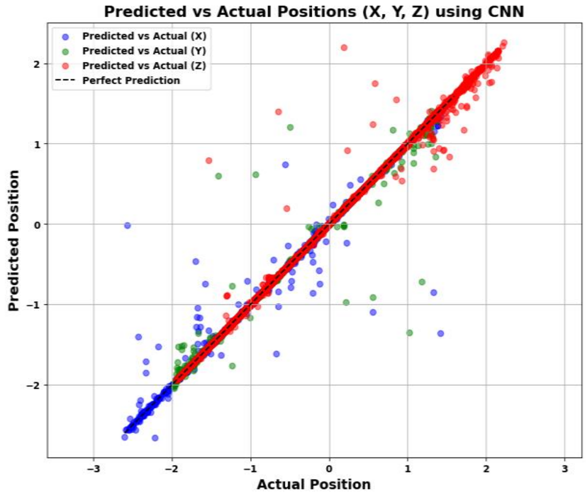


Figure 3. CNN comparative positional plot

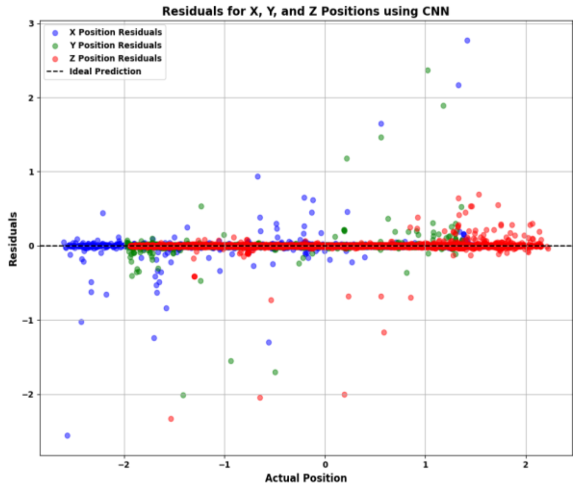


Figure 4. CNN residual error plot

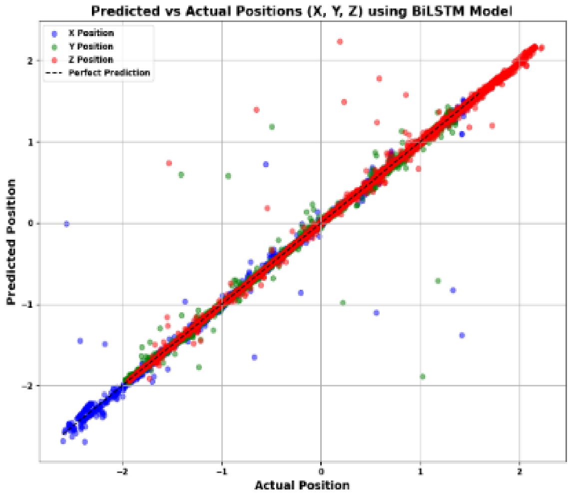


Figure 5. BiLSTM comparative positional plot

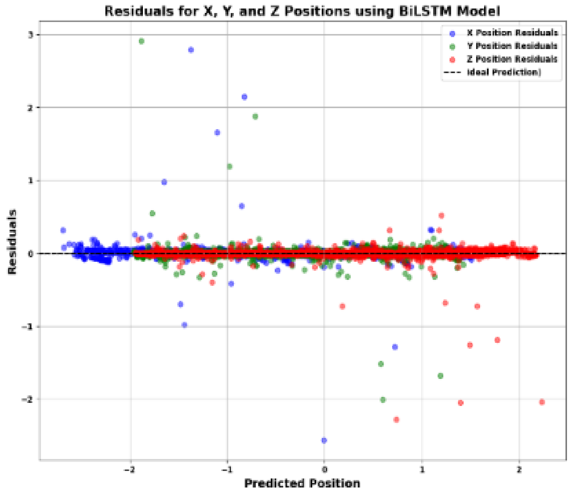


Figure 6. BiLSTM residual error plot

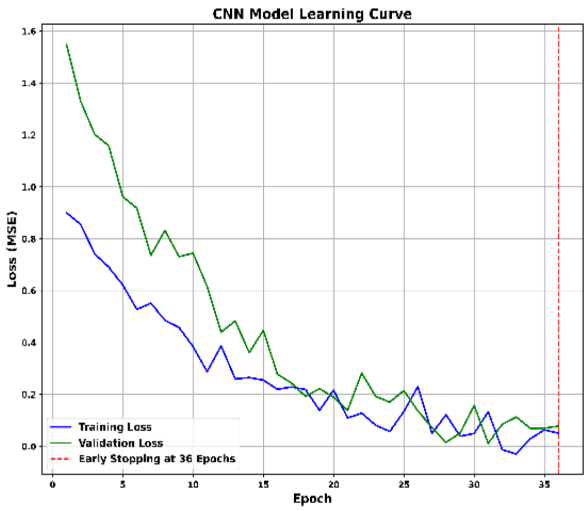


Figure 7. CNN learning curve

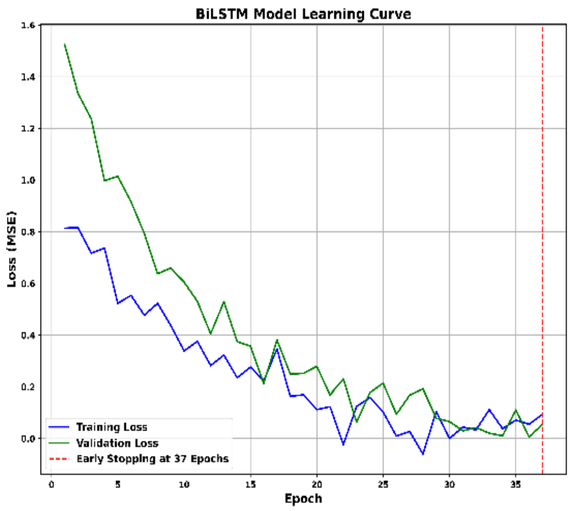


Figure 8. BiLSTM learning curve

4. CONCLUSION

In this study, we conducted a comparative analysis of traditional ML models (RF, SVM, XGB) and DL architectures (CNN, BiLSTM) for 3D positional forecasting using a real-world GPS dataset. Through a structured evaluation based on accuracy and uncertainty metrics, we demonstrated that BiLSTM offers the most robust and reliable performance. Its bidirectional temporal learning capability allows it to capture complex sequential dependencies, outperforming classical and spatial-only models in both predictive accuracy and confidence interval estimation. The results affirm that models capable of learning long-range temporal dependencies are better suited for trajectory forecasting tasks. While CNNs provide an efficient spatial alternative, and classical models offer solid baselines, BiLSTM's consistent accuracy, high PICP, and narrow MPIW make it the most viable candidate for real-world forecasting applications. This positions BiLSTM as a practical solution for deployment in fields such as autonomous navigation, logistics, and mobile robotics, where precise and dependable position prediction is mission-critical. Future work may explore hybrid architectures combining CNNs and BiLSTM layers to leverage both spatial and temporal features, as well as the integration of external variables such as speed, direction, or sensor noise. Additionally, testing across varied datasets and environmental conditions would further validate the generalizability and resilience of the proposed approach.

FUNDING INFORMATION

Authors state no funding involved.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] X. Guo, M. Adl, B. Abdi, and A. Emadi, "Intersection-specific trajectory prediction for road users: a review," *IEEE Access*, vol. 13, pp. 40054–40075, 2025, doi: 10.1109/ACCESS.2025.3546325.

[2] P. Sirish Kumar and V. B. S. Srilatha Indira Dutt, "The global positioning system: Popular accuracy measures," *Materials Today: Proceedings*, vol. 33, pp. 4797–4801, 2020, doi: 10.1016/j.matpr.2020.08.380.

