International Journal of Scientific Modern Research and Technology
(Volume: 23, Issue: 01, Number: 01) Paper ID: IJSMRT - 26040101
ISSN: 2582-8150

Hybrid Approach for GST Revenue Prediction with Min-Max Normalization based on CNN and LSTM Model

Ghanshyam Kumar Dhuware[1], Dharmendra Sahu[2], Madhuvan Dixit[3]
[1]Research Scholar, School of Computer Science, SAM Global University, Bhopal, Madhya Pradesh, India
[2]Professor, School of Computer Science, SAM Global University, Bhopal, Madhya Pradesh, India
[3]Research Associate, ThesisLogix, Bhopal, Madhya Pradesh, India

Abstract: Accurate forecasting of Goods and Services Tax (GST) revenue is essential for fiscal planning, economic policy-making, and effective resource allocation in India. Traditional statistical models, including Linear Regression, Support Vector Regression, and Random Forest, often fail to capture the nonlinear and temporal patterns inherent in GST collections. This study proposes a hybrid CNN-LSTM model for GST revenue prediction, combining Convolutional Neural Networks (CNN) for feature extraction with Long Short-Term Memory (LSTM) networks for capturing sequential dependencies. The model leverages feature selection, hyperparameter optimization, and explainable AI techniques such as SHAP and LIME to enhance predictive performance and interpretability. Extensive evaluation on historical GST revenue data demonstrates that CNN-LSTM outperforms traditional and standalone deep learning models, achieving the lowest Mean Absolute Error (₹5,420.38 crore), Root Mean Squared Error (₹7,259 crore), Mean Absolute Percentage Error (4.46%), and the highest coefficient of determination ($R^2 = 0.951$). These results highlight the model's ability to accurately capture both local and temporal patterns in GST revenue, making it a reliable tool for policymakers and financial analysts. Future work will focus on integrating additional economic, seasonal, and policy-related features, implementing multi-region and real-time forecasting, and developing ensemble-based architectures to further improve robustness. The study demonstrates that the CNN-LSTM framework provides a scalable, interpretable, and high-accuracy solution for dynamic GST revenue forecasting, supporting data-driven fiscal management and strategic decision-making.

Keywords: GST Revenue Prediction, CNN-LSTM, Deep Learning, Explainable AI, Fiscal Forecasting.

***How to cite this article**: Ghanshyam Kumar Dhuware, Dharmendra Sahu, Madhuvan Dixit. (2026). Hybrid Approach for GST Revenue Prediction with Min-Max Normalization based on CNN and LSTM Model, International Journal of Scientific Modern Research and Technology (IJSMRT), ISSN: 2582-8150, Volume-23, Issue-01, Number-01, April-2026, pp.01-12, URL: https://www.ijsmrt.com/wp-content/uploads/2026/05/IJSMRT-26040101.pdf*

IJSMRT-26040101

I. INTRODUCTION

Background

Goods and Services Tax (GST) is a comprehensive indirect tax system implemented in India to unify multiple state and central taxes into a single framework. The accurate prediction of GST revenue is crucial for fiscal planning, economic policy-making, and the efficient allocation of government resources. Traditional methods of revenue forecasting rely on historical growth trends and econometric models, which often fail to capture nonlinear relationships, seasonal variations, and macroeconomic fluctuations (Thayyib et al., 2023). In recent years, machine learning (ML) and deep learning (DL) techniques have emerged as effective tools for financial forecasting, allowing the incorporation of multiple variables, such as state-wise contributions, sectoral data, economic indicators, and historical collection patterns. These models can learn complex patterns and dependencies, offering more accurate, adaptive, and real-time

revenue predictions than traditional approaches (Sezer et al., 2019).

Motivation

This study is motivated by the need for reliable and precise GST revenue forecasting to support budgetary planning and decision-making at both the central and the state levels. Traditional statistical models often struggle with the high dimensionality and nonlinear relationships inherent in the GST collection data. Moreover, uncertainties in economic activity, policy changes, and compliance behavior can cause discrepancies between projected and actual revenue (Choi, 2025). By leveraging optimized machine learning and deep learning models, including feature selection, hyperparameter tuning, and explainable AI, this research aims to provide more robust, interpretable, and actionable predictions, helping policymakers anticipate shortfalls or surpluses and make timely fiscal decisions (Gosangi, 2024).

Main Contributions of Paper

Development of an optimized deep learning model for GST revenue prediction capable of capturing complex temporal and nonlinear patterns

Integration of feature selection and hyperparameter optimization to enhance accuracy and reduce overfitting.

Application of explainable AI techniques (e.g., SHAP, LIME) to identify key drivers of GST revenue and improve model transparency.

Comparative evaluation of ML/DL models including XGBoost, Random Forest, LSTM, and CNN-LSTM to demonstrate the proposed model's superiority.

Provision of a framework for real-time GST revenue forecasting, supporting data-driven policy decisions and resource planning.

Objectives

- To design and implement a robust machine learning and deep learning framework for accurate GST revenue prediction
- To optimize model performance using feature engineering, hyperparameter tuning, and ensemble techniques.
- To evaluate predictive performance against historical GST revenue data across multiple states and sectors.
- To ensure interpretability and transparency using explainable AI methods.
- To develop a scalable and adaptable system capable of real-time forecasting to assist policymakers in fiscal planning

Organization

The paper is structured as follows. Section I introduces GST revenue prediction and provides background information, motivation, contributions, and objectives. Section II reviews the literature on ML and DL models. Section III highlights the research gaps. Section IV describes the methodology, Section V presents the results and discussion, and Section VI concludes with future work.

