COLORADO
Department of Transportation

CDOT Office of Applied Research

i-BM: An Intelligent Bridge Management Tool for Bridge and Culvert Deterioration Forecasting and Anomaly Detection based on Physics-Guided Deep Learning

Farnoush Banaei-Kashani, Kevin Rens

February 2025

CDOT-2025-02

Technical Report Documentation Page

1. Report No.
CDOT-2025-02

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
i-BM: An Intelligent Bridge Management Tool for Bridge and Culvert Deterioration Forecasting and Anomaly Detection based on Physics-Guided Deep Learning

5. Report Date
February 2025

6. Performing Organization Code

7. Author(s)
Farnoush Banaei-Kashani, Kevin Rens

8. Performing Organization Report No.

9. Performing Organization Name and Address
University of Colorado Denver
1201 Larimer St, Denver, CO 80204

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
Colorado Department of Transportation
Office of Applied Research
2829 W. Howard Place
Denver, CO 80204

13. Type of Report and Period Covered
Final

14. Sponsoring Agency Code

15. Supplementary Notes
Prepared in cooperation with the US Department of Transportation, Federal Highway Administration

16. Abstract
Bridges and culverts deteriorate with time and use. The deterioration process is affected by several factors, such as structural materials, structural design and behavior, daily traffic, freeze and thaw cycles, climate, pollution, temperature variation. After a certain period of time has elapsed, the deterioration processes accelerate and in a relatively short time interval the components can lose the capacity to carry the loads they were designed to support. While in the past, various data-driven deterioration models, including Bayesian models, probit model, and Markov chains are proposed in the literature to model bridge deterioration, these models either suffer from low accuracy or are too complex to be applicable. In the past we have developed AI-based deep learning (DL) models for enhanced bridge management. In particular, we focused on developing DL-based models for bridge subtyping (or bridge family generation) and bridge (and culvert) deterioration forecasting. Our results show that DL-based models for bridge subtyping and bridge/culvert deterioration outperform other existing models and can be used to effectively enhance bridge management.

With this problem definition, we develop an intelligent bridge management tool (dubbed, i-BM, short for intelligent Bridge Management) for bridge and culvert deterioration forecasting and anomaly detection to be used by CDOT bridge engineers as a tool for effective bridge management. i-BM builds on and significantly extends our aforementioned prior work on DL models for bridge management in three ways: 1) integrating the DL models into a user-friendly software tool with graphical user interface and enhanced operational features, 2) developing enhanced physics-guided DL models that integrate traditional physics based bridge deterioration forecasting models with data-driven DL models for further improved performance in prediction of deterioration, and 3) introducing bridge performance anomaly detection as a new capability that allows for accurate prediction of bridge performance anomalies such as those that can lead to bridge failures/accidents.

17. Keywords
Bridge/Culvert Deterioration Forecast, Bridge/Culvert Anomaly Detection, Deep Learning Model, Data-Driven, Physics-guided Deep Learning Model

18. Distribution Statement
This document is available on CDOT's website:
https://www.codot.gov/programs/research

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages

22. Price

Reproduction of completed page authorized

The contents of this report reflect the views of the author(s), who is(are) responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views of the Colorado Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

Acknowledgements

We gratefully acknowledge the support and guidance of the CDOT Research Project Manager, Study Manager, Thien Tran (CDOT Office of Applied Research), and the Study Panel: Jessica Martinez, (CDOT Bridge - Co- Champion); Natasha Butler (CDOT Bridge - Co- Champion); Lynn Crosswell (CDOT Bridger - Co-Champion); Michael Collins, (CDOT Bridge - Sponsor). It has been a privilege to work with these collaborative partners on such an impactful and rewarding project. We also thank Steve Cohn for their guidance and support throughout the project. Finally, we are thankful for the contributions of the graduate students at the University of Colorado Denver: Tobby Lie, Khang Nguyen, Masoumeh Abolfathi, Rumana Sultana, and Ryan Cheng. This work was supported by CDOT contract 2025-02.

Executive Summary

Bridge engineers in governmental transportation agencies need to regularly forecast the deterioration condition of the bridges under their supervision in order to develop bridge maintenance plans, and even more importantly, identify anomalous bridge deterioration that can result in bridge accidents. Since 1970's, several U.S. Acts have mandated all local and state transportation agencies across the nation to perform regular inspections of the bridges (and culverts) in the regions under their jurisdiction. These inspections have generated valuable historical databases of bridge performance data, which have remained considerably underutilized to date. In this project, with the advent of machine learning and data mining methods, we envisioned data-driven solutions that could derive valuable hidden knowledge from these databases, the knowledge that could be effectively utilized for enhanced bridge management. Toward this end, with this study we have developed a hybrid deep learning methodology that combines advanced data-driven artificial intelligence (AI) models (namely, deep learning model) with traditional mechanistic models (namely, physics-based models) to accelerate model learning and improved performance that can leverage the existing historical bridge (and culvert) performance data, as well as weather data and traffic data, to enable (1) accurate bridge deterioration forecasting (i.e., predictive analytics), and (2) An interactive bridge anomaly detection framework that identifies abnormal deterioration patterns and potential inspection or reporting inconsistencies to enable early detection of bridge (and culvert) performance anomalies. With extensive experimental evaluation using multi-modal real datasets, including bridge performance data, traffic data, and weather data for all bridges in Colorado, we have demonstrated that a selection of our proposed models significantly outperforms existing models for the deterioration forecasting.