[3] R. Walters, J. Li, and R. Yu, "Trajectory prediction using equivariant continuous convolution," *arXiv preprint arXiv:2010.11344*, 2020.

[4] Z. Liu, J. Liu, X. Xu, and K. Wu, "DeepGPS: deep learning enhanced GPS positioning in urban canyons," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 376–392, Jan. 2024, doi: 10.1109/TMC.2022.3208240.

[5] S. Raina, J. Challagundla, and M. Singh, "Deep learning-based road user prediction in smart cities," *IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 00753–00759, 2025.

[6] Y. Shi, "Reliable uncertainty quantification in machine learning via conformal prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 28, pp. 29299–29300, Apr. 2025, doi: 10.1609/aaai.v39i28.35227.

[7] Y. Du and N. Ding, "A systematic review of multi-scale spatio-temporal crime prediction methods," *ISPRS International Journal of Geo-Information*, vol. 12, no. 6, p. 209, May 2023, doi: 10.3390/ijgi12060209.

[8] M. A. Belay, A. Rasheed, and P. S. Rossi, "Sparse non-linear vector autoregressive networks for multivariate time series anomaly detection," *IEEE Signal Processing Letters*, vol. 32, pp. 331–335, 2025, doi: 10.1109/LSP.2024.3520019.

[9] S. K. Pagoti and S. I. D. Vemuri, "Development and performance evaluation of correntropy Kalman filter for improved accuracy of GPS position estimation," *International Journal of Intelligent Networks*, vol. 3, pp. 1–8, 2022, doi: 10.1016/j.ijin.2022.01.002.

[10] M. E. Leverington, "Parametric study of thermal and chemical nonequilibrium nozzle flow," *Department of Electrical and Electronic Engineering Osaka University*, 2022.

[11] N. A. M. Razali, N. Shamsaimon, K. K. Ishak, S. Ramli, M. F. M. Amran, and S. Sukardi, "Gap, techniques and evaluation: traffic flow prediction using machine learning and deep learning," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00542-7.

[12] H. Abbasimehr and R. Paki, "Improving time series forecasting using LSTM and attention models," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 673–691, Jan. 2022, doi: 10.1007/s12652-020-02761-x.

[13] M. Fathi, R. Shah-Hosseini, A. Moghimi, and H. Arefi, "MHRA-MS-3D-ResNet-BiLSTM: A multi-head-residual attention-based multi-stream deep learning model for soybean yield prediction in the U.S. using multi-source remote sensing data," *Remote Sensing*, vol. 17, no. 1, p. 107, Dec. 2025, doi: 10.3390/rs17010107.

[14] X. Wang *et al.*, "Air quality forecasting using a spatiotemporal hybrid deep learning model based on VMD–GAT–BiLSTM," *Scientific Reports*, vol. 14, no. 1, Aug. 2024, doi: 10.1038/s41598-024-68874-x.

[15] S. Langman, N. Capicotto, Y. Maddahi, and K. Zareinia, "Roboethics principles and policies in Europe and North America," *SN Applied Sciences*, vol. 3, no. 12, Nov. 2021, doi: 10.1007/s42452-021-04853-5.

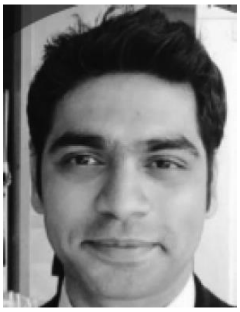
[16] A. B. S. Nzao, "Study and modeling of human biological tissue exposed to high frequency electromagnetic waves," *Open Journal of Applied Sciences*, vol. 11, no. 10, pp. 1109–1121, 2021, doi: 10.4236/ojapps.2021.1110083.