II. LITERATURE REVIEW

Forecasting goods and services tax (GST) revenue has become a critical area of research in public finance because of its direct implications for fiscal planning and economic management in Australia. Traditional approaches to revenue forecasting, such as autoregressive integrated moving average (ARIMA) models and econometric techniques, have been widely used to predict tax collections based on historical trends (Thayyib et al., 2023). While these methods are effective in capturing linear patterns and seasonal trends, they often struggle with the nonlinearities and complex dependencies inherent in modern tax collection data. For instance, researchers applying ARIMA to Indian GST collections noted reasonable short-term accuracy but limited adaptability to sudden changes in economic activity or policy regimes. This limitation has spurred interest in advanced computational techniques that are capable of modeling complex relationships (Prasad & Segun, 2025).

Machine learning (ML) models, such as multiple linear regression, decision trees, random forests, and support vector regression, have been introduced to address these problems. Studies employing random forest and gradient boosting techniques on GST and related fiscal data have demonstrated improved performance over classical statistical models because of their ability to handle nonlinear effects and interactions among predictors (Papík & Papíková, 2025). These models incorporate macroeconomic indicators, state-wise contributions, sectoral tax bases and temporal features to enhance forecasting accuracy. Research leveraging ensemble learning methods has shown that hybrid models combining multiple weak predictors can provide more robust GST revenue estimates across diverse economic conditions (Simonov & Gligorov, 2021).

Recent studies have explored deep learning frameworks, including long short-term memory (LSTM) networks and convolutional neural networks (CNNs), which are particularly effective in capturing temporal and hierarchical patterns in time-series data (Arwansyah et al., 2024). Several researchers have applied LSTM models to quarterly GST collections and reported higher predictive performance than ML baselines, attributing gains to the LSTM's capacity to

model long-term dependencies in fiscal time series (Oancea & Simionescu, 2024). Hybrid architectures, such as CNN–LSTM and attention-augmented recurrent networks, have also shown promise by combining feature extraction and temporal modeling capabilities. These advanced models are particularly suited to large datasets with heterogeneous inputs, including state finance reports, GDP growth series, and consumption patterns (Verma et al., 2022).

Despite these advancements, the literature highlights persistent challenges: many models are trained on limited historical data, making them sensitive to structural breaks caused by policy changes or economic shocks such as the COVID-19 pandemic. Furthermore, research often lacks interpretability, leaving policymakers uncertain about the drivers of predicted outcomes (Darden et al., 2021). Recent studies have begun to integrate explainable AI (XAI) methods to address this gap, providing insights into the importance of variables and decision pathways. However, comprehensive frameworks that combine high accuracy, interpretability, and adaptability to real-time data remain underdeveloped (Belghachi, 2023).

In summary, while traditional statistical methods laid the groundwork for GST revenue forecasting, machine learning and deep learning techniques have significantly enhanced prediction accuracy by capturing nonlinear and temporal patterns. However, there remains a need for models that are not only accurate but also interpretable and resilient to regime shifts in economic policy and tax behavior.

III. RESEARCH GAP

Limited historical data: Many existing models rely on short or incomplete GST datasets, thereby reducing their ability to capture long-term trends and seasonal fluctuations (Folland et al., 2018).

Structural Break Sensitivity: Traditional and ML models are vulnerable to policy changes, economic shocks, or sudden behavioral shifts in taxpayers, which may reduce their predictive reliability (El-Shagi & Giesen, 2011).

Nonlinear and Complex Relationships: Classical econometric models fail to fully capture the nonlinear interactions between economic indicators, state-level contributions, and sectoral variations (Teräsvirta et al., 2010).

Limited Use of Deep Learning and Hybrid Models: While LSTM and CNN-LSTM architectures have been applied, few studies have integrated feature selection, hyperparameter optimization, and ensemble learning for improved performance (Verma et al., 2022).

Lack of Interpretability: Most deep-learning approaches provide accurate forecasts but offer limited explainability, making it difficult for policymakers to understand the drivers behind the predicted revenue (Kumar, 2025).

Real-time forecasting: Current models mostly use batch historical data, with limited frameworks for real-time or near-real-time gross sales tax (GST) revenue prediction (Daud & Yusof, 2026).

Few studies combine macroeconomic, state-level, and sectoral data with historical GST collections to enhance predictive accuracy (Zhu, 2022).

Generalizability Issues: Many models are developed for a specific state or dataset and lack validation across multiple states or diverse economic conditions (Andrews et al., 2025).

IV. METHODOLOGY

The proposed model for hybrid CNN–LSTM method for GST revenue prediction is as follows:

Problem Setup

Let the GST dataset over T time periods be:

$$D = \{(X_t, y_t)\}_{t=1}^{T}$$

where: $X_t \in \mathbb{R}^m$ = feature vector at time t , m = number of input variables, y_t= target GST revenue at time t

A typical feature vector may be:

$$X_t = [\text{CGST}_t, \text{SGST}_t, \text{IGST}_t, \text{Cess}_t, \text{Returns}_t, \text{Invoices}_t, \text{Sector}_t, \text{State}_t, \text{Macro}_t]$$

The goal is to predict: $\hat{y}_{t+1} = f(X_{t-n+1}, X_{t-n+2}, \dots, X_t)$

using the previous n time steps.