Accordingly, we have turned the deep learning models along with their physics-guided extension developed under this project into an software system dubbed **Intelligent Bridge Management (i-BM)**, an advanced, data-driven artificial intelligence (AI) tool that can facilitate effective bridge management for Colorado Department of Transportation (CDOT) bridge engineers. Unlike existing bridge management tools such as BrM (i.e., the AASHTO sponsored

Bridge Management software) used by most of the bridge engineers across the nation, to the best of our knowledge, our proposed tool is the first to make accurate deterioration forecasts/predictions based on historical data, in a similar way weather forecasts are generated. This tool is developed as a standalone, web-based, and user-friendly software application.

Implementation Statement

We have developed and delivered a tool for bridge deterioration forecasting and bridge anomaly detection that can make accurate deterioration forecasts/predictions based on historical data. We have demonstrated accuracy of our proposed advanced physics-guided data-driven AI models (namely, physics-guided deep learning models) that enable this tool via extensive and rigorous experimental evaluation using multi-modal real datasets including bridge performance data, traffic data and weather data for all bridges in Colorado, and shown that a selection of our proposed models significantly outperform existing models for bridge deterioration forecasting and bridge anomaly detection.

We recommend the adoption of the developed tool by bridge engineers in state and local transportation agencies to enable further accurate and enhanced bridge deterioration forecasting, which in turn can improve the ability of these agencies for more cost-effective and efficient bridge management and maintenance planning.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Bridges deteriorate with time and use. The deterioration process is affected by several factors, such as materials, structural design and behavior, daily traffic, freeze and thaw cycles, climate, pollution, and temperature variation [24-26]. After a certain period of time has elapsed, the deterioration processes accelerate, and in a relatively short time interval, the components can lose the capacity to carry the loads they were designed to support.

To address this national issue, several US Acts [27] mandate the state and local governmental agencies (including cities, state transportation agencies, etc.) to perform regular bridge inspections. These Acts define the requirements, periodicity, and procedures for such inspections in the US. Inspections are required to assess the extension, implications, and current state of deterioration processes that may exist, and they need to be performed at regular time intervals, typically every 2 years. A bridge report is generated after each inspection. All bridge reports collect and offer specific data about the health of the inspected bridge, including condition rating, structure identification, year built, average daily traffic, and average daily truck traffic. For example, condition ratings (aka condition indexes) are quantitative descriptors of the state of structural parts that can be used in the assessment for the structure's maintenance [26, 27]. By associating a deteriorated state with a number, instead of using a qualitative description of the state, much more flexibility can be achieved in monitoring groups of similar structures [28-33]. The adoption of condition ratings in the evaluation of structures allows consistency and uniformity, making it possible to compare structural performance, establish priorities, and also prevent failures and accidents.

The aforementioned inspections across the nation, which have been conducted since the 1970s (including our region), have generated valuable historic databases of bridge data based in local and state governmental agencies. While these agencies currently use these inspections to prevent failure and to administrate the national bridge network by setting priorities and establishing criteria to allocate available resources to the structures in the most critical conditions, we believe these databases are heavily underutilized. In particular, with the advent of machine learning and data mining methods, we envision data-driven solutions that can derive much more valuable hidden knowledge that can be utilized for enhanced bridge management.

While in the past, various data-driven deterioration models, including Bayesian models, Probit models, and Markov chains, were proposed in the literature to model bridge deterioration [24, 25, 34-38], these models either suffer from low accuracy or are too complex to be applicable. Moreover, they only address the problem of deterioration forecasting. Recently, deep learning has been shown to significantly outperform other analytical modeling methodologies in a variety of application domains, such as computational biology, Electronic Health Record (EHR) data analysis, activity detection, scene labeling, image captioning, and object detection [39-47]. In the past, we have introduced and deployed various deep learning-based models, e.g., for sleep stage classification using brain signals [48, 49], mobility monitoring [50, 51], and activity classification [52]. In our previous study, we proposed to develop deep learning models for enhanced bridge management. In particular, we focused on the two problems of bridge subtyping (descriptive analysis) and data-driven bridge deterioration forecasting (predictive analysis) [53].

