Predictive Asset Stewardship using AI & Data Analytics

Sahil Dalal[1], Vaishnavi Gomashe[2], Renuka Nival[3], Prof. Nikita Khawase[4]
[1,2,3,4] Department of Artificial Intelligence and Data Science of Engineering (AI&DS), ISBM College of Engineering, Pune-412115, Maharashtra, India

***Abstract*—Road infrastructure deterioration is a great problem in safety and transportation efficiency in large cities. Reactive maintenance approaches that are based on tradition have led to increased costs and inefficient resource allocation and ultimately result in more road accidents. In this paper, the authors discuss the technical implementation of the hybrid predictive maintenance framework that uses both Long Short-Term Memory (LSTM) Networks and Gradient Boosting Machines (GBM) to predict early road wear. The system, which uses real-time traffic and weather data, produces predictive insights that help city planners and maintenance teams to make informed decisions;**

A comprehensive hybrid LSTM-GBM model is the result of a fully-tuned hyper-parameter process. In this application, the results are verified with the help of RMSE and R-Squared (R^2) metrics in order to ensure the highest precision of prediction. The deployment of the model as a web-based application is beneficial as its municipality infrastructure system integrates easily. The dashboard also includes geospatial mapping, real-time alerts, and the trend analysis of periodic work on the road when there is wear and tear signs inescapable to the drivers.

The experiment findings unambiguously establish that the hybrid model outperforms the conventional predictive models produced using LSTM, which still has a sequential dependency, and GBM, which is further processed into predictions. Such a product will enhance road safety and reduce long-term maintenance costs while becoming a robust, scalable, and data-driven asset stewardship framework. The use of reinforcement learning, satellite imaging, and autonomous maintenance scheduling are the future prospects for the full optimization of predictive capabilities.

This research says that the groundbreaking implications of AI-based infrastructure management and the necessity of real-time data analytics in smarter urban planning are underlined. This result is major headway toward a more intelligent, safer, and ecologically sound road network assuring the life and activity of assets in a growing metropolitan area.

***Index Terms*—Predictive Maintenance, Road Wear Index, LSTM, Gradient Boosting Machine (GBM), Infrastructure Analytics, Real-Time Data Processing, Smart Cities, Asset Management, Flask API, Dashboard Visualization, Time-Series Forecasting, Traffic Analytics, Weather Impact Modeling, Hybrid AI Models, Data-Driven Decision Making.**

I. INTRODUCTION

Urban transportation systems are the lifeblood of the city's functionality, particularly since the road infrastructure ensures responsible, smooth, and continuous mobility. However, the increase in city size and traffic density has led to road surfaces being worn out and deterioratinsg faster than before. The conventional method of road maintenance, usually reactive in nature, is based on visual or citizens' observations and regularly causes a situation when the necessary services are not done in time. This results in greater costs to maintain, and the risk of road accidents is also raised.

These drawbacks at the very same time point to the necessity for an innovative, proactive, intelligent, and data-driven way of road asset management.

The development in AI, machine learning, and real-time data processing has created the possibilities for the predictive infrastructure maintenance. Municipalities can predict road conditions and plan maintenance activities in time by using new technologies like AI, ML, and real-time data. In this study, a novel approach to predictive maintenance that is a mix of LSTM time series modeling with GBM decision-making techniques was introduced, which was capable of learning the time series trends and making robust future predictions, simultaneously.

The proposed system utilizes high-dimensional, time-sensitive data that is gathered under weather conditions and traffic congestion levels such as vehicle count, speed variation, congestion levels, as well as rainfall, temperature and humidity to approximate a Road Wear Index (RWI). It is a project that is created through a well-established pipeline that includes data cleanup, feature selection, model training, and real-time deployment via Flask API. The interactive dashboard developed with Dash is a part of the key components and it is possible to see the trend of road wear, be notified instantly and map the road geospatially for decision making all through it.

Performance metrics such as Root Mean Square Error (RMSE) and R-Squared (R^2) are some of the indicators to test the effecriveness of the model. The results have indicated the hybrid LSTM-GBM model being not only more accurate but also more efficient and reliable than the single models, thus giving a better alternative for predictive maintenance. Also, the system implemented through a web-based platform has ensured that the system remains scalable and is easily integrated into the city's existing infrastructure system. This article underscores the impact of AI on infrastructure management and urban planning. The authors have demonstrated that a new intelligent and safer creative circle through a more sustainable urban road network can be set in motion with the unification of predictive analytics and real-time monitoring.

II. LITERATURE REVIEW

The application of Artificial Intelligence (AI) and Facts Analytics inside the domain of infrastructure apex has seen developing interest in both academics and industry. Since the expansion and age of city road networks increases, traditional reactive renewal technology - where interventions are created only after deteriorating or damage - you are incapable, expensive, expensive and regularly high to save injuries or disintegration. To deal with this, researchers are actively searching for future protecting fashion that can rely on road fall using ancient and real -time information input. In the future maintenance, the initial study depended primarily on the statistical and linear regression models, which was expected to decline the road depending on elements such as the age of the pavement, the volume of site visitors, and the facts of the weather. While being beneficial in the managed environment, these models were banned in their ability to generalize various avenue types, climate, and uses in conditions.

For example, GBMS has been shown to advance linear regression in prediction of maintenance requirements, considering the accumulated effects such as recurrent heavy truck traffic or sudden environmental changes. Parallel to the rise of traditional ML techniques, Deep Teaching Methods-It is designed for sequential or time-and-rear data-became very relevant in infrastructure monitoring. Long short-term memory (LSTM) networks, a type of recurrent neural network (RNN) have shown great promise in modelling the time-based pattern. They can learn from traffic flows, temperature variations and history of rainstorm to predict future deterioration trends.

Some studies have applied LSTM for traffic forecasts, weather forecasts and pavement performances modelling, which have promising results. However, single LSTM or GBM models come with each limit. LSTM is very effective for temporal data but can underperform when dealing with a material type or maintenance history and dealing with tabular features. On the contrary, GBMS is best with non-numeric inputs but effectively fails to consider the temporal mobility.

As a result, the models of the hybrids connecting both techniques emerged as a strong solution. These models generally feed the time-and-rear data to the LSTM and merge their output with the stable or engineered features processed by GBM, resulting in a more holistic understanding of the position of the wear. While some academic works have invented the Hybrid Models Dello, very few people have implemented them in real-time, deployable systems. Most study models stop on evaluation and do not transition to real-world applications.

This highlights at important research interval-while the future algorithm is mature, the operation of these insight into actionable devices such as real-time dashboard, alert and maintenance recommendations remain underdeveloped. In addition, many literatures focus on national highways or major urban centers, which neglects secondary or local roads that are equally important but are often under-monitors. There is also limited emphasis on route-level granularity, which is required for targeted maintenance scheduling and resource allocation. Additionally, very little studies address the effects of delay in maintenance of the integration of phenomena data or the acceleration of future road wear. Given these limitations, the purpose of our study is to bridge the difference between the future modelling and the implementation of the real world. We construct a full-stack solution and construct on the foundation laid by previous research, which not only forecasts the road wearing using the hybrid LSTM + GBM model, but also integrates the prediction engine in a real-time dashboard. It facilitates the dashboard route-tier monitoring, threshold-based alert and active maintenance recommendations-thus offering a full ecosystem for the future-staging asset stewardship.