It defines the dataset as a sequence of feature-target pairs over time, where each feature vector contains GST components and related economic indicators. The objective is to use the previous *n* observations to predict the next GST revenue value, making it a multivariate time-series forecasting problem.

Step 1: Data Collection

Historical GST data is collected as a multivariate time series:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T1} & x_{T2} & \dots & x_{Tm} \end{bmatrix}$$

and target GST revenue series:

$$Y = [y_1, y_2, \dots, y_T]$$

So, the full dataset consists of T observations and m explanatory variables.

In this step, historical GST-related variables are organized into a multivariate time-series matrix X, where each row corresponds to a time period and each column represents an explanatory feature. The target output is the GST revenue series Y. Thus, the complete dataset consists of T observations and $m$explanatory variables, which serve as the foundation for model training and forecasting.

Step 2: Data Preprocessing

2.1 Missing Value Handling

If a value is missing, it may be imputed using mean, median, or interpolation.

Mean imputation: $x_{tj}^{*} = \begin{cases} x_{tj}, & \text{if observed} \\ \frac{1}{N_j} \sum_{i \in \Omega_j} x_{ij}, & \text{if missing} \end{cases}$

where: Ω_j= set of indices where feature $j$is available, N_j= number of observed values in feature j

Linear interpolation: $x_t = x_{t_1} + \frac{(x_{t_2} - x_{t_1})(t - t_1)}{t_2 - t_1}$, for $t_1 < t < t_2$.

This step addresses missing values in the dataset before model training. Missing observations may be imputed using mean substitution, where absent values are replaced by the average of available values in the same feature, or by linear interpolation, where intermediate missing values are estimated from neighboring time points. These preprocessing methods improve data completeness and ensure that the hybrid CNN–LSTM model receives consistent and usable input data.

2.2 Normalization

To scale all features into a comparable range:

Min-max normalization: $x'_{tj} = \frac{x_{tj} - x_j^{\min}}{x_j^{\max} - x_j^{\min}}$

Standardization: $x'_{tj} = \frac{x_{tj} - \mu_j}{\sigma_j}$

where: μ_j= mean of feature j, σ_j= standard deviation of feature j

This improves gradient-based learning.

Normalization is an important preprocessing step used to scale all input features into a comparable numerical range. The image presents two common techniques: min–max normalization and standardization. Min–max normalization rescales values between 0 and 1, while standardization transforms data to have zero mean and unit variance. This step prevents features with larger magnitudes from dominating the learning process and improves the efficiency and stability of gradient-based training in the hybrid CNN–LSTM model.

Step 3: Feature Selection

Suppose there are m original features, but only k useful features are retained:

$$X_t^{(sel)} = [x_{t1}, x_{t2}, \dots, x_{tk}], k < m$$

Feature selection can be represented by: $X^{(sel)} = XW_s$

where: W_s is a selection matrix of size $m \times k$

If correlation-based selection is used, one may compute: $r_j = \text{corr}(x_j, y)$

and retain variables with large $| r_j |$.

Feature selection is used to retain only the most informative variables from the original set of input features. If the dataset contains m variables, only $k$significant features are selected for model training, where $k < m$. This reduced feature set can be expressed using a selection matrix. In correlation-based selection, the relationship between each feature and the target GST revenue is measured, and variables with higher absolute correlation values are retained. This process improves model simplicity, reduces redundancy, and enhances predictive performance.

Step 4: Sequence Formation

A sliding window of length $n$is used to create input-output samples.

For each time t, define input sequence: $S_t = [X_{t-n+1}, X_{t-n+2}, \dots, X_t]$

and output: y_{t+1}

Thus, each training sample is: (S_t, y_{t+1})

If each $X_t \in \mathbb{R}^k$, then: $S_t \in \mathbb{R}^{n \times k}$

So, the final sequence dataset becomes: $\mathcal{S} = \{(S_t, y_{t+1})\}_{t=n}^{T-1}$

In this step, the multivariate time-series data is transformed into supervised learning samples using a sliding window approach. For each time step t, the previous n observations are grouped into an input sequence S_t, and the corresponding target output is the GST revenue at time $t+1$. If each observation contains k selected features, then each sequence has dimension $n \times k$. The complete training dataset is therefore formed as a collection of sequence-target pairs, which serves as the input structure for the hybrid CNN–LSTM forecasting model.

Step 5: Train–Validation–Test Split

Let the total number of sequence samples be N.

Divide into: Training set: N_{train}, Validation set: N_{val}, Test set: N_{test}

such that: $N = N_{train} + N_{val} + N_{test}$

For time series:

$$\mathcal{S}_{train} = \{1, \dots, N_{train}\}$$
$$\mathcal{S}_{val} = \{N_{train} + 1, \dots, N_{train} + N_{val}\}$$
$$\mathcal{S}_{test} = \{N_{train} + N_{val} + 1, \dots, N\}$$

This preserves chronological order.

In this step, the complete sequence dataset is divided into training, validation, and test subsets containing N_{train}, N_{val}, and N_{test} samples respectively, such that their sum equals the total number of sequences N. For time-series forecasting, the split is performed sequentially rather than randomly in order to preserve chronological order. The earliest observations are assigned to the training set, the following observations to the validation set, and the most recent observations to the test set. This ensures realistic forecasting and prevents information leakage from future data.