Traditionally, many researchers have introduced mechanistic physics-based methods, such as simulation-based [54] and finite element modeling, to explore and predict signs of deterioration such as corrosion, fatigue, and cracking [55]. Because data-driven models overlook physics-based mechanisms and rely solely on data patterns, while physics-based models disregard real historical data and depend only on equations and parameters, a persistent gap remains in deterioration forecasting. In many other domains, in order to bridge the similar research gap, researchers have introduced the so-called physics-informed neural networks (PINN) [56-58], which integrate physics-based models with AI-driven data-driven models to facilitate model learning from the valuable insights of both modeling approaches. In this study, we extend our earlier work by developing a full-fledged, data-driven software platform that integrates advanced AI models for bridge monitoring and decision support in a user-friendly graphical user interface with enhanced operational features. This platform incorporates previously developed deep learning models as well as additional hybrid physics-guided models that embed physics-based knowledge into the data-driven models. Moreover, anomalous data points can hamper model training and negatively affect model performance. To address this issue, we have designed an anomaly detection framework as part of this study to identify and manage anomalies in real-time bridge data. Effective solutions for these problems will significantly advance the state-of-the-art in bridge management.

Below, we summarize three contributions of this research project:

1. **Intelligent Bridge Management (i-BM) System:** We have undertaken a full-phased software development effort to design and implement the Intelligent Bridge Management (i-BM) system—an integrated, user-friendly platform that unifies data management, forecasting, and anomaly detection modules. The i-BM software serves as an operational environment where bridge engineers can visualize, analyze, and manage bridge performance data through interactive, data-driven, and physics-informed tools.
2. **Physics-Guided Bridge Deterioration Forecasting Model for Predictive Analysis:** With this advanced deterioration forecasting model, one can perform predictive analysis of bridges through accurate forecasting of quantitative descriptors of structural deterioration (e.g., condition ratings) by integrating both data-driven learning and physics-based principles. The incorporation of physics-informed constraints enables the model to capture real-world deterioration mechanisms more accurately, ensuring parameter-consistent and interpretable predictions. Accurate prediction of these descriptors is not only crucial for establishing maintenance priorities and performing proactive bridge monitoring with optimized resource allocation, but also, more importantly, essential for failure prevention.

3. **Anomaly Detection Algorithm for Data Quality and Predictive Diagnostics**: With this novel algorithm, one can perform automated detection of anomalous bridge performance, addressing issues that may lead to unexpected failures or accidents. This tool enables diagnostic and predictive analysis of bridge performance data by (1) automatically identifying anomalous or inconsistent patterns in historical and real-time bridge condition data abnormal deterioration trends; (2) validating and refining anomaly labels through an **interactive active learning interface**, where users can visually review

and confirm anomalies; and (3) continuously retraining the model using these user-verified samples to enhance its detection capabilities across new datasets and use cases. This iterative learning process enables the users to perform predictive analysis of the bridge performance by accurate prediction of quantitative descriptors for the structure deterioration state (e.g., condition ratings) while proactively identifying and flagging possible anomalies in the deterioration pattern of the bridge structure in advance. Accurate prediction of such anomalies is essential not only for improving the quality and reliability of the data but also for failure prevention by supporting early detection of potential safety issues (e.g., bridge accidents), facilitating timely maintenance, and informed bridge management decisions.

Note that the second and third algorithmic contributions are also incorporated in i-BM to offer all aforementioned capabilities under a unified software platform.

I-BM allows for enhanced bridge management by improving depth, accuracy, and efficiency/speed in descriptive, diagnostic, and predictive analysis of the historic bridge data reported by bridge inspectors. In turn, this can lead to more effective resource allocation for bridge monitoring, maintenance, and construction.

In the remainder of this report, Section 2 introduces the Intelligent Bridge Management (i-BM) system, describing its overall architecture, software modules, and underlying technologies. It also details the integrated data sources and the functionalities of each major component, including the Forecasting and Anomaly Detection modules. Thereafter, Section 3 presents the Physics-Guided Bridge Deterioration Forecasting approach, including related work, methodological framework, and model evaluation. Section 4 focuses on the Anomaly Detection Framework for Bridge Deterioration, reviewing relevant literature and explaining the underlying methods and implementation strategy, respectively. Finally, in Section 5, we will conclude and briefly discuss future directions for this research and system enhancement.