III. PROBLEM STATEMENT

The integrity and longevity of the urban road infrastructure are central for the functioning of the transport network of any city. Roads become the backbone of economic activity, emergency services, public mobility and logistics. However, these important assets are conveyed to constant dynamic stresses such as traffic loads, diverse vehicle types, rigid environmental conditions, and dynamic stresses such as inconsistent maintenance practices. Over time, these factors contribute to gradual but compounding road fall. If not constantly addressed, this decline leads to an accident rate, congestion, low fuel efficiency and excessive repair costs. Despite the importance of timely road maintenance, many urban administrations continue to follow reactive maintenance protocols, where the repair work is started only after visual fall - such as pit or cracks - form. This reactive model is fundamentally flawed as it enhances the possibility of infrastructure failure, increases long -term maintenance costs, and leaves authorities over the minimum lead time to function. In addition, it does not provide any forecast capacity for maintenance scheme or resource optimization. From a technical point of view, several research studies and pilot implementation have detected the use of machine learning algorithms to predict infrastructure wear. However, most of these models are limited to the range. They are either trained on static dataset, ignore time-series dependence, or fail to include real-time updates. While focusing on some completely regression accuracy, without considering how the model will be deployed, will be interpreted, or used by the decision -makers.

Additionally, user interfaces or alert system deficiency provides these solutions non-carvable in real-world maintenance scheme workflows. Another significant limit in the current body of the work is silent treatment of data sources. Road fall is not powered by the same factor, but by complex interaction between many variables - such as vehicle density, average vehicle speed, type of road surface, physical aging, rainfall and final maintenance since time. Most future -stating framework either does not effectively combine these features or fails to model the temporary aspects of road use patterns. In addition, real -time data ingestion, analysis and visuals remain unspecified or absent in existing systems. Consequently, there is a significant difference in the deployment of intelligent, scalable and fully integrated platforms that provide both accurate future compliant analysis and operating dashboard for infrastructure plan teams. The purpose of this research is to address these important intervals and implement a comprehensive future maintenance structure which includes the following: Hybrid Predictive Modeling: LSTM (long-term short-term memory) to handle a novel combination time-series data and GBM (gradient boosting machine) to handle, to handle, categorized and static variables. This hybrid model is designed to capture both short-term fluctuations and long-term decline trends. Real-time data processing: Integration of streaming or hourly data input (traffic load, weather, maintenance log, event, etc.) to ensure that predictions reflect the current conditions rather than the static historical pattern.

Interactive dashboard systems: A front-end interface for stakeholders, designed to display the root-wise wear index, highlight the priority zone, set the threshold alert, and imagine performance over time-capable of taking pre-dated, pre-dated. Scalable and deployable architecture: A modular architecture that can be scaled in cities and can be compatible with various road types, materials, and environment, which makes it suitable for both local and regional authorities. In summary, the problem is not only the absence of a prediction model, but a deficiency of practically deployable, real-time, multi-factor and explanatory solution that empowers civic bodies to transfer active maintenance from reactive repair. This task creates such a solution, evaluate its performance hardly, and shows real -time integration and its protectiveness through visualization tools.

1. System Architecture

The architecture of the proposed future asset stewardship system is designed to enable the ingestion of multi-source data, using AI using hybrid predictive modeling, and provides actionable insight through an interactive dashboard. Architecture consists of six integrated layers: data acquisition layer, data processing layer, modeling layer, payment layer, alerting and visualization layer and feedback layer. Each layer plays an important role in ensuring road maintenance and public safety to ensure real -time decision making.

1.1 Data acquisition layer

This layer serves as the foundation of collecting both historical and real -time data. This collects data from the following major sources:

Traffic data: vehicle count, vehicle type (car, truck), and 11 important COEP flyover routes include average speed on the routes.

Weather data: Rainfall (mm) covers parameters such as temperature (° C), and humidity (%).

Road maintenance records: Final maintenance dates, content types, road age and road types provide information.

Report of the incident: Previous incidents and accidents on specific routes include calculations. The data is collected at an hour, daily and monthly intervals to catch both short-term fluctuations and long-term wear patterns.

1.2 data processing layer

This layer ensures the quality and readiness of data for analysis. It performs:

Data cleaning: removing discrepancies, handling missing values.

Feature Engineering: The construction of derivative features such as final maintenance time, seasonal tagging (eg, summer, rain), and root risk score.

Dataset partitions: Simulation data is used to generate different training (one year) and test dataset, ensuring realistic future prediction without data leakage.

1.3 Modeling layer

The system includes a hybrid model:

LSTM (long short -term memory): Capture temporary dependence and pattern in sequential traffic and weather data.

GBM (gradient boosting machine): Static or less gradual features such as material types, road age and route characteristics. These models are trained independently and then their output is fused in a post-modeling phase to generate a broad road wear index.

1.4 Prediction layer

This layer produces a predictive output:

The road wear index for each route produces daily and monthly.

Predictions are updated as new data flows to ensure real -time accuracy.

The system identifies the routes at risk of fall or potential accidents.

Feature Scaling: standardization using Z-score generalization to align input for both LSTM and GBM models.

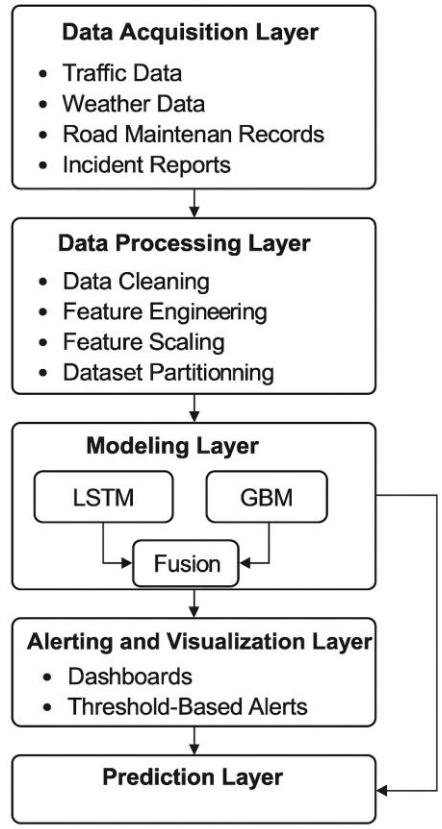


Figure1. System Architecture

1.5 Alerting and visualization layer

This layer through interfaces with end-use (city planners, maintenance teams):

Dashboard (developed using power BI and/or dash) that perform: Status of hour, daily and monthly route. Predicted the index of wearing across the routes. Priority, the route-wise priority using the severity score. Circumstances of risky areas, trends and maps.

Threshold-based alerts: automated alerts produce when the wear indexes cross pre-defined safety boundaries, which enable pre-refractions.

2. Data Collection

The foundation of our Predictive Road Wear Monitoring System is a strong and diverse dataset that simulates the situation of real -world traffic, weather and road maintenance. The data collection was planned to carefully cover several variables that greatly affect the road wearing with time. Dataset includes both real -time and historical data, which are classified into different dimensions.

2.1 Traffic Data:

This includes the number of vehicles per hour, which is classified by vehicle types such as cars, trucks and heavy vehicles. The dataset also records the speed of the average vehicle on each route. These parameters are important because vehicle load and speed patterns directly affect road surface erosion.