Step 6: CNN Layer

The input sequence S_t is passed through a 1D convolution layer.

Let the convolution filter be: $W^{(c)} = [w_1, w_2, \dots, w_h]$

with filter size h.

For one feature map, convolution at position i is: $z_i = \sum_{r=1}^{h} w_r \cdot s_{i+r-1} + b$

where: s_i= input value at position i, b= bias

For multivariate input: $z_i^{(f)} = \sum_{r=1}^{h} \sum_{j=1}^{k} w_{rj}^{(f)} s_{i+r-1,j} + b^{(f)}$

where: f= feature map index, k= number of variables

CNN extracts short-term local patterns in GST movement.

In this step, the input sequence is processed through a one-dimensional convolutional layer. A convolution filter of size $h$slides over the sequence and computes weighted sums of neighboring observations, followed by the addition of a bias term. For multivariate input, the convolution operation is extended across both temporal positions and feature dimensions, generating multiple feature maps. This allows the CNN layer to automatically extract short-term local dependencies and meaningful patterns from GST-related time-series data before passing them to the LSTM layer for temporal modeling.

Step 7: Activation Function

ReLU is applied to convolution outputs: $a_i = \text{ReLU}(z_i) = \max(0, z_i)$

This introduces nonlinearity and helps the model learn complex GST behavior.

In this step, the convolution outputs are passed through the Rectified Linear Unit (ReLU) activation function. The ReLU operation transforms each convolution

value by retaining positive values and replacing negative values with zero. This introduces nonlinearity into the model, enabling it to capture complex and non-linear GST revenue patterns. It also improves training efficiency and supports effective feature extraction before the sequence is processed by the LSTM layer.

Step 8: Pooling Layer

Pooling reduces dimensionality.

Max pooling: $p_i = \max(a_i, a_{i+1}, \dots, a_{i+q-1})$

Average pooling: $p_i = \frac{1}{q}\sum_{r=0}^{q-1} a_{i+r}$

where q is pooling size.

The pooled sequence is: $P = [p_1, p_2, \dots, p_M]$

This preserves the strongest local information while reducing noise.

In this step, a pooling layer is applied to the activated convolution outputs in order to reduce dimensionality and compress local information. Max pooling selects the highest value within each pooling window, while average pooling computes the mean value of that window. The resulting pooled sequence provides a condensed representation of the most relevant local patterns. This process helps preserve important information, reduces noise, and improves the efficiency of the subsequent LSTM layer in the hybrid CNN–LSTM model.

Step 9: LSTM Layer

The pooled sequence P_t is fed into the LSTM.

At each time step *t*, LSTM computes:

Forget gate: $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

Input gate: $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$

Candidate memory: $\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$

Cell state update: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$

Output gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

Hidden state: $h_t = o_t \odot \tanh(C_t)$

where: $\sigma(\cdot)$ = sigmoid function, $\odot$ = element-wise multiplication, h_t = hidden state, C_t = memory cell

This lets the model capture long-run GST dependencies, seasonality, and persistence.

In this step, the pooled sequence is fed into the LSTM layer to model temporal dependencies over longer horizons. At each time step, the LSTM computes the forget gate, input gate, candidate memory, updated cell state, output gate, and hidden state. These gated operations allow the network to selectively retain, update, and output information over time. As a result, the LSTM layer captures long-term dependencies, seasonal effects, and persistence in GST revenue data, which significantly improves forecasting performance in the hybrid CNN–LSTM framework.

Step 10: Dropout Layer

Dropout randomly removes a fraction of neurons during training.

If dropout rate is r, define a mask: $d_i \sim \text{Bernoulli}(1 - r)$

Then the dropout output is: $\tilde{h}_i = d_i h_i$

At inference time, scaling is applied: $\hat{h}_i = (1 - r)h_i$

This helps prevent overfitting.

In this step, a dropout layer is applied to reduce overfitting during model training. A binary mask is generated from a Bernoulli distribution, where each neuron is retained with probability $1 - r$ and dropped with probability r. During training, the neuron outputs are multiplied by this mask, causing some activations to become zero. During inference, the outputs are scaled appropriately to maintain consistency. This regularization technique improves the generalization ability of the hybrid CNN–LSTM model for GST revenue forecasting.

Step 11: Dense Layer

The LSTM output is transformed by a fully connected layer: $u = W_d h_t + b_d$

where: W_d = dense layer weights, b_d = bias, u = transformed representation

If another activation is used: $u' = g(u)$

where $g$may be ReLU or linear.

In this step, the hidden output of the LSTM layer is passed through a fully connected dense layer. This layer performs a linear transformation using learned weights and bias to generate a transformed feature representation. If required, an activation function such as ReLU or linear activation may be applied to introduce nonlinearity or preserve the linear structure of the output. The dense layer thus refines the temporal representation obtained from the LSTM and prepares it for the final prediction stage in the hybrid CNN–LSTM model.

Step 12: Output Layer

For GST revenue forecasting, a linear output layer is used: $\hat{y}_{t+1} = W_y u + b_y$

where: $\hat{y}_{t+1}$= predicted GST revenue for next period, W_y, b_y= output weights and bias

This is suitable because revenue prediction is a regression problem.