Chapter 2. Intelligent Bridge Management (i-BM) System

The Intelligent Bridge Management (i-BM) system is a user-friendly, data-driven web-based software platform equipped with a graphical user interface and advanced analytical capabilities. It is designed to assist CDOT bridge engineers in making accurate, data-informed predictions of bridge and culvert deterioration and to support proactive maintenance planning. The i-BM system integrates multiple data sources—such as the National Bridge Inventory (NBI) and weather datasets—along with several artificial intelligence (AI) modules, including baseline deep learning forecasting models, physics-guided forecasting models, and anomaly detection frameworks, into a unified operational environment for intelligent decision-making. Its interconnected architecture enables seamless communication among components, ensuring efficient data processing, model execution, and visualization through an interactive interface. This section presents the overall system design, architecture, software modules, and core technologies that comprehensively enable the i-BM system's intelligent bridge management capabilities. The subsequent sections—2.1 (System Architecture), 2.2 (Data Module), 2.3 (Forecast Module), and 2.4 (Anomaly Detection Module)—describe these components and their workflows in detail.

2.1 System Architecture

The Intelligent Bridge Management (i-BM) system follows a modular, web-based, cloud-hosted architecture that integrates data management, model development, model registry, and deployment components within a uniform structure. The system is designed for scalability, interoperability, and efficient execution of AI-driven bridge management tasks, enabling smooth interaction between server-side data processing and client-side visualization. The overall system architecture of the i-BM platform is illustrated in Figure 1, which depicts the interaction among key components and the data flow across different layers—from data ingestion and preprocessing to model training, registry, deployment, and visualization through the user interface.

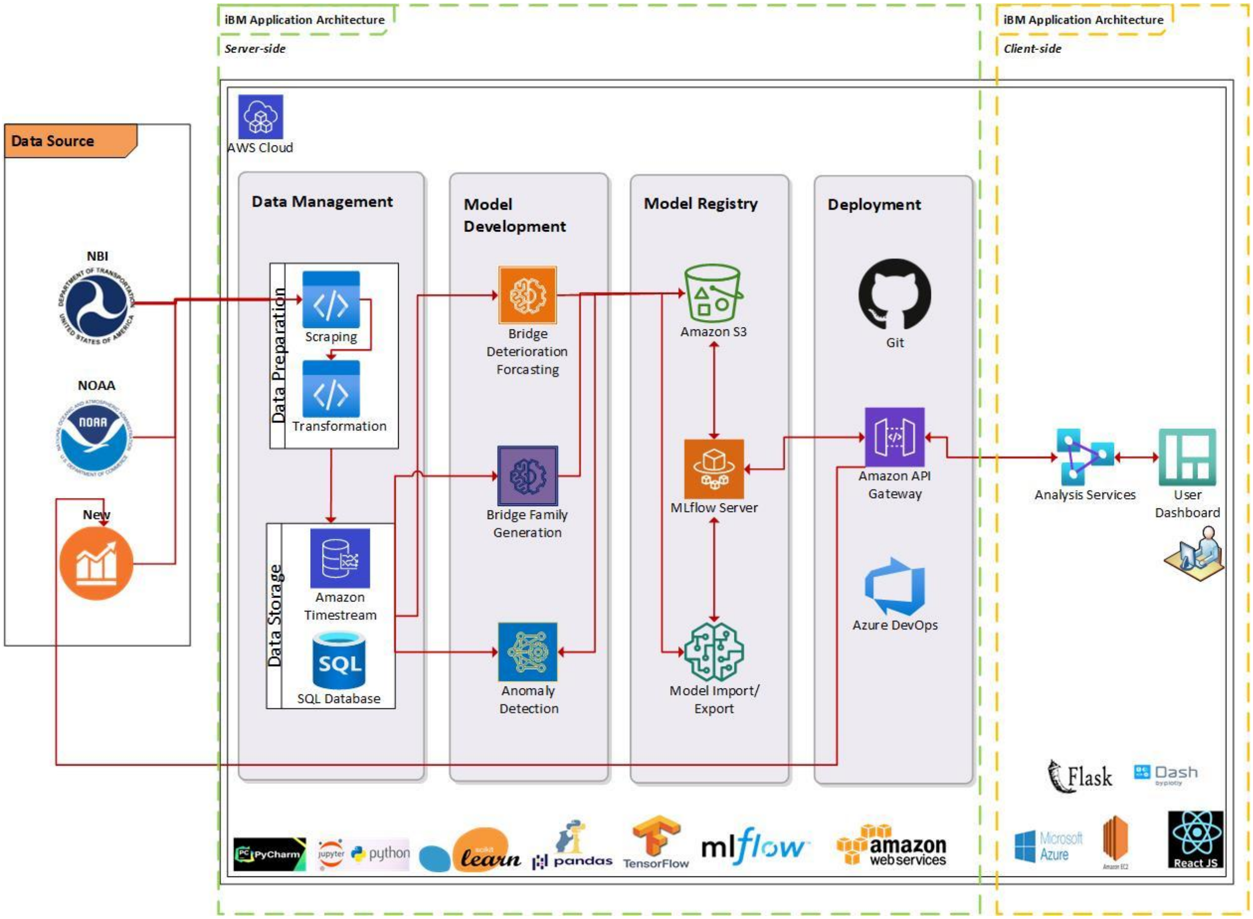


Figure 1. Overall Architecture of the Intelligent Bridge Management (i-BM) System