2.2Weather data:

Environmental factors such as rain (in mm), temperature (° C), and humidity (in %) were included to simulate the impact of climatic conditions on road longevity. Rainfall, especially, pairs with heavy vehicle movement increases the road wearing. Maintenance records: The dataset covers the last maintenance date for each route and the type of materials used for road construction. Additionally, road age (over years) is involved to assess the decline over time and assess its effect on the need for maintenance. incident report: The counting of the incident, including a minor and major road incidents, was tracked.

These serve as an indicator of the deteriorating road conditions, indirectly supporting the future model by correlating the situation of the bad road with high phenomenon rates. Route and time metadata: The data has been tagged with date and time values, and tagged with specific routes to enable time-series analysis and passage-wise predictable insights. A total of 11 different routes were considered, following both urban and semi-urban traffic patterns.

The dataset was mainly designed and managed using Excel, ensuring manual control over each simulation variable. For machine learning model training and evaluation, data was separated in training and testing sets. While the training dataset covered seasonal extreme (rainy and sunny months) in a year, the test dataset had a separate simulation in a different year for fair performance evaluation in the dataset.

This diverse and polymorphic dataset enables us to effectively train a hybrid LSTM + GBM model to predict the road wearing patterns and trigger proactive maintenance alerts, aimed at reducing the possibility of road accidents and adaptation of asset management.

3. Data Preprocessing

Effective machine learning model is an important step in preparing raw data collected for training. For this project, we followed a structured approach to clean, replace and format the dataset, ensuring that both LSTM and GBM models received the optimal input to learn.

3.1 Handling missing values:

The dataset was manually cured, but the simulated gaps were introduced to mimic the real -world landscapes where the sensor data could be unavailable. Further-fill and projected techniques were implemented to apply missing weather or traffic values where necessary.

3.2 Feature Engineering:

The derivative features were created to enrich the dataset and highlight complex relations. For example:

Vehicle load index = weighted score of vehicle count (gives overweight to trucks and buses).

Road stress score = combination of road age, recent maintenance and vehicle count.

Climate Effect Index = Humidity, rainfall and rapid score of ups and downs in temperature.

3.3 Encoding classified variables:

The route and material type columns were encoded in numerical formats. Label encoding was used for root ID (as they are gradual in terms of monitoring) and a-hot encoding was used to preserve the category of relationships for material types.

3.4 Time-series format for LSTM:

Since the LSTM model requires sequential data, the dataset was converted into a 3D format with dimensions [samples, timesteps, features]. The data of each route was converted into time windows per hour/daily variations to enable temporary pattern recognition by the LSTM model.

3.5 Feature Scaling:

To bring all the facilities into a uniform range and to speed up convergence during model training:

Z-score standardization (mean = 0, STD = 1) was applied to all continuous features (eg, speed, temperature, temperature).

This preprocessing pipeline enabled the manufacture of high quality, structured inputs for both models-LSTM and engineer to learn from engineer facilities and stable variables to capture the traffic-visitor pattern. Clean and converted data ensured high reliability and accuracy during model estimate and dashboard integration.

4. Feature Engineering

Facility Engineering plays a key role in increasing the influence of prediction models by creating new input variables that acquire important patterns in raw data. In this project, we have formed a hybrid facility set that combines the Domain J Knowledge, environmental conditions, traffic behavior and infrastructure to improve the forecasts of road wear on various routes.

4.1 Derived Traffic Features

To better understand the stress on the roads, we calculated the matrix that captures the effect of different vehicles:

Vehicle Load Index (VLI):

VLI = (Cars/Bikes ×1) + (Buses/Trucks ×3)

These indexes weigh more significantly to heavy vehicles, as they contribute more to the degeneration of the road.

Average vehicle density:

The number of vehicles per normal hour by the length of the road (if available or assumed uniforms) is useful for comparing the intensity of high vs traffic.

4.2 Weather -based stress features

Weather conditions can accelerate road damage, especially in areas with tropical or monsoon -filled areas like Pune. We Engineered:

Climate effect score: Combined score obtained from temperature variations, rainfall intensity and humidity levels.

Rain deviation: Steps to how daily rainfall is different from the long -term average of the route to flag the unusual climatic events that can affect the integrity of the road.

4.3 Maintenance -driven Predictors

Historical Maintenance Records was transformed into quantifiable inputs:

Time after last maintenance (days): Originated from the current date and the last maintenance date of recorded.

Road Age Factor: Helps to find deteriorating roads that have not been serviced recently.

Road Age Factor= Age of Road (Years)/ Time Since Last Maintenance

4.4 Route-level and specific features

Encoded Route ID:

Each route is assigned a unique ID that is label-encoded to track the route-specific pattern. Material flexibility rating (MR):

The material was assigned the durability score (eg, concrete = 0.9, asphalt = 0.7) to estimate the resistance of wear over time.

4.5 Incident correlation features

Event frequency rate:

Incidents of 1000 vehicles per 1000 - are used to connect road conditions with the possibility of accident.

Event wearing score:

Counting of the event and the age of the road to prioritize risky areas.

5. Model Selection and Justification

The main objective of this project was to develop a highly accurate, time-comprehensive future system to assess the road wear index in many routes. Given the complexity of the data, a mixture of instant (time-series) and classified/stable features-a hybrid model approach was adopted to take the best advantage of many algorithms' strength.

5.1 Hybrid Model Architecture: LSTM + Gradient Boosting Machine (GBM)

To address the specific nature of a dataset, we designed a hybrid model that adds:

Long short-term memory (LSTM) to capture temporary dependence in traffic and weather conditions over time.

Gradient Boosting Machine (GBM) for modeling such as road types, materials, phenomena calculations and maintenance history.

5.2 Justification for model options

LSTM (Long short -term memory)

LSTM is a type of recurrent neural network (RNN) known for learning long-term dependence in time-series data.

Over time, vehicle flow, rainfall and ideal for capturing trends in road fall.

Able to manage the missing shield problem, it is well suited to sequences such as per hour or daily comments in months.

GBM (Gradient Boosting Machine)

GBM is a powerful attire method that produces models in a stage-wise fashion by reducing predicted errors.

Effective in handling asymmetrical tabular data (eg, road type, maintenance history, materials).

In cases, Excel where feature interactions and non-lectured relations play an important role-which is a matter of deteriorating road due to complex interaction between traffic, weather and infrastructure.

5.3 Integration strategy

LSTM processes Temporal features to the first (eg, historical traffic and weather conditions) and produces a learned representation.

This output is included with static or relevant features (eg, material type, final maintenance date) and passed to the GBM model.

This hybrid stacking architecture ensures that both time-elastic patterns and non-time-free indicators are effectively captured.

5.4 Benefits of hybrid model

Better accuracy: Each model compensates for other boundaries.

Normally: Better performance in different routes with different conditions.

Lecturer: GBM feature allows for importance extraction, while the LSTM model traffic-weather mobility.

5.5 Alternative model considered

Arima and Prophet were considered, but there was a lack of ability to include complex multi-commercial features.

Standalone LSTM or GBM model was tested, but inferior results were produced compared to hybrid design.

