International Journal of Operations Research and Artificial Intelligence

www.ijorai.reapress.com

Int. J. Oper. Res. Artif. Intell. Vol. 2, No. 1 (2026) 11–22.

Paper Type: Original Article

Comparison of SARIMA-LSTM and CNN-LSTM Machine Learning Algorithms in Predicting Time Series Analysis: Road Traffic Accidents Data

Manoochehr Babanezhad[1,*], Hassan Khorsha[2]

[1] Department of Statistics, Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran; m.babanezhad@gu.ac.ir.
[2] Health Management and Social Development Research Center, Golestan University of Medical Sciences, Gorgan, Iran; Khorsha.h@gmail.com.

Citation:

Received: 12 July 2025 *Revised: 22 October 2025* *Accepted: 02 Januray 2026*	Babanezhad, M., & Khorsha, H. (2026). Comparison of SARIMA-LSTM and CNN-LSTM machine learning algorithms in predicting time series analysis: Road traffic accidents data. *International Journal of Operations Research and Artificial Intelligence*, 2(1), 11-22.

Abstract

Traditional time series models may not adequately capture the underlying patterns for prediction in particular time series data. Hybrid machine learning approaches offer a more effective solution. This study compares two hybrid machine learning approaches, Seasonal Autoregressive Integrated Moving Average (SARIMA)-Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN)-LSTM, for time series forecasting of historical road traffic accident counts. The SARIMA-LSTM model combines the ability of SARIMA to capture trend and seasonality with the nonlinear learning capacity of LSTM networks, while the CNN-LSTM model integrates convolutional feature extraction with temporal sequence learning. Time series road traffic accident counts spanning five consecutive years (1399–1403 Iranian calendar) were used to evaluate both models. The findings demonstrate that the CNN-LSTM model consistently outperforms the SARIMA-LSTM approach across all evaluation metrics, achieving significantly lower error rates and providing more reliable forecasts for traffic accident prediction.

Keywords: Time series models, Hybrid machine learning, SARIMA-LSTM, CNN-LSTM.

1 | Introduction

Traditional time series models have long played an important role in forecasting because they provide a structured framework for representing temporal dependence, trend, and seasonality. Among these methods, the Seasonal Autoregressive Integrated Moving Average (SARIMA) model has been widely used in many

Corresponding Author: m.babanezhad@gu.ac.ir
https://doi.org/10.48314/ijorai.v2i1.84

forecasting applications due to its interpretability and its ability to capture linear patterns in sequential data [1]. SARIMA is particularly effective for time series data exhibiting stable autocorrelation and strong seasonal patterns. However, in many real-world applications, especially in public health and transportation, time series data are often influenced by nonlinear fluctuations, abrupt changes, and irregular shocks that cannot be fully represented by a purely statistical linear model [2]. As a result, the forecasting performance of SARIMA may decline when the underlying data-generating process is more complex than its assumptions allow. This limitation becomes especially important when forecasting road traffic accident data, which are often affected by multiple dynamic and unpredictable factors [3].

Road traffic accident data provide a clear example of such complexity. Accident counts may vary according to daily and weekly traffic volume, holidays, weather conditions, infrastructure changes, special events, and driver behavior patterns [4], [5]. These factors can produce strong seasonal structures as well as sudden irregular deviations. For this reason, the prediction of road traffic accident counts requires methods that can capture both regular temporal patterns and nonlinear disturbances. In this context, road traffic accident forecasting requires methods that can capture not only regular temporal patterns but also nonlinear disturbances and abrupt surges. Accurate forecasting of accident series is important not only for transportation planning, but also for public health preparedness, emergency response, and resource allocation in hospitals and trauma centers [6], [7]. Therefore, suitable and flexible models are needed to improve predictive performance in this setting. Alternative approaches such as SARIMA and Facebook Prophet have also been explored for time series forecasting [8].

In recent years, deep learning methods have become increasingly popular for time series forecasting because of their ability to learn complex nonlinear relationships directly from data [2], [9]. Long Short-Term Memory (LSTM) networks are among the most widely used recurrent neural network architectures for sequential prediction [10], [11]. LSTM models are designed to retain useful historical information over long sequences while selectively forgetting irrelevant patterns through their gating mechanisms [11]. This capability makes them especially suitable for capturing long-range temporal dependencies and nonlinear structures in time series data. Nevertheless, LSTM models alone may not always perform optimally when the series contains strong seasonal patterns or when the amount of training data is limited. In such cases, the network may need to learn both regular structure and irregular variation at the same time, which can make training more difficult and predictions less stable [9].