2.1.1 Server-Side Architecture

The i-BM backend is hosted on an Amazon EC2 instance within the Colorado Department of Transportation (CDOT) cloud infrastructure. The system adopts microservices-based architecture, where multiple backend services—such as Data Management, Forecasting, Anomaly Detection, Family Generation, and Model Registry—are deployed as independent Docker containers or system-level services. This containerized structure ensures modularity, scalability, and ease of maintenance across the platform. All services are integrated under a single secure public interface, accessible via https://intelligencebmtool.codot.gov, which is routed through an AWS Application Load Balancer (ALB). The ALB dynamically manages web traffic across services, ensuring high availability, fault tolerance, and optimized performance. This deployment strategy allows each service to be updated or scaled independently without system downtime, providing a robust and flexible operational environment for intelligent bridge management. The different layers of the architecture are described in detail in the following subsections.

2.1.1.1 Data Source Layer

The system integrates data from multiple authoritative sources, including the National Bridge Inventory (NBI), the National Oceanic and Atmospheric Administration (NOAA) Weather Database, and CDOT's repair datasets. These serve as foundational inputs for both predictive and diagnostic modeling.

2.1.1.2 Data Management Layer

The Data Management Layer handles data ingestion, preprocessing, transformation, and storage. Raw datasets from multiple sources introduced in the previous section are collected, cleaned, and standardized through automated ETL (Extract, Transform, Load) pipelines. The system employs a TimescaleDB (PostgreSQL-based) time-series database to manage and query temporal datasets—such as bridge condition histories, traffic volumes, and weather records, while Microsoft SQL Server is used to store static datasets, including bridge inventory and structural attributes. This dual-database configuration enables efficient handling of both dynamic and static data, ensuring high performance and data consistency across the forecasting and anomaly detection modules.

2.1.1.3 Model Development Layer

Three major AI components are implemented in this layer:

- **Bridge Deterioration Forecasting** – incorporates both baseline deep learning and physics-guided models to predict bridge and culvert condition deterioration over time.
- **Bridge Family Generation** – currently implemented using a clustering algorithm that groups newly added bridges with similar existing structures. This facilitates smoother forecasting for bridges with limited historical data, providing foundational support for future enhancement into a full descriptive analytics module.

- **Anomaly Detection** – identifies and flags abnormal deterioration trends or data inconsistencies using an interactive active learning framework to improve data quality and model reliability.

2.1.1.4 Model Registry Layer

Trained models and their metadata are tracked and version-controlled using MLflow, which serves as the centralized Model Registry for managing the lifecycle of each experiment and trained models. The registry uses a Microsoft SQL Server database as the backend store to maintain model configurations, parameters, metrics, and version history, while model artifacts are saved locally within the /mlruns directory on the EC2 instance. This configuration ensures reproducibility, transparency, and efficient management of model versions. Models can be smoothly loaded, registered, retrieved, and deployed within the i-BM system for real-time forecasting and anomaly detection tasks.

2.1.1.5 Deployment Layer

The Deployment Layer manages the delivery and operation of all i-BM services through a hybrid environment hosted on an Amazon EC2 instance. Core analytical services—such as data management, forecasting, family generation, and anomaly detection—run as persistent background processes managed via system-level services, while supporting services such as databases and authentication are containerized using Docker to allow modular updates and isolated operation. The frontend (React-based) interface is built and served using npm, while the backend services (Node.js/Express and Python APIs) run continuously on the same EC2 instance. All web traffic is routed through an AWS Application Load Balancer (ALB), which performs path-based routing to direct user requests to the corresponding service. This setup ensures reliable load distribution, high availability, and flawless integration between the analytical backend and the user-facing dashboard, supporting continuous and uninterrupted operation of the i-BM platform.

2.1.2 Client-Side Architecture

The client interface consists of a web-based interactive visualization dashboards developed using React.js, allowing CDOT bridge engineers to generate and access analysis results, visualize model outputs, and interact with bridge data in real time. The web interface communicates securely with the backend services through a Flask-based API layer, ensuring efficient data exchange between the analytical engine and the user interface.

2.1.3 Technological Stack

The i-BM system is developed using a robust stack of technologies, including Python, Pandas, TensorFlow, Scikit-learn, React.js, Flask, MLflow, TimescaleDB, and Microsoft SQL Server. Development and experimentation are supported through Jupyter Notebook, Visual Studio Code, and PyCharm, while Git and Azure DevOps are used for version control, collaborative development, and codebase management. Deployment is managed on an Amazon EC2 instance using Docker-based containerization and path-based routing through an AWS Application Load Balancer (ALB).