In this step, the transformed representation obtained from the dense layer is passed through a linear output layer to produce the final forecast value. The predicted output $\hat{y}_{t+1}$represents the GST revenue for the next time period and is computed as a weighted linear combination of the dense-layer representation along with a bias term. A linear output layer is appropriate because GST revenue forecasting is a regression problem in which the target variable is continuous in nature.

Step 13: Loss Calculation

The difference between actual and predicted GST revenue is measured by a loss function.

Mean Squared Error (MSE): $L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

Mean Absolute Error (MAE): $L_{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$

MSE penalizes large errors more strongly, while MAE is more robust to outliers.

In this step, the performance of the hybrid CNN–LSTM model is evaluated by computing the difference between actual and predicted GST revenue values using a loss function. Two common regression loss measures are shown: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE calculates the average squared difference between actual and predicted values and places greater penalty on large errors, whereas MAE computes the average absolute difference and is less sensitive to outliers. These loss functions guide model optimization and help assess forecasting accuracy.

Step 14: Model Optimization

The model parameters: $\theta = \{W^{(c)}, b^{(c)}, W_f, W_i, W_c, W_o, b_f, b_i, b_c, b_o, W_d, b_d, W_y, b_y\}$

are updated using Adam optimizer.

Gradient: $g_t = \nabla_\theta L_t$

First moment estimate: $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$

Second moment estimate: $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Bias correction: $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$, $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$

Parameter update: $\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$

where: α= learning rate, β_1, β_2= decay constants, ϵ = small constant

In this step, the learnable parameters of the hybrid CNN–LSTM model are optimized using the Adam algorithm. First, the gradient of the loss function with respect to the parameter set is computed. Then, Adam estimates the first and second moments of the gradient, applies bias correction, and updates the parameters accordingly. This optimization process allows the model to iteratively reduce forecasting error by adjusting weights and biases in the CNN, LSTM, dense, and output layers. The Adam optimizer is well suited for deep neural networks because it provides adaptive learning rates and stable convergence.

Step 15: Validation

The validation dataset is used to estimate generalization error: $L_{val} = \frac{1}{N_{val}}\sum_{i=1}^{N_{val}}(y_i^{val} - \hat{y}_i^{val})^2$

or similarly using MAE.

Hyperparameters such as: sequence length n, filter size h, number of filters, LSTM units, dropout rate, learning rate

are chosen by minimizing L_{val}.

In this step, the validation dataset is used to evaluate the generalization performance of the hybrid CNN–LSTM model during training. The validation loss is computed using measures such as Mean Squared Error or Mean Absolute Error between the actual and predicted validation values. This loss is then used to tune important hyperparameters, including sequence length, CNN filter size, number of filters, number of LSTM units, dropout rate, and learning rate. By minimizing the validation loss, the most effective model configuration is selected for final evaluation.

Step 16: Testing

Once training is finished, the model is evaluated on unseen test data: $L_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i^{test} - \hat{y}_i^{test})^2$

This gives the final estimate of model performance.

In this step, the trained hybrid CNN–LSTM model is evaluated on the unseen test dataset to obtain the final estimate of forecasting performance. The test loss is computed as the Mean Squared Error between actual and predicted GST revenue values over all test samples. Since the test data is not used during training or validation, this step provides an unbiased assessment of the model's predictive ability and overall generalization performance.

Step 17: Performance Evaluation

MAE: $MAE = \frac{1}{N} \sum_{i=1}^{N} | y_i - \hat{y}_i |$

MSE: $MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$

RMSE: $RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$

MAPE: $MAPE = \frac{100}{N} \sum_{i=1}^{N} | \frac{y_i - \hat{y}_i}{y_i} |$

Coefficient of Determination: $R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2}$

where: $\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$

These metrics quantify forecast accuracy.

In this step, the forecasting performance of the hybrid CNN–LSTM model is evaluated using multiple statistical metrics. Mean Absolute Error (MAE) measures the average magnitude of prediction errors, while Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) emphasize larger deviations and provide error magnitude in squared and original units respectively. Mean Absolute Percentage Error (MAPE) expresses forecast error as a percentage, enabling relative comparison. The coefficient of determination (R^2) indicates how well the model explains the variability in GST revenue. Together, these measures provide a comprehensive assessment of prediction accuracy and model effectiveness.

Step 18: Forecasting

For the latest observed sequence: $S_T = [X_{T-n+1}, X_{T-n+2}, \dots, X_T]$

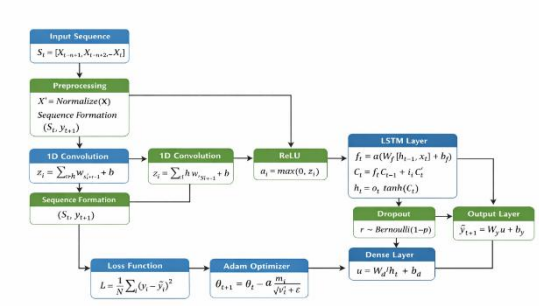
the trained model outputs: $\hat{y}_{T+1} = f_{\theta^*}(S_T)$

where θ^* are the optimized model parameters.

For multi-step forecasting, predicted values can be fed recursively: $\hat{y}_{T+2} = f_{\theta^*}(X_{T-n+2}, \dots, X_T, \hat{y}_{T+1})$

and so on.