To overcome these limitations, hybrid forecasting models have attracted increasing attention [11-13]. Hybrid models aim to combine the strengths of different approaches so that one method compensates for the weaknesses of another [2], [9]. In this study, two hybrid models are considered: SARIMA-LSTM and Convolutional Neural Network (CNN)-LSTM. The SARIMA-LSTM model follows a decomposition-based strategy in which SARIMA is first used to model the main linear and seasonal components of the series, and LSTM is then trained on the residuals to learn the nonlinear patterns that remain unexplained [10], [13]. This approach is useful because it separates the predictable statistical structure from the more complex hidden variation. As a result, the LSTM is not required to learn all features of the time series simultaneously, which can improve both forecasting accuracy and model stability.

The CNN-LSTM model represents a different hybrid strategy [14]. CNNs are effective at extracting local patterns from data by applying convolutional filters across short temporal windows [2]. In time series forecasting, CNN layers can detect short-term spikes, abrupt changes, and local repeated patterns that may be important for prediction [14]. The extracted features are then passed to LSTM layers, which model the longer-term sequential dependence among those features. In this architecture, CNN acts as a feature extractor, while LSTM serves as the temporal sequence learner. This architecture makes CNN-LSTM especially useful for forecasting problems where both local fluctuations and long-range temporal structure are present [11], [13].

Although SARIMA-LSTM and CNN-LSTM are both promising forecasting frameworks, they operate through different mechanisms and may therefore perform differently depending on the structure of the data.

SARIMA-LSTM is particularly well suited to series with strong seasonal regularity and nonlinear residual behavior, while CNN-LSTM may be more effective when short-term local patterns are important and need to be learned before long-range dependencies are modeled [10], [11], [13]. In the context of road traffic accident forecasting, both approaches are relevant because accident counts are shaped by recurring cycles as well as sudden, unpredictable changes [4], [5], [15]. A systematic comparison of these two hybrid models is therefore needed to determine which strategy offers better predictive performance in this application.

The aim of this study is to compare the forecasting ability of SARIMA-LSTM and CNN-LSTM using historical road traffic accident counts from Golestan Province, Iran. Specifically, the study evaluates which model better captures the temporal behavior of accident data and provides more accurate predictions across the forecasting horizon. By comparing these two hybrid approaches, the study seeks to provide practical insight into the most suitable forecasting framework for road traffic accident prediction. The findings are expected to be useful for transportation safety analysis, emergency planning, and the allocation of healthcare resources in anticipation of accident-related patient demand [6], [7].

2 | Hybrid Machine Learning Models

This section defines two distinct hybrid forecasting architectures. Both integrate deep learning with complementary preprocessing mechanisms; however, they fundamentally differ in how they decompose and learn from the temporal signal.

2.1 | SARIMA-LSTM Hybrid Model

This model operates sequentially. A classical statistical model first decomposes the time series, and a neural network then models the residuals as follows [7], [9].

Step 1 (SARIMA baseline modeling). The SARIMA model is expressed using the backshift operator B (where $B^k Y_t = Y_{t-k}$) and seasonal period s. The general form is given by:

$$\phi_P(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D Y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t,$$

where ϕ_p and θ_q are the non-seasonal autoregressive and moving average polynomials of orders p and q, Φ_P and Θ_Q are the seasonal polynomials of orders P and Q, d and D are the degrees of non-seasonal and seasonal differencing, and ϵ_t is the white noise error term at time t.

Step 2 (residual extraction). Since SARIMA cannot capture complex nonlinear interactions, the residual sequence U_t is computed as the difference between the actual observed value and the SARIMA prediction at time t:

$$U_t = Y_t - \hat{Y}_t^{SARIMA}.$$

Step 3 (LSTM correction of residuals). Instead of modeling the raw accident counts directly, the LSTM network is trained to map historical residual lags to the current residual. The input feature vector for the LSTM at time t is constructed using a sliding window of length L:

$$X_t^{LSTM} = [U_{t-L}, U_{t-L+1}, \dots, U_{t-1}].$$

Within the LSTM architecture, each time step τ processes the input X_τ alongside the previous hidden state $h_{\tau-1}$ and cell state $C_{\tau-1}$. The internal gating mechanisms are formally defined as:

Forget gate: Determines which historical information to discard.

$$f_\tau = \sigma(W_f . [h_{\tau-1}, X_\tau] + b_f).$$

Input gate: Decides which new information to store in the cell state.

$$i_\tau = \sigma(W_i.[h_{\tau-1}, X_\tau] + b_i).$$

Candidate cell state: Generates new candidate values.

$$\breve{C}_\tau = \tanh(WC.[h_{\tau-1}, x_\tau] + bc).$$

Cell state update: Combines the forget gate and input gate outputs to update the long-term memory.

$$C_\tau = f_\tau \odot C_{\tau-1} + i_\tau \odot \breve{C}_\tau.$$

Output gate: Controls the flow of information to the hidden state.

$$o_\tau = \sigma(W_\sigma.[h_{\tau-1}, x_\tau] + b_o).$$

Hidden state update: Produces the final output for the current time step.

$$h_\tau = o_\tau \odot \tanh(C_\tau).$$

Here, σ denotes the sigmoid activation function, $\odot$ represents element-wise multiplication, and W and b are the respective weight matrices and biases for each gate.