2.2 Data Module

The Data Module in the Intelligent Bridge Management (i-BM) system serves as the foundation for all data-driven operations, enabling the import, organization, exploration, and export of bridge-related datasets. It provides a combined interface that connects multiple data sources, backend databases, and analytical modules, ensuring smooth data exchange across the i-BM platform. Additional details are discussed in the following subsections.

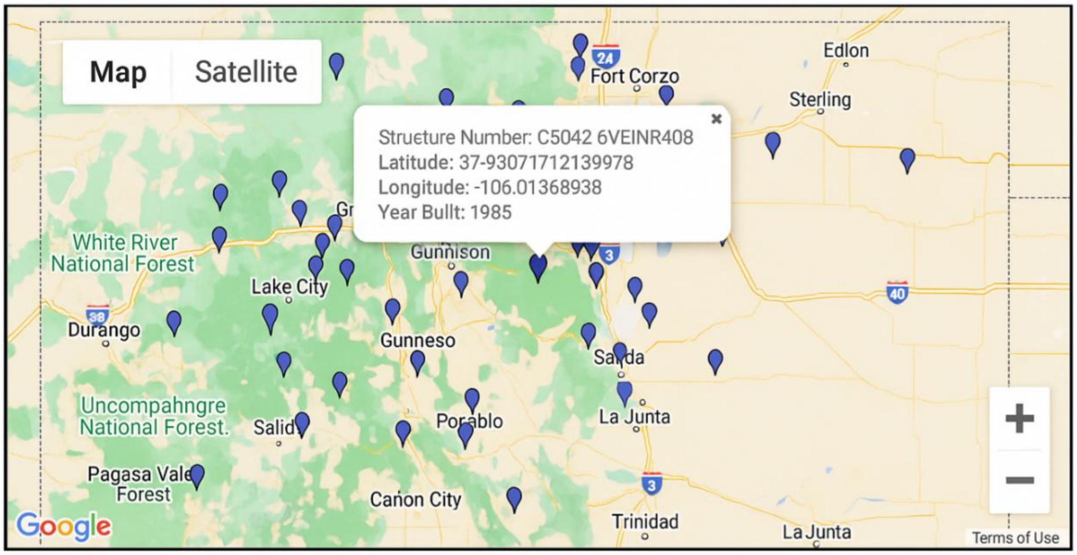
2.2.1 Data Sources

The i-BM platform integrates multiple authoritative data sources that provide the foundation for its analytical and predictive capabilities. These datasets inclusively capture structural, traffic, and environmental characteristics of bridges and culverts across Colorado.

- **National Bridge Inventory (NBI):**

  For the bridge evaluation and traffic data, we utilized publicly available data from the National Bridge Inventory (NBI) [1] database. This dataset contains national bridge inspection/evaluation data collected over the years 1992-current for different bridge structures all over the USA. Figure 2 illustrates a map of bridges within Colorado. We extracted 32 primary-level features - Year Built, ADT, Traffic Lanes on, Traffic Lanes

Under, Design Load, Structure Type, Main Unit Spans, Approach Unit Spans, Horizontal Clearance Measurement, Maximum Span Length Measurement, Structure Length Measurement, Roadway Width Measurement, Deck Width Measurement, Deck Condition Rating, Superstructure Condition Rating, Substructure Condition Rating, Culvert Condition Rating, Channel Condition Rating, Operating Rating, Inventory Rating, Structural Evaluation, Deck Geometry Evaluation, Under Clearance Evaluation, Posting Evaluation, Waterway Evaluation, Approach Road Evaluation, Traffic Direction, Deck Structure Type, Surface Type, Deck Protection, Percent ADT Truck, Future ADT - detailed in Table 1, from this dataset with the help of a bridge engineering expert which served as inputs into our model for each year's worth of evaluations.



Structure Number: C5042 6VEINR408
Latitude: 37-93071712139978
Longitude: -106.01368938
Year Bullt: 1985

Figure 2. Map of Bridges (Blue Markers) Within Colorado State

Table 1. NBI Bridge and Culvert Features (Gold Indicates Traffic Input Data, Sky Blue Indicates Bridge Evaluation Input Data, And Blue with Red Border Indicates Bridge Condition Rating Input/Output Data)

Feature	**NBI Item No.**	**Description**	**Data Type**	**Target Forecasting Condition**
Year Built	027	The construction year of the structure	Datetime	Deck, Superstructure, Substructure, Culvert
ADT	029	Average daily traffic volume of a bridge	Number	Deck, Superstructure, Substructure, Culvert
Traffic Lanes on	028A	Number of traffic lanes on a bridge structure	Number	Deck, Superstructure, Substructure, Culvert
Traffic Lanes Under	028B	Number of traffic lanes under a bridge structure	Number	Deck, Superstructure, Substructure, Culvert