[17] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12357 LNCS, Springer International Publishing, 2020, pp. 507–523.
[18] C. Feng *et al.*, "MacFormer: map-agent coupled transformer for real-time and robust trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6795–6802, Oct. 2023, doi: 10.1109/LRA.2023.3311351.
[19] D. Singh and R. Srivastava, "Graph neural network with RNNs based trajectory prediction of dynamic agents for autonomous vehicle," *Applied Intelligence*, vol. 52, no. 11, pp. 12801–12816, Feb. 2022, doi: 10.1007/s10489-021-03120-9.
[20] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks," *Information Sciences*, vol. 577, pp. 852–870, Oct. 2021, doi: 10.1016/j.ins.2021.08.042.
[21] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 657–666, doi: 10.1109/CVPR42600.2020.00074.
[22] I. Bae, J. H. Park, and H. G. Jeon, "Learning pedestrian group representations for multi-modal trajectory prediction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13682 LNCS, Springer Nature Switzerland, 2022, pp. 270–289.
[23] H. Zhang, G. Yang, H. Yu, and Z. Zheng, "Kalman filter-based CNN-BiLSTM-ATT model for traffic flow prediction," *Computers, Materials and Continua*, vol. 76, no. 1, pp. 1047–1063, 2023, doi: 10.32604/cmc.2023.039274.
[24] P. S. Kumar, V. B. S. S. I. Dutt, V. A. Kumar, A. Jayalaxmi, P. K. Rao, and K. P. Rao, "Optimizing GPS positioning: a deep learning approach to improve accuracy," in *2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies, INSPECT 2024*, Dec. 2024, pp. 1–6, doi: 10.1109/INSPECT63485.2024.10896189.
[25] O. Y. Al-Jarrah, A. S. Shatnawi, M. M. Shurman, O. A. Ramadan, and S. Muhaidat, "Exploring deep learning-based visual localization techniques for UAVs in GPS-denied environments," *IEEE Access*, vol. 12, pp. 113049–113071, 2024, doi: 10.1109/ACCESS.2024.3440064.

BIOGRAPHIES OF AUTHORS

Dr. P. Sirish Kumar is an Associate Professor in the Department of Electronics and Communication Engineering at Aditya Institute of Technology and Management, with over 17 years of experience in the field. He is also currently a Postdoctoral Researcher at Lincoln University College, Malaysia. His areas of expertise include Global Navigation Satellite Systems (GNSS), Machine Learning, and Embedded Systems. A Ph.D. Gold Medalist from GITAM University, he has published over 60 journal papers and presented in more than 50 conferences. He holds 10 patents, authored 6 textbooks, and has successfully guided 28 student projects. He has completed 56 global certifications and actively contributes as a reviewer, session chair, technical committee member, and review board member for various reputed journals and conferences. He can be contacted at: pdf.sirish@lincoln.edu.my/ sirishdg@gmail.com.

Dr. V. B. S. Srilatha Indira Dutt is a Professor in the Department of Electrical, Electronics and Communication Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam. She has over 22 years of experience in Satellite Communications, GNSS, Wireless Communications, and Machine Learning algorithms. Her core research interests focus on applying ML techniques to GNSS and wireless systems. She has authored more than 75 research publications in reputed national and international journals and conferences. Under her guidance, seven Ph.D. scholars have been awarded their doctorates, and seven more are currently pursuing research. She can be contacted at email: svemuri@gitam.edu.



Prof. Sai Kiran Oruganti is currently working as an Associate Professor in the Faculty of Engineering and Built Environment, Lincoln University College, Kuala Lumpur, Malaysia. He is also affiliated with Technology Innovation Hub (TIH), Indian Institute of Technology Patna. From 2019 to 2022, he was associated with the School of Electrical and Automation Engineering, Jiangxi University of Science and Technology, China, where he contributed to the establishment of an advanced wireless power transfer research laboratory. His expertise lies in wireless power transmission through radio-shielded zones. He earned his Ph.D. in 2016 from the Ulsan National Institute of Science and Technology (UNIST), South Korea, where he developed wireless communication systems for Hyundai Heavy Industries and Samsung Heavy Industries. His doctoral work led to the creation of the spin-off company ZN-Ocean Technologies. He has previously served as a faculty member at IIT Tirupati and is a recipient of the URSI Young Scientist Award (2016). He can be contacted at email: saisharma@lincoln.edu.my.