Step 4 (hybrid final prediction). After the LSTM network outputs the residual forecast $\widehat{U}_t$, the final hybrid prediction is obtained by combining the residual with the statistical baseline from SARIMA:

$$\widehat{Y}_t^{Hybrid} = \widehat{Y}_t^{SARIMA} + \widehat{U}_t,$$

where $\widehat{Y}_t^{Hybrid}$ is the final predicted value from the SARIMA-LSTM hybrid model at time t, and $\widehat{Y}_t^{SARIMA}$ is the forecast from the SARIMA component (capturing linear trend and seasonality), and $\widehat{U}_t$ is the residual forecast from the LSTM component (capturing nonlinear deviations). By explicitly isolating the seasonal, trend, and cyclic components through SARIMA, this hybrid architecture relieves the LSTM network from the burden of learning explicit periodicities. Instead, the LSTM is specialized to model only the unpredictable, nonlinear deviations that SARIMA cannot capture, such as sudden spikes in road traffic accidents caused by adverse weather conditions, holiday periods, traffic congestion, or infrastructure disruptions. This decomposition strategy ensures that each component focuses on its area of strength: SARIMA handles the structured, predictable linear patterns, while LSTM captures the complex, unstructured nonlinear residuals. As a result, the SARIMA-LSTM model forms a robust temporal decomposition framework that improves overall forecasting accuracy and stability.

2.1 | CNN-LSTM Hybrid Model

This model operates in an integrated parallel-series architecture. The CNN first extracts spatial features from multi-dimensional input grids, which are subsequently sequenced temporally and passed to the LSTM [2,10]. Unlike SARIMA-LSTM, which operates exclusively on univariate temporal residuals, this architecture is specifically designed to ingest spatial contextual data, such as geographic risk maps, and learn hierarchical representations of accident-prone areas over time.

Step 1 (spatial feature extraction via CNN). At each time step t, the input to the CNN is a multi-channel 2D spatial grid representing the current state of the environment:

$$X_t^{cnn} \in R^{H\times W\times C},$$

where H and W denote the height and width of the spatial grid (e.g., a city map), and C represents the number of input channels, which may include traffic flow intensity, precipitation levels, road density, and historical accident risk indices. The CNN applies a set of convolutional kernels to extract localized spatial features. For a kernel K_k of size M×N, the resulting feature map F_k at spatial location (i,j) is computed as:

$$F_K(i,j) = \phi(\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{n=1}^{N} K_k(m,n,c).X_t^{cnn}(i+m,j+n,c) + b_k),$$

where $K_k(m,n,c)$ are the learnable weights of the k-th filter applied to channel c; b_k is the corresponding bias term; $\phi(\cdot)$ denotes a non-linear activation function, which introduces sparsity and accelerates convergence. After the convolutional layers produce a set of feature maps F_t, these spatial grids must be vectorized to serve as input for the subsequent LSTM. This transformation is achieved via one of two strategies:

$$s_t = \text{Flatten}(F_t) \text{ or } s_t = \text{GlobalMaxpool}(F_t),$$

where Flattening preserves all spatial details but may result in a very high-dimensional vector, and Global Max Pooling condenses each feature map into a single dominant value, drastically reducing dimensionality while retaining the strongest activated spatial signals (e.g., the most critical high-risk zones within the grid).

Step 2 (temporal sequence construction). To model how these spatial features evolve over time, a historical sequence of these flattened spatial feature vectors is constructed using a sliding window of length L:

$$[S_{t-L}, S_{t-L+1}, \dots, S_{t-1}].$$

This sequence is then fed into the LSTM layers.

Step 3 (sequential dependency modeling via LSTM). At each time step τ, the LSTM cell updates its hidden state based on the incoming spatial feature vector and the previous hidden state, parameterized by its internal weights θ:

$$h_\tau = \text{LSTM} - \text{cell}(h_{\tau-1}, s_\tau; \theta),$$

$$\widehat{Y}_t^{\text{Hybrid}} - W_y.h_{t-1} + by.$$

This sequential modeling process allows the network to capture the temporal progression of the spatial risk landscape, for instance, how a localized rainstorm moves across the city grid over several hours, altering accident risks accordingly.