Design Load	031	The live load for which the structure was designed	Number	Deck, Superstructure, Substructure, Culvert
Structure Type	043B	Type of structural design of the construction	Number	Deck, Superstructure, Substructure, Culvert
Main Unit Spans	045	Number of spans in the main unit	Number	Deck, Superstructure, Substructure, Culvert
Appr Unit Spans	046	Number of approach spans in the major bridge	Number	Deck, Superstructure, Substructure, Culvert
Horizontal Clearance Measurement	047	Largest available horizontal clearance for wide loads	Number	Deck, Superstructure, Substructure, Culvert
Maximum Span Length Measurement	048	Length of the maximum span	Number	Deck, Superstructure, Substructure, Culvert
Structure Length Measurement	049	Total length of the structure	Number	Deck, Superstructure, Substructure, Culvert
Roadway Width Measurement	051	Distance between curbs or rails on the structure roadway	Number	Deck, Superstructure, Substructure, Culvert
Deck Width Measurement	052	Out-to-out width of the deck	Number	Deck, Superstructure, Substructure
Deck Condition Rating	058	Overall condition rating of the bridge deck (1 to 9)	Number	Deck, Superstructure, Substructure
Superstructure Condition Rating	059	Physical condition of all structural members (1 to 9)	Number	Deck, Superstructure, Substructure
Substructure Condition Rating	060	Physical condition of piers, abutments, and other elements (1 to 9)	Number	Deck, Superstructure, Substructure
Culvert Condition Rating	062	Evaluates alignment, settlement, joints, scour, etc.	Number	Culvert
Channel Condition Rating	061	Physical condition related to the water flow through the bridge	Number	Deck, Superstructure, Substructure, Culvert
Operating Rating	064	Numeric value indicating the structure's service sufficiency	Number	Deck, Superstructure, Substructure, Culvert
Inventory Rating	066	Load level for safe, indefinite use	Number	Deck, Superstructure, Substructure, Culvert
Structural Evaluation	067	Evaluation score of the structure	Number	Deck, Superstructure, Substructure, Culvert
Deck Geometry Evaluation	068	Evaluation score for deck geometry	Number	Deck, Superstructure, Substructure
Under Clearance Evaluation	069	Measures vertical and horizontal under clearances	Number	Deck, Superstructure, Substructure
Posting Evaluation	070	Load limit postings when the legal load exceeds the operating rating	Number	Deck, Superstructure, Substructure, Culvert
Waterway Evaluation	071	Likelihood of bridge overtopping	Number	Deck, Superstructure, Substructure, Culvert
Approach Road Evaluation	072	Evaluation of the approach roadway alignment	Number	Deck, Superstructure, Substructure, Culvert
Traffic Direction	102	Direction of traffic flow	Number	Deck, Superstructure, Substructure, Culvert

Deck Structure Type	107	Type of deck structure	Number	Deck, Superstructure, Substructure
Surface Type	108A	Type of wearing surface on the deck	Number	Deck, Superstructure, Substructure
Deck Protection	108C	Protective system on the bridge deck	Number	Deck, Superstructure, Substructure
Percent ADT Truck	109	Percentage of daily truck traffic	Number	Deck, Superstructure, Substructure, Culvert
Future ADT	114	Projected average daily traffic	Number	Deck, Superstructure, Substructure, Culvert

Note: Background colors are used to visually distinguish categories of input data. **Gold** indicates average daily traffic (ADT)–related variables. **Sky blue** indicates bridge or culvert evaluation variables, such as structural evaluation, deck geometry evaluation, etc., and **Blue with a red border** indicates bridge condition rating input or output variables (e.g., deck condition rating, culvert condition rating, etc.) used in forecasting.