6. Model Development

The development of the predictive model was a multi-phase process involving the implementation, training, and fine-tuning of a hybrid LSTM + GBM model. The model was designed to utilize both temporal sequences and non-temporal contextual features to predict the Road Wear Index across 11 different routes, using historical data from traffic, weather, and maintenance records.

6.1 Input Design and Data Structuring

The input dataset consisted of 29 engineered features, including:

Time-series features: vehicle count, cars, trucks, average speed, rainfall (mm), temperature (°C), humidity (%)

Static/contextual features: route ID, road type, material type, incident count, last maintenance date, age of road

Data was reshaped for LSTM input as 3D arrays with shape (samples, time steps, features).

6.2 LSTM Model Construction

Purpose: To capture trends and temporal dependencies in traffic and weather patterns.

Configuration:

Input Layer: Time-series features

Hidden Layers: 2 LSTM layers with dropout

Output Layer: Dense layer producing intermediate features

Hyperparameters:

Epochs: 50

Batch Size: 64

Optimizer: Adam

Loss: MSE (Mean Squared Error)

The LSTM model generated sequence-based embeddings, which were used as inputs for the second-stage GBM model.

6.3 GBM Model Construction

Purpose: To learn relationships between static features and the target (Road Wear Index).

Algorithm: XGBoost implementation of Gradient Boosting.

Features used: Output from LSTM + contextual features like road type, incident count, material type, etc.

Hyperparameters:

Learning rate: 0.1

Max depth: 3

Estimators: 100

Output: Final predicted Road Wear Index for each route and time period.

6.4 Training Strategy

The model was trained on 12 months of simulated data, including rainy (August) and sunny (May) seasons to ensure diverse conditions.

Separate files were used for training and testing data to avoid leakage and maintain real-time forecasting accuracy.

Model performance was monitored using metrics like:

Mean Absolute Error (MAE)

Root Mean Squared Error (RMSE)

R^2 Score (for variance explanation)

6.5 Model Checkpoints and Logging

Checkpoints were used during LSTM training to save best weights.

Logs were maintained for:

Training vs Validation loss

Performance comparisons of different hyperparameter configurations

Predictions vs Actuals for testing dataset

This development phase concluded with a fully trained hybrid model

7. Model Training and Hyperparameter Tuning

To ensure the strong and accurate prediction of the road wear, we developed two models and performed fine tune: Gradient Boosting Machine (GBM) and a long short-term memory (LSTM) neural network. Each model was trained individually, followed by a hybrid approach that combined their strength.

7.1 Data Preparation for Model Training

Prior to training, data passes through pre -processing stages to ensure stability and model compatibility. This includes:

Cleaning column names and forming data types

Encoding the classified features using label encoding.

Scale using min-max generalization.

Model Evaluation and Performance Metrics

Feature-target separation was then performed by isolating the Wear_Index column as the target variable, while the remaining columns were used as input features.

```
from sklearn.model_selection import train_test_split
X = train_data.drop('Wear_Index', axis=1)
y = train_data['Wear_Index']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

7.2 Gradient Boosting Machine (GBM) Model

The GBM model decision is a powerful dress method based on trees. This is conducted by the training model sequentially to fix the errors of the previous ones. It is particularly effective for structured data and is capable of capturing non-lectured patterns without the need of large versions of data.

Key hyperparameters that were tuned include:

n_estimators: Number of boosting stages (trees).

learning_rate: Controls the contribution of each tree.

max_depth: Maximum depth of individual trees.

Model training:

```
from sklearn.ensemble import GradientBoostingRegressor
gbm_model = GradientBoostingRegressor(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=3,
    random_state=42
)
gbm_model.fit(X_train, y_train)
```

Hyperparameter tuning was carried out manually and iteratively to avoid overfitting and achieve a balance between bias and variance.

7.3 LSTM Neural Network

LSTM is a type of recurrent nervous network (RNN) that is favorable for sequence predicting problems. In our case, LSTM was used to modeling temporary dependence between traffic flows, weather conditions and road material lifespan, which develop over time.

Before feeding data into the LSTM, it was reshaped into a 3D structure as required: [samples, time_steps, features].

```
import numpy as np
X_lstm = X.values.astype(np.float32).reshape(X.shape[0], 1, X.shape[1])
```

The architecture of the LSTM model included:

One LSTM layer with 100 units to capture sequential patterns.

A Dropout layer to prevent overfitting.

A Dense layer to produce the final prediction.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
stm_model = Sequential([
    LSTM(100, input_shape=(X_lstm.shape[1], X_lstm.shape[2])),
    Dropout(0.2),
    Dense(1)
])
lstm_model.compile(optimizer='adam', loss='mean_squared_error')
lstm_model.fit(X_lstm, y, epochs=50, batch_size=64)
```

Epochs and batch size were chosen through trial runs to ensure convergence without significant overfitting or underfitting.

7.4 Hybrid Model: Combining GBM and LSTM

To take advantage of the strength of both models, we created a hybrid dress by combining predictions from GBM and LSTM. The purpose of this approach is to balance:

Explain and accuracy of GBM with LSTM sequence modeling capabilities. Two strategies were considered:

Average: Simple average of predictions from both models.

Weighted fusion: Giving different weight to each model depending on their credibility.

The ensemble method improves prediction robustness and can help mitigate the limitations of individual models.

```
# Example: Weighted Ensemble
final_prediction = 0.6 * lstm_preds + 0.4 * gbm_preds
```

Weight values were empirically adjusted based on validation performance.

7.5 Saving Models for Deployment

To support deployment in a real-time dashboard, all trained models and preprocessing tools (label encoder, scaler) were serialized and saved:

```
import joblib
lstm_model.save('road_wear_lstm_model.h5')
joblib.dump(gbm_model,
'road_wear_gbm_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
joblib.dump(le, 'label_encoder.pkl')
```

This ensured smooth integration with frontend systems for live road wear prediction and alert generation.

8. Model Evaluation and Performance Metrics

To evaluate the effectiveness of our future stating structure for forecasting the road wear index, we adopted a hybrid modeling approach by mixing long short -term memory (LSTM) network and gradient boosting machines (GBM). This section data is detailed on preprocessing, model training, individual model performance and joint hybrid model assessment.

8.1 Data Loading and Preprocessing

The merged dataset was first cleaned to address column formatting issues. In column names, additional spaces (eg, 'Incident_Count') and Missed header (eg, 'average_temperature ((° c) _x') were corrected.

All relevant numerical columns-traffic counts, vehicle weight, environmental factors, and infrastructure characteristics, including-were competent with forcibly enabled to handle-nutritious values with Pd.to_numeric, enabled to handle non-numeric values.

8.2 Feature Normalization

We applied MinMaxScaler to normalize all continuous features in range [0, 1]. This ensured optimal training convergence, especially for LSTM models, which is sensitive to the magnitude of input. The same scaling was continuously implemented in training and test dataset.

8.3 Train-Test splitting

The target variable Wear_index was separated from input features. Using 80–20 train-test split (randam_state = 42), we ensured reproducibility in all model assessments. This partition strategy was equally used for LSTM, GBM and hybrid models.

8.4 Individual Model Architectures

LSTM Model

The LSTM model was designed using the Keras API with the following structure:

Input Shape: Reshaped to (samples, 1, features)

LSTM Layer: 100 units

Dropout Layer: 20% dropout

Dense Layer: 1 neuron for regression output

It was compiled using the Adam optimizer and trained on mean_squared_error for 50 epochs with a batch size of 64. The trained model was saved to road_wear_lstm_model.h5.