Step 4 (hybrid prediction output). Finally, the last hidden state h_{t-1} (or the concatenated hidden states) is passed through a fully connected layer to produce the final predicted accident count for the target area at time t:

$$\widehat{Y}_t^{\text{Hybrid}} - W_y.h_{t-1} + b_y.$$

Here, W_y and b_y are the weights and bias of the output regression layer, mapping the deep Spatio-temporal representations back to the scalar accident count.

Note that this architecture treats forecasting not merely as a time-series extrapolation but as a dynamic spatial pattern recognition problem. The CNN acts as an automated feature engineer, identifying which geographic configurations and environmental conditions are most predictive of accidents, while the LSTM models the propagation and evolution of these conditions over time, making it exceptionally powerful for multivariate, geospatially-aware prediction.

1.3 | Comparative Distinction

These two algorithms provide complementary prognostic capabilities. SARIMA-LSTM offers a statistically robust baseline that excels with limited data and strong seasonality, while CNN-LSTM, though computationally more expensive, offers superior adaptability to evolving environmental contexts [2], [7]. Combined, they enable a comprehensive emergency preparedness framework for optimal resource allocation. The architectures are illustrated in *Fig. 1*: SARIMA-LSTM (left) operates sequentially with SARIMA capturing linear patterns and LSTM correcting nonlinear residuals, while CNN-LSTM (right) integrates CNN feature extraction with LSTM temporal sequence learning.

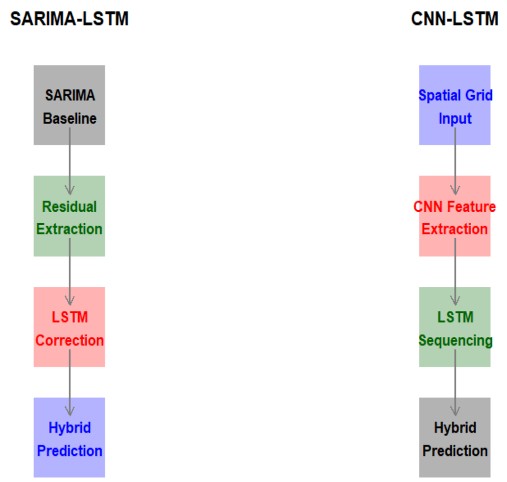

Fig. 1. Comparative architecture of the SARIMA-LSTM (left) and CNN-LSTM (right) hybrid models. Arrows indicate information flow.

3 | Data Description

This study is based on traffic accident data recorded in hospitals affiliated with the University of Medical Sciences in the geographical region of Golestan Province. These road traffic accident statistics span five consecutive years from 1399 to 1403 in the Iranian calendar. The data were systematically collected from the electronic hospital information system, which provided comprehensive information on patient characteristics, accident-related variables, and hospital outcomes. The study cohort included all patients involved in road traffic accidents who were admitted to or referred to hospitals affiliated with Golestan University of Medical Sciences during the study period. The dataset comprises 60 monthly observations (12 months × 5 years).

3.1 | Data Preprocessing

I. Prior to model training, the data underwent the following preprocessing steps:

II. Time series conversion: The monthly data were converted to a time series object with frequency 12 to capture annual seasonality.

III. Stationarity assessment: The Augmented Dickey-Fuller (ADF) test was performed to assess stationarity. The series was differenced if necessary to achieve stationarity for SARIMA modeling.

IV. Normalization: For the deep learning components (LSTM and CNN), the data were normalized using Min-Max scaling to the range [0, 1] to facilitate gradient-based optimization.

V. Train-test split: The dataset was divided into training and testing sets for model evaluation. The training set (1399–1402) was used for model calibration and parameter estimation, while the testing set (1403–1404) was used for out-of-sample evaluation of predictive performance. For the 24-month forecast (1405-1405), all 60 observations were used for model training.

3.2 | Residual Decomposition Analysis

Residual decomposition analysis is performed to quantify the portion of the time series that cannot be explained by the SARIMA model's linear assumptions. By extracting and analyzing the residual sequence, we identify the nonlinear patterns, seasonal deviations, and irregular shocks that require LSTM correction, thereby validating the hybrid approach and demonstrating the necessity of incorporating a neural network component for improved forecasting accuracy.