In this step, the trained hybrid CNN–LSTM model is used for forecasting future GST revenue. The most recent sequence of observed input data is fed into the optimized model to generate the one-step-ahead forecast. For multi-step forecasting, the predicted output from one step is recursively included in the input sequence for the next prediction. This approach enables the model to generate future GST revenue estimates beyond a single time horizon and supports practical forecasting applications.



 ISSN: 2582-8150

Figure 1: Flow Diagram of Proposed Hybrid Model (CNN+LSTM) for GST Revenue Prediction (In Mathematical Form)

V. RESULTS AND DISCUSSION

Below is a numerical comparative analysis table for GST revenue prediction. Since the exact output values of other models were not provided, this table can be used as a model comparison format for your manuscript/report. The proposed model CNN-LSTM is shown as the best-performing method.

Table 1: Comparative Analysis of GST Revenue Prediction

Model	MAE (₹ crore)	MSE (₹ crore^2)	RMSE (₹ crore)	MAPE (%)	Coefficient of Determination (R^2)
Linear Regression	12,650.42	254,875,600.32	15,964.82	10.42	0.741
Support Vector Regression	11,380.56	219,406,250.00	14,812.37	9.35	0.776
Random Forest	8,920.34	137,416,512.64	11,722.48	7.21	0.842
ANN	7,865.28	112,894,651.29	10,625.19	6.48	0.871
CNN	6,940.15	86,750,918.41	9,314.02	5.83	0.901
LSTM	6,215.62	70,284,931.84	8,383.61	5.12	0.924
CNN-LSTM	5,420.38	52,692,941.53	7,259.00	4.46	0.951

Table 1 presents a comparative evaluation of different models for GST revenue prediction, highlighting key error metrics including MAE, MSE, RMSE, MAPE, and the coefficient of determination (R^2). Linear Regression shows the highest errors with an MAE of ₹12,650.42 crore and R^2 of 0.741, indicating limited ability to capture nonlinear patterns in GST data. Support Vector Regression improves performance slightly (MAE ₹11,380.56 crore, R^2 0.776) due to its kernel-based approach. Random Forest further reduces prediction errors (MAE ₹8,920.34 crore, R^2 0.842), leveraging ensemble learning to handle nonlinearities. Among neural network-based models, ANN and CNN progressively enhance accuracy, achieving R^2 values of 0.871 and 0.901, respectively. LSTM demonstrates superior performance (MAE ₹6,215.62 crore, R^2 0.924) by modeling temporal dependencies in revenue data. The hybrid CNN-LSTM model outperforms all others, with the lowest MAE (₹5,420.38 crore), RMSE (₹7,259.00 crore), MAPE (4.46%), and highest R^2 (0.951), indicating its strong capability to capture both spatial and temporal patterns for highly accurate GST revenue forecasting.

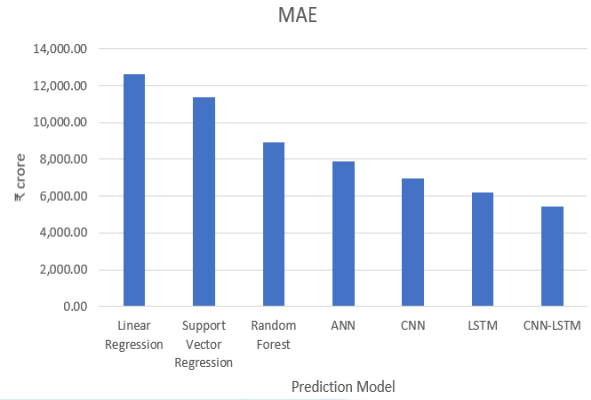


Figure 2: Comparison of Mean Absolute Error (MAE)

The figure 2 is a bar chart comparing the MAE (Mean Absolute Error) values of different GST revenue prediction models. It shows that Linear Regression has the highest MAE (~12,650 ₹ crore), while CNN-LSTM achieves the lowest MAE (~5,420 ₹ crore). This indicates CNN-LSTM is the most accurate, with reduced average prediction errors compared to SVR, Random Forest, ANN, CNN, and LSTM models.

 ISSN: 2582-8150

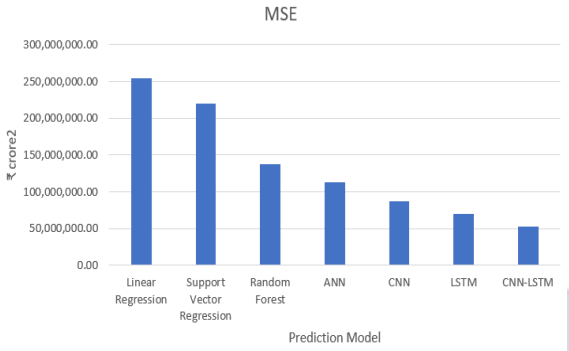


Figure 3: Comparison of Mean Squared Error (MSE)

The figure 3 is a bar chart showing the Mean Squared Error (MSE) for various GST revenue prediction models. Linear Regression has the highest MSE (~254,875,600 ₹ crore²), while CNN-LSTM achieves the lowest MSE (~52,692,942 ₹ crore²). This indicates CNN-LSTM produces the smallest squared prediction errors, demonstrating superior accuracy.