GBM Model

We used GradientBoostingRegressor with:

n_estimators = 100

learning_rate = 0.1

max_depth = 3

This model was trained on the same training set and evaluated using the identical test set.

8.5 Hybrid Model Strategy

The hybrid model was designed to leverage the strengths of both LSTM and GBM. The final prediction was computed by averaging the outputs of both models using simple arithmetic mean:

Hybrid Prediction=1/2(LSTM Prediction+ GBM Prediction)

8.6 Evaluation Metrics

Mean Absolute Error (MAE) was selected as the primary metric, given its interpretability in regression contexts.

Model	Mean Absolute Error (MAE)
LSTM	545.79
GBM	8.17
Hybrid	7.94

LSTM showed high error due to its complexity and the tabular nature of the data.

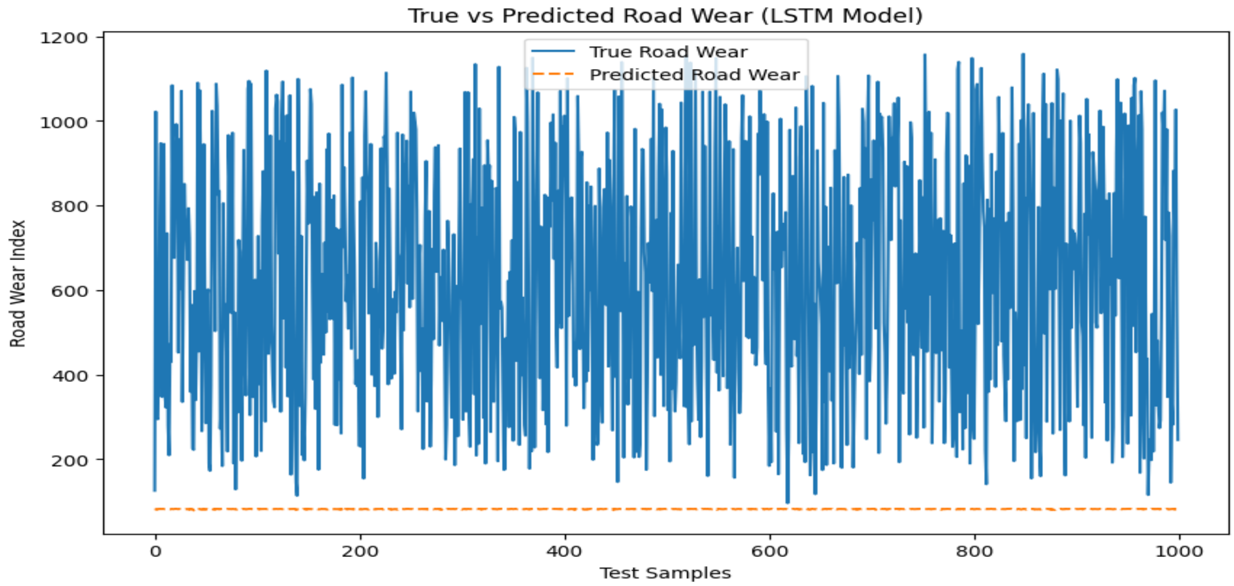
GBM demonstrated strong performance with a low MAE of 8.17.

Hybrid Model slightly outperformed GBM, achieving the lowest MAE of 7.94, indicating a more robust prediction by combining deep learning and ensemble methods.

8.7 Visualization

A comparative visualization was plotted using matplotlib to compare actual vs. predicted road wear indices across models. The hybrid predictions followed the actual values more closely than either individual model, with fewer large deviations.

The results of the assessment strongly suggests that while GBM is highly effective on structured tabular data, its forecast can be further enhanced when combined with deep learning models such as LSTM. The hybrid model's lowest MAE indicates its potential for real-world deployment in predictive road maintenance systems, offering accurate and reliable forecasts.



9. Model Deployment

Successful training and evaluation of hybrid models required a efficient deployment mechanism to serve predictive in real time and integrated with downstream systems such as dashboard, alert mechanisms or city maintenance platforms. For this purpose, we employed an API for the service of model predictions on HTTP, employed a mild python web framework, flask.

9.1 Deployment Objectives

The deployment strategy was designed with the following goals:

Real time prediction: Enable the prediction of wearing roads on real -time data.

System integration: Allow integration with web-based dashboards, alerting systems and other applications, which through restful API &Points.

Modular Design: Provide flexibility to update the model without re -design API.

Scalability and Maintainability: Design a production-friendly and easily expandable codebase for future model versions.

9.2 API Architecture and Workflow

The flask API was designed to load both LSTM and GBM models, prepares upcoming input data, predict hybrid, and return the output in a structured JSON format. Architectural Workflow is mentioned below:

9.2.1 Model Loading:

The GBM model was loaded using jblib.load ('GBM_MODEL.PKL').

The LSTM model was loaded using keras.Models.load_Model ('Road_Wear_LSTM_MODEL.H5').

Minmaxscaler used during training was also loaded to ensure frequent convenience scaling.

9.2.2 Request handling:

The prediction route accepts post requests with new route-specific input data in JSON format.

Example input: traffic count, weather details, road age, date of maintenance, etc.

9.2.3 Data Preprocessing:

Input JSON converts to a dataframe.

Numeric typecasting is applied.

Feature normalization is done using a pre -fitted MinMaxScaler.

The LSTM input is reshaped to 3D as required ((samples, timesteps, features)).

9.2.4 Prediction generation:

Predictions are obtained from both models:

lstm_pred = lstm_Model.predict (Reshaped_Data)

GBM_PRED = GBM_MODEL.Predict (scled_data)
Hybrid_pred = (lstm_pred.flatten () + gbm_pred) / 2
The final hybrid prediction is returned as an API response.
9.2.5 API Response:
The result is back in JSON format:
9.3 Integration with dashboard and alerting system
Flask API & endPoint was integrated with a dashboard interface (eg, power BI, streamlight, or dash) that imagines:
Real-time wear index per route
Historical vs. forecast comparison
Alert for road classes near Maintenance Threshold
The API was also configured to trigger trigger notifications (email or SMS) when predicted road wear index exceeded a significant range, ensuring proactive infrastructure management.
9.4 Deployment environment
Server: Localhost for development; Deployment for cloud hosting-Taiyar (eg, AWS EC2 or Heroku)
Port: API serves Port 5000
Security: CORS capable of dashboard communication; Production certification includes certification and HTTPS.
The flask-based deployment pipeline successfully eliminated the gap between model growth and operating utility. By highlighting models' predictions through a restful API, the system enables real -time, scalable and spontaneous integration with visualization platforms and maintenance decision systems. This deployment strategy gives an example of the practical utility of the machine learning model in the smart city infrastructure monitoring and the prepaid asset management.