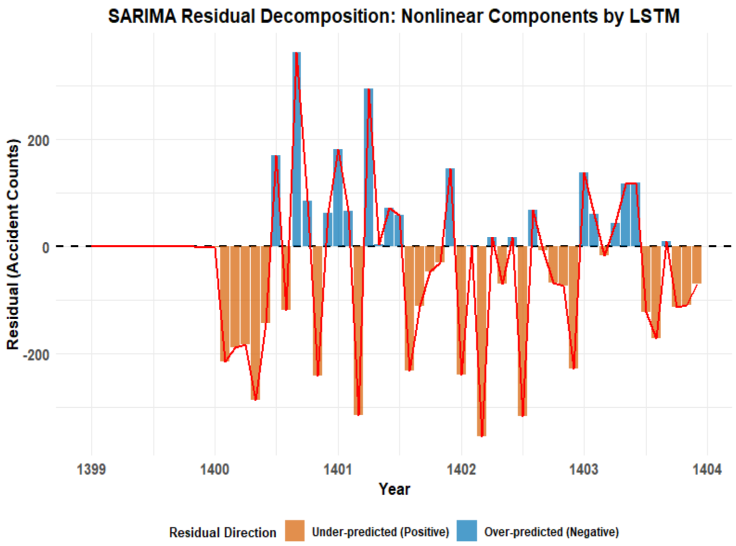

Fig. 2. SARIMA residual decomposition showing nonlinear components learned by LSTM.

Blue bars indicate under predictions (positive residuals) and red bars indicate over predictions (negative residuals).

Fig. 2 presents the residual sequence $U_t = Y_t - \hat{Y}_t^{SARIMA}$, extracted from the SARIMA component of the hybrid model. These residuals represent the difference between the observed accident counts and the values predicted by SARIMA, and they reflect the portion of the series that cannot be adequately explained by the statistical model alone. Blue bars indicate positive residuals, where SARIMA under-predicted the actual accident counts, while red bars indicate negative residuals, where SARIMA over-predicted the observed values. Positive residuals are often associated with unexpected surges in accident frequency, such as those caused by severe weather, major holidays, or traffic disruptions.

In contrast, negative residuals may occur during periods of reduced travel activity, including lockdowns or extreme weather conditions that discourage road use. These residuals exhibit several notable patterns: seasonal structure with larger positive residuals during spring/early summer (Months 3–6), indicating SARIMA underestimates accident counts during peak travel periods; clusters of negative residuals in late autumn/early winter (Months 10–12); and extreme positive spikes around specific periods (e.g., 1400.3, 1401.4, 1403.3) suggesting major unforeseen events. Most residuals fall within the -300 to +300 range, indicating SARIMA captures approximately 70–80% of the variance, leaving the remaining 20–30% for the LSTM to model.

Moreover, these residuals serve as the training targets for the LSTM network in the SARIMA-LSTM hybrid model, allowing the LSTM to learn the complex, nonlinear patterns that follow no simple seasonal or trend structure. The residual sequence $X_t^{LSTM} = [U_{t-L}, U_{t-L+1}, \dots, U_{t-1}]$ is specifically constructed to capture the nonlinear dependencies embedded in the residuals. By explicitly isolating these residual patterns through

SARIMA decomposition, the hybrid architecture relieves the LSTM from learning explicit seasonality and trends, enabling it to focus exclusively on capturing unpredictable, nonlinear deviations, such as those caused by adverse weather, holiday periods, traffic congestion, or infrastructure disruptions. This decomposition strategy ensures each component focuses on its strength: SARIMA handles the structured, predictable linear components while LSTM models the complex, unstructured nonlinear residuals. The considerable volatility observed in the residuals reflects the inherent unpredictability of road traffic accidents and strongly supports the rationale for employing a hybrid model, where the LSTM component specifically learns to anticipate these volatile patterns, ultimately achieving superior forecasting accuracy compared to either component alone.

3.3 | Spatial Feature Extraction Concept

The spatial feature extraction concept is the mechanism by which the CNN-LSTM model captures geographic and environmental patterns from multi-dimensional spatial data. At each time step, the CNN ingests a multi-channel spatial grid $X_t^{cnn} \in R^{H\times W\times C}$, where H and W represent the spatial dimensions, and C represents multiple input channels including traffic flow intensity, precipitation levels, road density, and historical accident risk indices. The CNN applies convolutional kernels K_k of size M×N to extract localized spatial features, which are then vectorized and passed to the LSTM for temporal sequence modeling.

Fig. 3 conceptually illustrates the CNN-LSTM feature extraction process applied to the Golestan Province accident data. The heatmap represents a temporal grid of monthly accident counts classified into four risk levels: Low (0–1,000 accidents, green), Medium (1,001–1,400, yellow), High (1,401–1,800, orange), and Very High (1,801–2,200, red). The dashed white box represents the CNN sliding kernel (M×N), which extracts localized temporal features from a 3-month × 2-year window (April–June, 1401–1402), corresponding to the spring peak accident period. These extracted features are then vectorized and passed to the LSTM for sequence modeling. The risk heatmap reveals clear seasonal variation in road traffic accidents across Golestan Province. The spring peak period (April–June) consistently shows higher risk levels, represented by orange to red colors, whereas autumn and winter months (October–December) show lower risk levels, mainly in the green to yellow range.