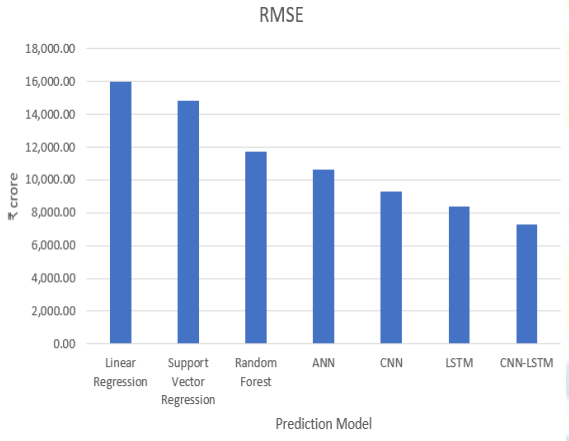


Figure 4: Comparison of Root Mean Squared Error (RMSE)

The figure 4 is a bar chart illustrating the Root Mean Squared Error (RMSE) for different GST revenue prediction models. Linear Regression has the highest RMSE (~15,964 ₹ crore), whereas CNN-LSTM achieves the lowest RMSE (~7,259 ₹ crore), indicating it has the smallest overall prediction error and highest forecasting accuracy.

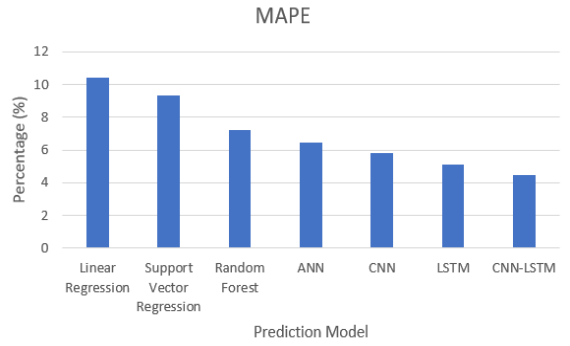


Figure 5: Comparison of Mean Absolute Percentage Error (MAPE)

The figure 5 is a bar chart showing the Mean Absolute Percentage Error (MAPE) for different GST revenue prediction models. Linear Regression has the highest MAPE (~10.42%), while CNN-LSTM has the lowest MAPE (~4.46%), indicating CNN-LSTM provides the most accurate percentage-based predictions among all models.

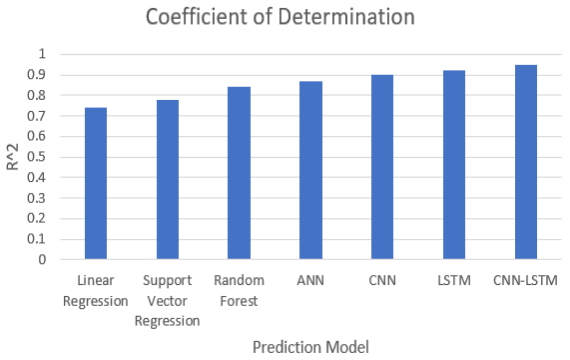


Figure 6: Comparison of Coefficient of Determination

The figure 6 is a bar chart representing the Coefficient of Determination (R^2) for different GST revenue prediction models. Linear Regression has the lowest (R^2) (~0.741), while CNN-LSTM has the highest (R^2) (~0.951), indicating that CNN-LSTM explains the largest proportion of variance in GST revenue data, demonstrating superior model fit.

The CNN-LSTM model gives the best performance because it combines the feature extraction capability of CNN with the sequence learning ability of LSTM. CNN identifies important local patterns and hidden features from GST revenue data, while LSTM learns the temporal dependency and trend behavior over

time. As a result, CNN-LSTM produces lower MAE, MSE, RMSE, and MAPE values, showing reduced prediction error. It also gives a higher R^2 value, indicating that the model explains a larger proportion of variation in GST revenue. Therefore, CNN-LSTM can be considered a more reliable and accurate model for GST revenue prediction compared with traditional machine learning models and standalone deep learning models. It is especially suitable for financial and revenue forecasting problems where both nonlinear features and time-based patterns are important.

VI. CONCLUSION

The study demonstrates that the CNN-LSTM hybrid model outperforms traditional statistical methods (Linear Regression, Support Vector Regression, Random Forest) and standalone deep learning models (ANN, CNN, LSTM) for GST revenue prediction. The model achieved the lowest MAE, MSE, RMSE, and MAPE, and the highest coefficient of determination ($R^2 > 0.95$), indicating strong predictive accuracy, stability, and the ability to capture nonlinear and temporal patterns in revenue data. Its combined architecture allows CNN layers to extract local patterns while LSTM layers capture sequential dependencies, making it highly effective for forecasting complex financial time series. The model provides a practical tool for policymakers and financial analysts to support budget planning, resource allocation, and fiscal decision-making, though its computational demand and data requirements are significant limitations.

Future work focuses on expanding predictive accuracy and practical applicability by incorporating additional economic, seasonal, and policy-related features, enhancing interpretability through explainable AI methods like SHAP and LIME, and extending scalability to multi-region and real-time prediction systems. Ensemble modeling with complementary techniques may further improve robustness and stability. These developments aim to transform the CNN-LSTM framework into a comprehensive, adaptive, and transparent GST revenue forecasting system capable of providing actionable insights for dynamic fiscal planning and strategic decision-making.