10. Dashboard Development and Visualization
The dashboard hybrid is an important component of the predictive maintenance system, serving as a visual and analytical layer that bridges the difference between the future -fashioned model output and real -world decision making. This has been developed not only to display the Road Wear Index (RWI) values, but also to provide a broad, real -time interface for road maintenance officers to monitor and react to road conditions in various routes in the city. This dashboard brings several data currents together - including weather, traffic, and road wear predictions - in a single consistent platform that aids active maintenance plan and resource allocation. The development of the dashboard was directed by two main objectives: first, to provide a spontaneous and accessible interface to technical and non-technical stakeholders; And secondly, to ensure that models from LSTM and GBM models turn out to output actionable insights that can support timely maintenance tasks. Using HTML, CSS (with tailwind framework), and using a JavaScript for frontend rendering, the dashboard interfaces with a flask-based backend that handles API communication and serves the real-time output of the prediction model. The dynamic capabilities of the JavaScript were used to bring new predictions from the backend and update various visual components on the interface. The core Road Wear Index (RWI) of the dashboard has a real -time performance. This index, which is predicted by the hybrid model, is dynamically updated and is shown prominently on the dashboard.

RWI severity is color-coded based on threshold: for example, the green indicates normal conditions, yellow suggestions to wear medium, and the red indicates the level of important wear to which demands immediate attention. This allows color-coded visual cue users to immediately interpret the road status without the need for depth technical knowledge.
In addition to the RWI display, the dashboard covers root-specific analytics. Users can select a special passage from the dropdown menu, on which the dashboard updates all relevant charts and matrix for the specific passage. This functionality is essential for road network management, where each route may have separate traffic intensity, construction materials or environmental risk, and thus may have a unique wear pattern.
To provide a temporary reference to predictions, the dashboard contains time-series charts that conspire various time limitations, such as per hour, daily or weekly intervals such as RWI values. These charts help identify the trends of wearing and detect a sudden decline period, allowing engineers to estimate issues instead of reacting to them after the damage. Additionally, another chart current day wear level versus presents a side-by-side comparison of the previous day's wear, offering a direct view of whether the road conditions are improving or deteriorating.
Complementing road wear data, the dashboard integrates live weather and traffic information, including temperature, humidity, rainfall levels, vehicles count and types of vehicles (car, trucks, etc.).

These matrics are displayed in the summary card and updated in real time. Their presence is not only informative; They are involved to show the relationship between external conditions and the rate of road deterioration. For example, the increase in truck volume or rainfall may be reflected in an uptick in the immediate wear index.

Another major element of dashboard is maintenance suggestion module. The feature analyzes an estimated RWI against predetermined security and demonstration threshold. When the wear index crosses a certain range-for example, the system above 7 on the 10-point scale automatically produces a maintenance alert. In alerts are shown visually on the dashboard and designed to draw attention through color changes or blinking animations. In addition, the routes with significant wear conditions are highlighted automatically, making it easier for road engineers to prefer maintenance scheduling.

The visualization capabilities of the dashboard are mainly operated by Chart.JS, a powerful and flexible JavaScript charting library. It enables highly interactive and responsible graphs, which can easily handle the data updates continuously without the requirement of page reload. Behind the curtain, the dashboard requests the prediction data updated from the flask API from time to time, which provides results from the hybrid LSTM + GBM model. This setup ensures a spontaneous and low-lonely flow of information from data input to actionable insight.

The set of this dashboard is not only its visual design, but also its functional depth. This does not only display static predictions; Instead, it continuously interacts with live input, adopts visualization based on the path and time, and automates the alert system to reduce human dependence. It has been structured to ensure that real -time model outputs can directly support the decision -making workflows of city planners, engineers and road maintenance teams.

Finally, the dashboard plays an important role in translating a complex output of machine learning models into an operating tool for maintenance of infrastructure. The integration of real -time season and traffic data with future stating analytics enables it to act as a wise monitoring system. This system does not only show the current situation of road wear - it empowers users to predict the probable failures, plan intervention and eventually reduce the risk of accidents to facilitate timely repair. This is a good example of how AI and data visualization, when effectively brought together, can be smart, safe and more cost-skilled urban structure management.

11. Challenges Faced and Mitigation Strategies

Throughout the development of the forecasting wear monitoring system and its visualization dashboard, many technical and operational challenges faced. One of the primary issues of the weather API's real-time data and the primary issues of integrating into the dashboard without delaying simulated traffic feeds. Real-time systems demand a high level of response, and can mislead any leg users in bringing or updating data or disrupt the forecast flow. To reduce this, asynchronous programming techniques were applied using Javascript's async/await mechanism, allowing background data when the user responded. Loading indicators and placeholder values were presented to handle any temporary data recovery delay.

Another major challenge was to synchronize the forecast output from the hybrid LSTM + GBM model, with continuous updating real-time input data. The displayed Road VIAR Index (RWI) reflected the latest traffic, weather and maintenance inputs. All of these incoming data were accomplished by ensuring compatible timestamp configuration in streams and model input pipelines. Flask API was always modified to return predictions corresponding to recent data records, thus ensuring consistent and real-time visualization updates.

The design of a dashboard with user -friendly yet information created another problem. The goal was to display various parameters such as temperature, humidity, traffic volume, RWI forecast and warning level, within a single interface without the user. To address this, a modular and grid-based layout was adopted using Tailwind CSS. Components such as compressed cards, live-updating graphs and dropdown menus were still applied to separate information from the hierarchy to maintain accession. The design approach also created a dashboard scalable, which allowed the adding new features to the full interface without redesigning.

The definition of accurate threshold for warning was also a challenge. There is no standard scale for the intensity of RWI, so carefully analyzing how to raise alerts or indicate maintenance intervention is necessary. This issue was reduced by studying Historical RWI distributions and applying a percentile-based threshold-20% of the values of the top as complex wear. These thresholds were validated using past maintenance records to ensure that they correspond to the views of real-world road waste.

In addition, testing the dashboard under different combinations of environmental and traffic conditions caused a combined explosion of the test cases. It was not possible to imitate each view directly from real-time feeds. Therefore, synthetic data generation techniques were working to make test cases to represent extreme and edge views such as high rainfall, or high humidity and old road temperatures. The cases of these tests were then executed by the dashboard to test the strength and flexibility of both model and visualization levels.

The team also had to deal with the missing or corrupt data of live API, especially during network fluctuations or when some dimensions of the weather were temporarily unavailable. This poses risks for the predictions and forecasts of both models. To handle this, strong data validation methods were made in both the backend flask API and the front dashboard. When detecting incomplete data .In the system warning the user through error messages, the default transfers to the default lt values or temporarily disabled components, preventing a crash or misleading display.

Finally, the issues of deployment-related consistency came up when integrating various components in machines and atmosphere. Different differences in the Python environments for the model, Node.js versions and the issues of communication between the flask API and the priority of Javascript created many obstacles. These dockers were resolved by containerization, where each ingredient was included with its dependence. This ensures consistency and fertility in the development, testing and phase of production.

IV. RESULTS AND DISCUSSION

The developed hybrid predictive maintenance system was evaluated through an interactive dashboard that imagines real -time and historical data, which enables the road wearing and detailed analysis of related parameters. This section key discusses the insight obtained from dashboard components and visual analytics. Road wear status component provides a quick visual signal of the current wearing index on various road segment. It is dynamically updated depending on traffic and weather input, and it acts as an important real -time metric for decision manufacturers. As a combination with it, the weather conditions collect the section temperature, humidity and freeze level data, which are the main features affecting the pavement decline. These indicators help to predict potential risks in the near future.