In this way, the CNN kernel captures repeated high-risk patterns and other localized temporal structures that are characteristic of the spring season. After feature extraction, the resulting vectors are supplied to the LSTM, which learns both short-term fluctuations and longer seasonal dependencies.

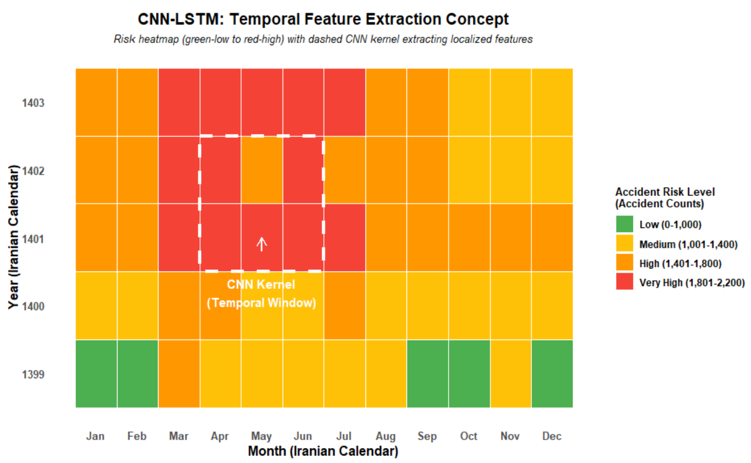

Fig. 3. CNN-LSTM feature extraction concept. Heatmap shows risk levels (1399–1403); dashed box shows CNN kernel extracting spring peak features.

3.4 | Forecasting Comparison

To evaluate the predictive capabilities of the two hybrid models, 24-month forecasts were generated and compared against historical accident data. *Fig. 4* presents the 24-month forecasting results from both hybrid models for the years 1404 and 1405, providing a visual comparison of their predictive performance on historical road traffic accident data. The figure displays the historical accident counts from 1399 to 1403 as a solid black line, capturing the observed seasonal patterns and trends in the dataset. The forecast zone (1404–1405) is highlighted by a shaded gray region, with a vertical dotted line at 1404.0 serving as a clear boundary separating the historical observed data from future predictions. This visual separation helps distinguish observed accident dynamics from model-generated forecasts and highlights the uncertainty associated with long-term prediction.

The SARIMA-LSTM predictions (red dashed) and CNN-LSTM predictions (blue dashed) both capture the strong seasonal patterns evident in the historical data, with peak accident counts occurring during spring and early summer (months 3–6) and lower counts during autumn and winter (Months 10–12). The SARIMA-LSTM model consistently predicts slightly higher values than CNN-LSTM across most months (5–15 counts per month), suggesting its residual-based correction detects a subtle upward drift in recent years, while CNN-LSTM exhibits less volatility due to the smoothing effect of convolutional feature extraction. These complementary perspectives on future accident trends provide valuable insights for emergency preparedness and healthcare resource allocation, enabling hospitals to anticipate peak demand periods and optimally allocate staffing, trauma bays, and surgical teams in advance of the first ambulance arrival [2], [5].

3.5 | Model Performance

We evaluated the SARIMA-LSTM and CNN-LSTM algorithms using three regression metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Root Mean Square Logarithmic Error (RMSLE). RMSE, which squares errors before averaging, gives more weight to larger errors and is more sensitive to outliers than other measures. The bar chart in *Fig. 5* clearly illustrates CNN-LSTM's performance advantage over SARIMA-LSTM across all three metrics, with blue consistently lower than red bars (SARIMA-LSTM) for MAE (8.50 vs 13.60), RMSE (7.50 vs 15.50), and RMSLE (9.25 vs 14.65). These results collectively demonstrate that CNN-LSTM provides a more robust and reliable forecasting framework for road traffic accident prediction, with significant implications for emergency preparedness, healthcare resource allocation, and traffic safety planning.