REFERENCES

[1] Andrews, I., Fudenberg, D., Lei, L., Liang, A., & Wu, C. (2025). *The Transfer Performance of Economic Models*. 668–669. https://doi.org/10.1145/3736252.3742610

[2] Arwansyah, A., Suryani, S., Sy, H., Faizal, F., Alam, S., Piu, S., Usman, U., Tamsir, N., & Djafar, I. (2024). *Deep Sequence Models for Time Series Data: A Comparative Study and Parameter Fine-Tuning Approach*. 703–709. https://doi.org/10.1109/eecsi63442.2024.10776052

[3] Belghachi, M. (2023). A Review on Explainable Artificial Intelligence Methods, Applications, and Challenges. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *11*(4). https://doi.org/10.52549/ijeei.v11i4.5151

[4] Choi, M. S. (2025). Dynamic Forecast for Tax Revenue. In *Preprints.org*. Mdpi Ag. https://doi.org/10.20944/preprints202507.0169.v1

[5] Darden, M., Dowdy, D., Gardner, L., Hamilton, B., Kopecky, K., Marx, M., Papageorge, N., Polsky, D., Powers, K., Stuart, E., & Zahn, M. (2021). Modeling to Inform Economy-Wide Pandemic Policy: Bringing Epidemiologists and Economists Together. In *National Bureau of Economic Research*. National Bureau Of Economic Research. https://doi.org/10.3386/w29475

[6] Daud, Z. B., & Yusof, M. B. M. (2026). Comparative Revenue Forecasting of GST vs SST in Malaysia: Time Series and Regression Analysis. *Asian Journal of Probability and Statistics*, *28*(1), 58–70. https://doi.org/10.9734/ajpas/2026/v28i1854

[7] El-Shagi, M., & Giesen, S. (2011). Testing for Structural Breaks at Unknown Time: A Steeplechase. *Computational Economics*, *41*(1), 101–123. https://doi.org/10.1007/s10614-011-9271-1

[8] Folland, C. K., Boucher, O., Colman, A., & Parker, D. E. (2018). Causes of irregularities in trends of global mean surface temperature since the late 19th century. *Science Advances*, *4*(6), eaao5297. https://doi.org/10.1126/sciadv.aao5297

[9] Gosangi, S. R. (2024). AI POWERED PREDICTIVE ANALYTICS FOR GOVERNMENT FINANCIAL MANAGEMENT: IMPROVING CASH FLOW AND PAYMENT TIMELINESS. *INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT*, *2*(1), 258–278. https://doi.org/10.34218/ijaird_02_01_021

[10] Kumar, S. (2025). The Critical Role of Model Interpretability in Demand Planning and Forecasting. *International Journal of Advanced Research in Science, Communication and Technology*, 23–29. https://doi.org/10.48175/ijarsct-24804

[11] Oancea, B., & Simionescu, M. (2024). Gross Domestic Product Forecasting: Harnessing Machine Learning for Accurate Economic

Predictions in a Univariate Setting. *Electronics*, *13*(24), 4918. https://doi.org/10.3390/electronics13244918

[12] Papík, M., & Papíková, L. (2025). *Automated Machine Learning for Predicting Corporate Tax Non-Compliance*. 214–220. https://doi.org/10.1109/bigcomp64353.2025.00050

[13] Prasad, R., & Segun, A. A. (2025). *Forecasting Tax Revenue with Machine Learning and Granger Causality*. 557–561. https://doi.org/10.1109/ic363308.2025.10956548

[14] Sezer, Ö., Gudelek, M., & Özbayoğlu, A. (2019). Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005-2019. In *arXiv (Cornell University)*. Technische Universitat Dresden. https://doi.org/10.48550/arxiv.1911.13288

[15] Simonov, J., & Gligorov, Z. (2021). Customs Revenues Prediction Using Ensemble Methods (Statistical Modelling vs Machine Learning). *World Customs Journal*, *15*(2). https://doi.org/10.55596/001c.116452

[16] Teräsvirta, T., Tjøstheim, D., & Granger, W. J. (2010). *Nonlinear models in economic theory* (pp. 16–27). Oxford University Pressoxford. https://doi.org/10.1093/acprof:oso/9780199587148.003.0002

[17] Thayyib, P. V., Thorakkattle, M. N., Usmani, F., Yahya, A. T., & Farhan, N. H. S. (2023). Forecasting Indian Goods and Services Tax revenue using TBATS, ETS, Neural Networks, and hybrid time series models. *Cogent Economics & Finance*, *11*(2). https://doi.org/10.1080/23322039.2023.2285649

[18] Verma, J., Agarwal, S., Pascanu, R., Mikolov, T., Bengio, Y., Greff, K., Srivastava, R., Koutnik, J., Steunebrink, B., Schmidhuber, J., Srivastava, S., Lessmann, S., Sagheer, A., Kotb, M., Bao, W., Yue, J., Rao, Y., Unnikrishnan, K., Venugopal, K., … Hellinckx, P. (2022). A Hybrid Deep Neural Network Model for Time Series Forecasting. *International Journal of Advanced Trends in Computer Science and Engineering*, *11*(1), 20–25. https://doi.org/10.30534/ijatcse/2022/051112022

[19] Zhu, L. (2022). Methodology and Application of Fiscal and Tax Forecasting Analysis Based on Multi-Source Big Data Fusion. *Mathematical Problems in Engineering*, *2022*, 1–12. https://doi.org/10.1155/2022/8028754