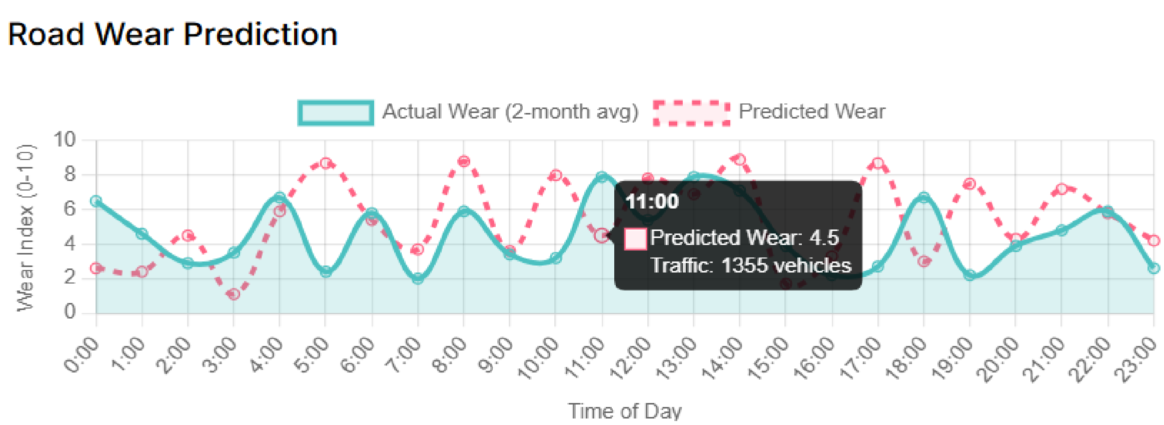


Figure 1. Road Wear Prediction over Time of Day.

The traffic volume panel imagines the load of vehicles in real time, a major contributor for structural road wearing. For this, the complement, traffic composition breaks the type of chart vehicles (eg, car, truck, buses), which enables the analysis of the vehicle categories, contributes to the most deteriorating.

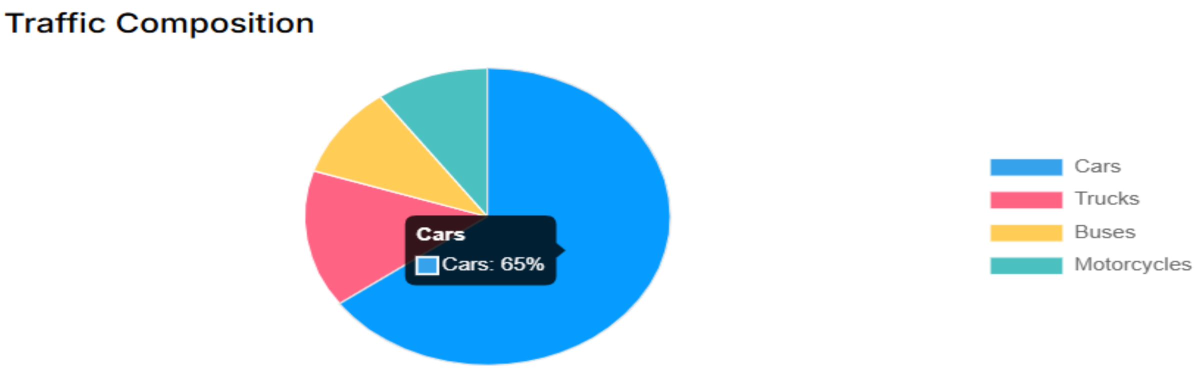


Figure 2. Real-Time Traffic Volume and Composition.

A speed distribution bar chart further enhances situational awareness, which shows the number of vehicles working within various speed brackets. The flow of high -speed vehicles combined with load concentration is often correlated with increased wear, and this chart validate such conversations.

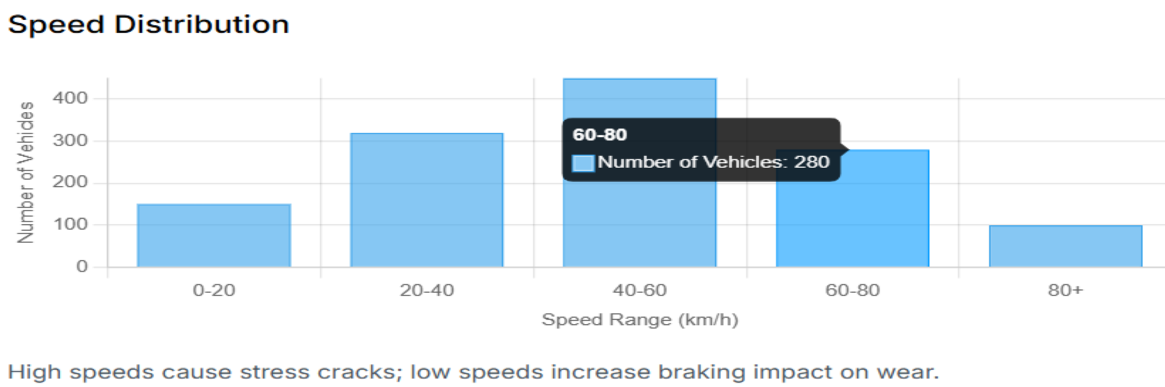


Figure 3. Impact of speed of vehicles on Road wear Index

A particularly valuable analysis is shown through the effect of weather on the graph of road wear, which presents the effect of separate environmental conditions (temperature, humidity, freeze events) on the wearing index. This confirms the hypothesis that environmental factors increase road damage beyond traffic-inspired stress.

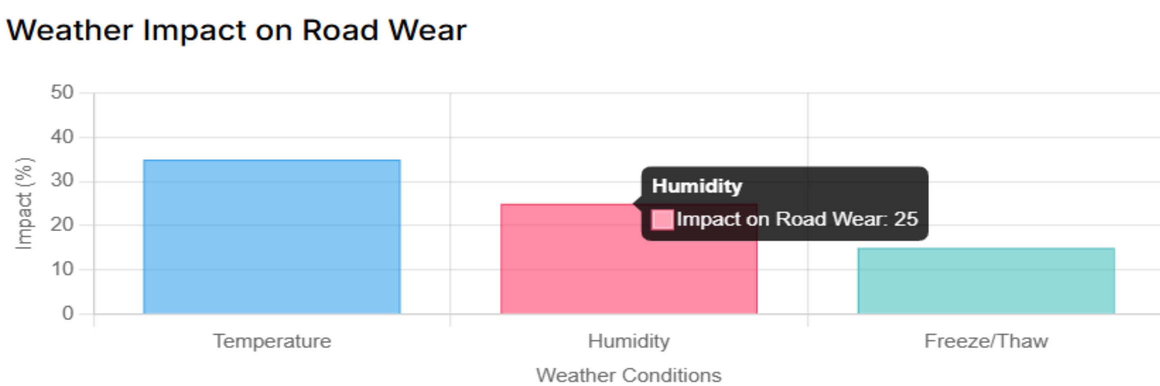


Figure 4. Impact of Temperature, Humidity, and Freeze on Road Wear.

The forecast area chart of one per hour predicts changes in the index of wear in the next 24 hours. This allows preemptive maintenance scheduling to avoid proper forecast growth. The system also involves active alerts with sound notification, when it is more than the significant threshold to wear, it is automatically triggered. These alerts appear on the dashboard and serve as a real -time warning for operators and field staff.