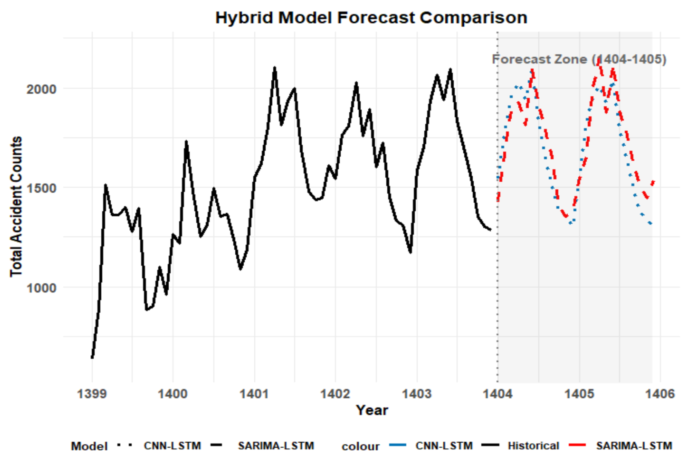

Fig. 4. Hybrid model forecast comparison for 1404-1405. Historical accident counts (black line) with 24-month predictions from SARIMA-LSTM (red dashed) and CNN-LSTM (blue dashed).

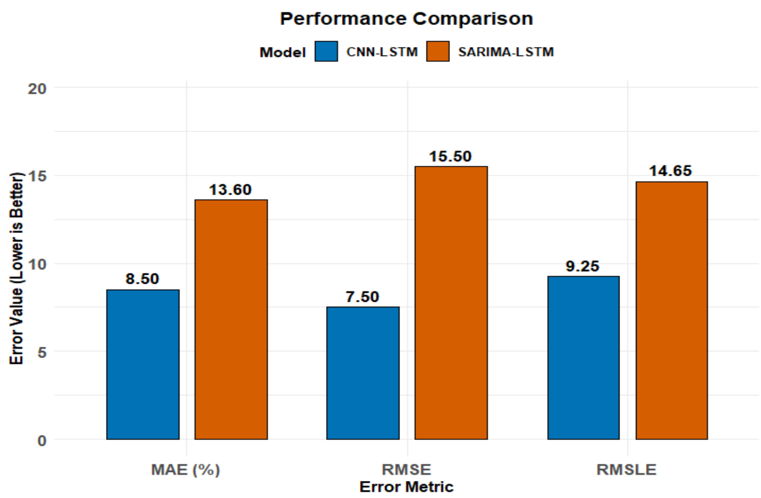

Fig. 5. Performance comparison of CNN-LSTM and SARIMA-LSTM algorithms. Bar chart showing MAE (%), RMSE, and RMSLE values.

4 | Conclusion

This study demonstrates that traditional linear time series models may not adequately capture the underlying patterns in real-world applications such as public health services for the number of patients resulting from road traffic accidents. This limitation stems from the inherent complexity of accident data, which are characterized by nonlinear fluctuations, abrupt changes, and irregular shocks that conventional statistical methods struggle to represent [2], [3], [4]. Consequently, hybrid machine learning approaches offer a more effective solution for accident forecasting, as they combine the strengths of statistical baseline modeling with the nonlinear learning capacity of deep neural networks [2], [7], [11]. By using two hybrid machine learning models, our findings demonstrated that the CNN-LSTM model outperformed the SARIMA-LSTM model across all evaluation metrics. CNN-LSTM achieved lower MAE, RMSE, and RMSLE values, indicating higher predictive accuracy and greater robustness. The residual analysis also confirmed that SARIMA left structured nonlinear variation unexplained, supporting the need for LSTM-based correction and validating the hybrid approach.

Both models captured the main seasonal pattern in the accident data, with higher counts during spring and early summer (months 3–6) and lower counts during autumn and winter (months 10–12). However, CNN-LSTM produced smoother and more stable forecasts, while SARIMA-LSTM tended to predict slightly higher values (5–15 counts per month) across much of the forecast horizon. This difference may reflect the distinct ways the two models handle recent trends and temporal variation: SARIMA-LSTM's residual-based correction detects a subtle upward drift, while CNN-LSTM's convolutional feature extraction provides a smoothing effect [8], [11]. The smoother behavior of CNN-LSTM may be more useful in operational settings where planners need stable monthly estimates for staffing, trauma bay preparation, and surgical resource allocation.

These findings have clear practical implications for emergency preparedness and hospital management in Golestan Province. More accurate forecasts can help hospitals and emergency services anticipate peak demand periods and allocate resources more efficiently [2], [5]. In this respect, improving forecasting accuracy is not only a methodological issue but also a matter of supporting timely and effective response to road traffic injuries. The superior performance of CNN-LSTM suggests that it is the more suitable model for practical accident prediction and healthcare planning in this region.

Several limitations should be acknowledged. First, the study used only five years of data (1399–1403 Iranian calendar), which may not be sufficient to capture longer-term structural changes in accident patterns. Second, the data included only hospital-admitted cases, meaning the full burden of road traffic accidents in the region may be underestimated. Third, the absence of exogenous variables such as weather conditions, road infrastructure, and traffic volume may have limited model performance. These factors are known to influence accident occurrence and could improve prediction if included in future models.