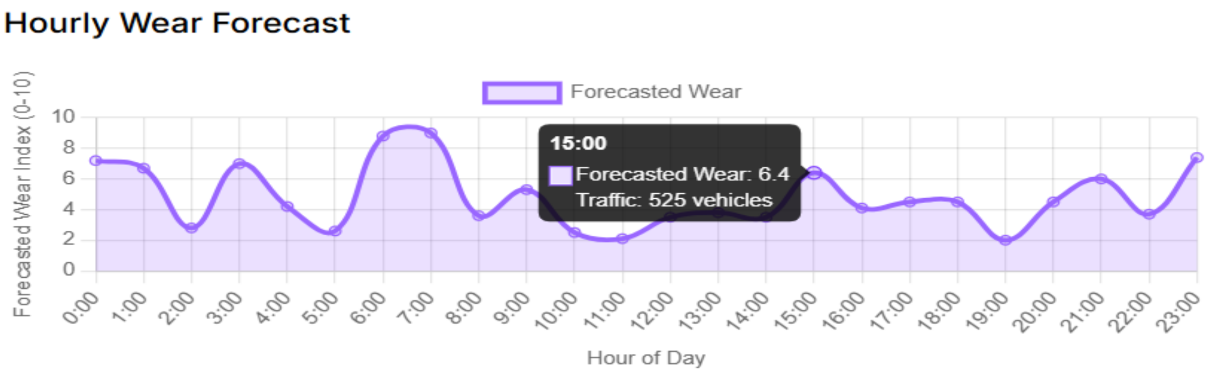


Figure 5. Hourly Forecast of Road Wear Index.

The decision of maintenance is further supported by a maintenance history line chart, which imagines the progression of the index of wear in months. Users can examine previous interventions and correlated them with post-delivery results. Additionally, the root timeline chart presents a longitudinal view of the wearing index in several routes (eg, root a, b, and c), which enables comparison of the route-by-way.

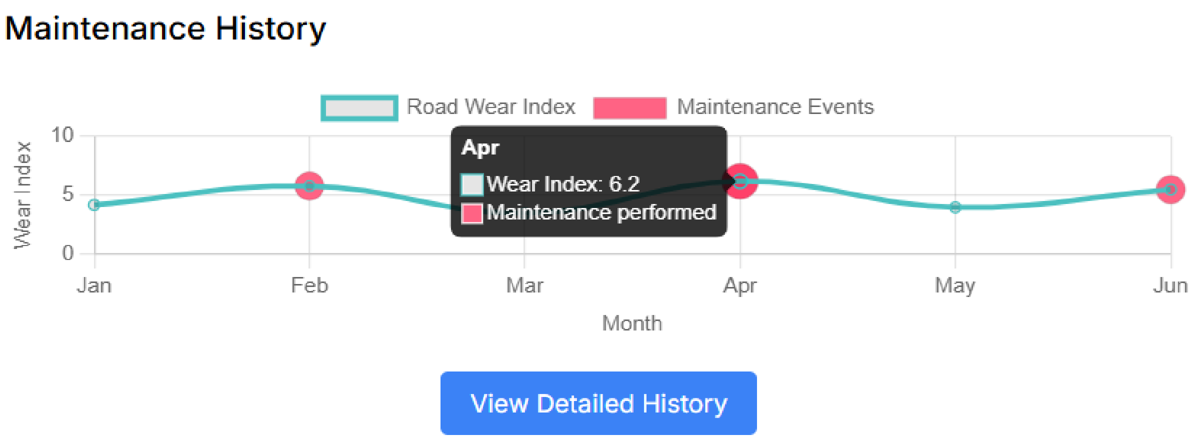


Figure 6. Maintenance History of the road

A historical RWI repository allows users to analyze the historic road wear index by both date and route. This time-series insight model contributes to the detection of retrenching and seasonal patterns.

An important feature landscape is a Scenario Forecasting Module, where users can simulate future conditions (eg, traffic surge, rain duration) to wear an estimated road under imaginary landscapes. This adds flexibility to the future flexibility in the dashboard. Finally, data filter users strengthen users to select specific conditions (eg, route, date, vehicle type), ensuring that the analysis is targeted and the decision is evidence-operated.

V. CONCLUSION AND FUTURE SCOPE

This research presented a hybrid -predictive maintenance structure for the purpose of monitoring road infrastructure using advanced AI techniques. By combining long short -term memory (LSTM) network for temporary pattern recognition with gradient boosting machines (GBM) for feature importance and strong predictions, the system effectively predicts the road wear index (RWI) in several urban routes. The integration of traffic patterns, weather conditions, road content and maintenance history enable a comprehensive model that goes beyond traditional reactive approaches. The developed dashboard not only imagines real -time predictions, but also facilitates actionable insights through dynamic alerts and comparative wear.

Results suggest that the proposed hybrid model may make accurate prediction of road wear trends and identify high -risk routes that require immediate or scheduled maintenance. Through real -time visual representation and alert mechanisms, the dashboard enhances decision making for urban planners and maintenance teams. This change in reactive -to -future asset stewardship ensures timely intervention, better allocation of resources, reduction in cost of maintenance and eventually, safe road conditions for passengers. The project also shows how AI and data analytics can practically be applied in civil infrastructure projects, which paves the route to intelligent cities.

Looking forward, the framework has a significant potential for growth. The direction of an immediate future is the integration of real -time sensor data, such as vibration, load pressure, or imagination from traffic cameras to further improve the prediction accuracy.

Scalability of the system for the entire city's road networks can enable wide deployment and large -scale benefits. Introduction to adaptive threshold through learning reinforcement can dynamically optimize when and where the alerts are triggers, make the system more intelligent over time. Additionally, incorporating cost-based priority in the maintenance scheme will allow authorities to align the road repair with lack of budget and strategic goals.

There is also scope to develop a mobile version of the dashboard to support field teams with real-time data and task allocation. Including seasonal factors, construction activities and long -term urban planning elements can increase the depth and foresight of predictions. Overall, this research lays a foundation for a smart, AI-powered infrastructure management solution that not only predicts the deteriorating road, but also guides continuous intervention, which is maintained by the city roads safe, more durable and more wisely.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” The Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[3] K. Zhang, C. Qiao, and Y. Ding, “Data-Driven Pavement Maintenance Decision-Making Using Machine Learning Models,” Journal of Infrastructure Systems, vol. 26, no. 3, p. 04020029, 2020.

[4] J. Li and H. Cai, “A Review of Machine Learning Applications in Predictive Maintenance for Infrastructure,” Automation in Construction, vol. 107, p. 102941, 2019.

[5] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2016, pp. 785–794.

[6] J. Brownlee, Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery, 2018.

[7] F. Chollet, “Keras: Deep Learning Library for Theano and TensorFlow,” 2015. [Online]. Available: https://keras.io/

[8] Plotly, “Dash: Build Analytical Web Apps in Python,” 2023. [Online]. Available: https://dash.plotly.com/

[9] Microsoft, “Power BI Documentation,” 2022. [Online]. Available: https://learn.microsoft.com/en-us/power-bi/

[10] OpenWeatherMap, “Weather API,” 2023. [Online]. Available: https://openweathermap.org/api