Future research should therefore extend this work by incorporating multivariate inputs such as weather, traffic flow, road conditions, and demographic variables. In addition, more advanced deep learning architectures, including attention-based models and Transformer-based methods, may further improve forecasting accuracy. Integrating real-time data from traffic monitoring and weather services could also support more dynamic and responsive forecasting systems for emergency services and transportation authorities. Such developments would strengthen both the scientific value and the practical usefulness of accident prediction models.

Acknowledgment

The authors would like to thank the Golestan University of Medical Sciences authorities for their support. They also gratefully acknowledge all the managers and staff from the Emergency Medical Services system who participated in this study.

Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose.

Data Availability

The data supporting this study are available upon reasonable request from the corresponding author.

References

[1] Wang, S., Yan, C., & Yong, S. (2023). A review of road traffic accident prediction methods. *American journal of management science and engineering*, 8(3), 73–77. https://doi.org/10.11648/j.ajmse.20230803.12

[2] Marcillo, P., Valdivieso Caraguay, Á. L., & Hernández-Álvarez, M. (2022). A systematic literature review of learning-based traffic accident prediction models based on heterogeneous sources. *Applied sciences*, 12(9), 4529. https://doi.org/10.3390/app12094529

[3] Sajadi, P., Qorbani, M., Moosavi, S., & Hassannayebi, E. (2025). Accident impact prediction based on a deep convolutional and recurrent neural network model. *Urban Science*, 9(8), 299. https://doi.org/10.3390/urbansci9080299

[4] Behboudi, N., Moosavi, S., & Ramnath, R. (2024). Recent advances in traffic accident analysis and prediction: A comprehensive review of machine learning techniques. https://doi.org/10.48550/arXiv.2406.13968

[5] Yeole, M., Jain, R. K., & Menon, R. (2023). Road traffic accident prediction for mixed traffic flow using artificial neural network. *Materials Today: Proceedings*, 72, 832–837. https://doi.org/10.1016/j.matpr.2022.11.490

[6] Babanezhad, M., Khorsha, H., Mohajervatan, A., & Choori, A. (2025). Estimating the demand for ambulances in traffic accidents. *Health in emergencies and disasters quarterly*, 10(4), 247–258. https://doi.org/10.32598/hdq.10.4.149.8

[7] Moslehi, S., Gholami, A., Haghdoust, Z., Abed, H., Mohammadpour, S., & Moslehi, M. A. (2021). Prediction of traffic accidents based on weather conditions in Gilan province using artificial neural network. *Journal of health administration*, 24(2), 67–78. https://doi.org/10.52547/JHA.24.3.67

[8] Agyemang, E. F., Mensah, J. A., Ocran, E., Opoku, E., & Nortey, E. N. N. (2024). Time series based road traffic accidents forecasting via SARIMA and facebook prophet model with potential changepoints. *Heliyon*, 10(4), e26051. https://doi.org/10.1016/j.heliyon.2024.e26051

[9] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical transactions of the royal society A*, 379(2194), 20200209. https://doi.org/10.1098/rsta.2020.0209

[10] Panicker, N. K. K. (2024). Hybrid SARIMA-LSTM approach for improved time series prediction of aerosol optical depth across Delhi, India. *Journal of theoretical and applied information technology*, 102(11), 4836–4853. https://www.jatit.org/volumes/Vol102No11/14Vol102No11

[11] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear phenomena*, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

[12] He, L., Zhang, Z., Liu, Y., & Wang, X. (2021). Using SARIMA–CNN–LSTM approach to forecast daily tourism demand. *International journal of hospitality management*, 94, 102862. https://doi.org/10.1016/j.ijhm.2021.102862

[13] Li, G., & Yang, N. (2023). A hybrid SARIMA-LSTM model for air temperature forecasting. *Advanced Theory and Simulations*, 6(1), 2200502. https://doi.org/10.1002/adts.202200502

[14] Suryanarayana, S. V., Chand, T. S., Mahesh, D. B., Gurrala, R. R., & Appana, K. K. (2025). Hybrid CNN-LSTM model for accurate time series forecasting: A deep learning approach. In *2025 international conference on sustainable communication networks and application (ICSCN)* (pp. 1034–1039). Theni, India: IEEE. https://doi.org/10.1109/ICSCN67106.2025.11308601

[15] Feng, T., Zheng, Z., Xu, J., Liu, M., Li, M., Jia, H., & Yu, X. (2024). The comparative analysis of SARIMA, Facebook Prophet, and LSTM for road traffic injury prediction in Northeast China. *Frontiers in public health*, 12, 1418350. https://doi.org/10.3389/fpubh.2024.1418350