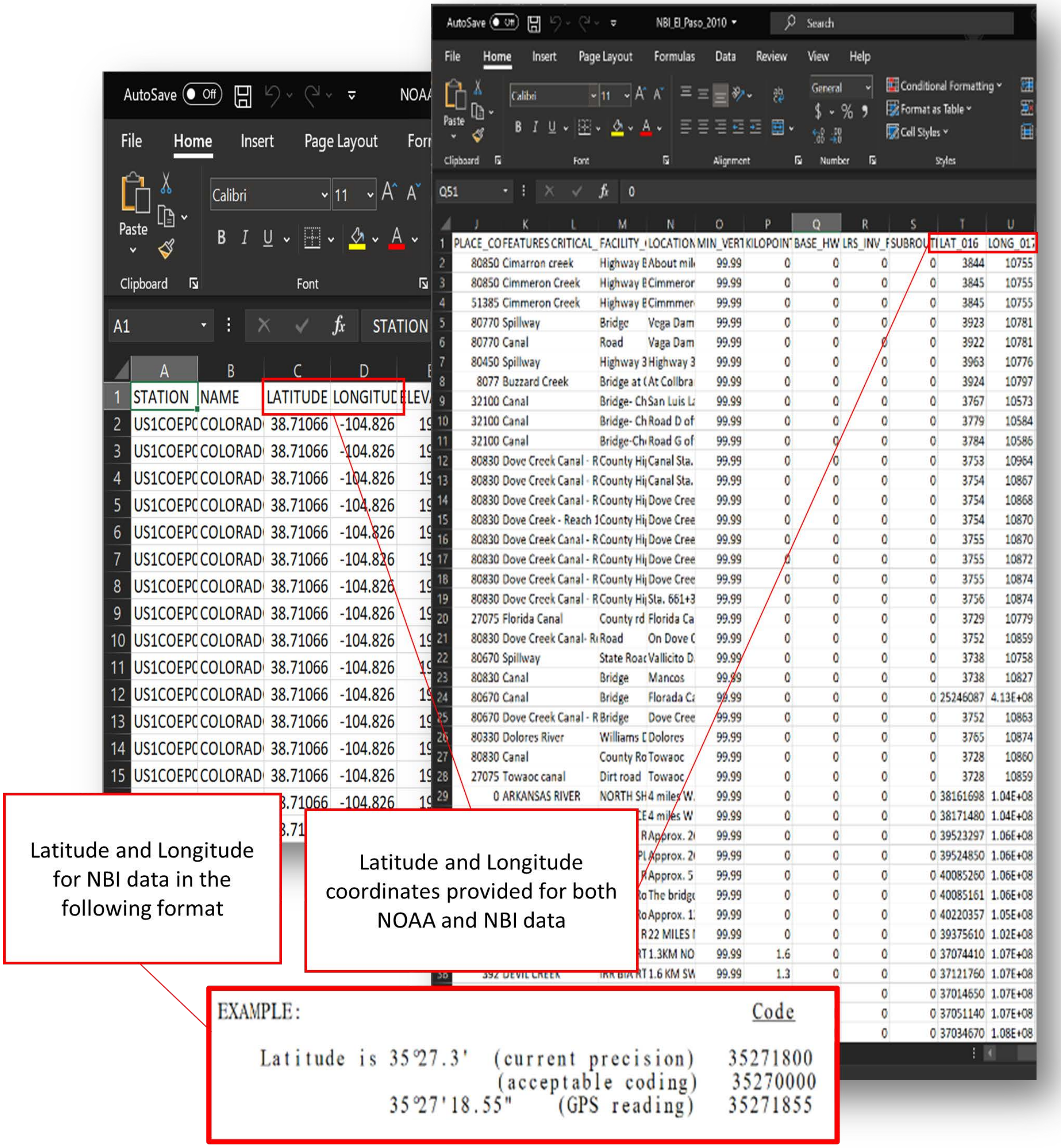
- **National Oceanic and Atmospheric Administration (NOAA):**

  For the weather data, we utilized another publicly available dataset from the National Oceanic and Atmospheric Administration’s (NOAA) online weather database, the dataset is the Normals Daily dataset [2]. This dataset contains daily precipitation and snow records over global land areas. The following five features from this data were used: Precipitation, snowfall, snow depth and maximum temperature, Average temperature, and Minimum Temperature. Once again, domain experts were consulted to determine relevant features for the task of bridge deterioration forecasting. There was an emphasis on the importance of precipitation in predicting bridge deterioration forecasting. This is because protective epoxy coatings may be affected negatively when moisture is introduced to a bridge structure. In addition to this, there was also an importance in considering snowfall and snow depth since the freeze/thaw cycles of bridge structures contribute to how bridges may deteriorate, as well as how maintenance strategies are scheduled. Finally, the daily temperature has an effect on the bridge deck over time.

- **Traffic and Repair Data (CDOT):**

  The Colorado Department of Transportation (CDOT) provides supplemental datasets containing traffic statistics and repair histories. These data are linked with the NBI and NOAA records within the i-BM databases to support forecasting analyses.

All external data sources (e.g., NOAA and NBI datasets) are preprocessed and synchronized through the i-BM Data Source Service. The service integrates datasets based on latitude and longitude coordinates and aligns them before storing them in the databases. **Figure 3** illustrates the preprocessing approach where bridge and weather station data are matched by geographic coordinates and standardized for forecasting.



	A	B	C	D	E
1	STATION	NAME	LATITUDE	LONGITUD	ELEV
2	US1COEPC	COLORAD	38.71066	-104.826	19
3	US1COEPC	COLORAD	38.71066	-104.826	19
4	US1COEPC	COLORAD	38.71066	-104.826	19
5	US1COEPC	COLORAD	38.71066	-104.826	19
6	US1COEPC	COLORAD	38.71066	-104.826	19
7	US1COEPC	COLORAD	38.71066	-104.826	19
8	US1COEPC	COLORAD	38.71066	-104.826	19
9	US1COEPC	COLORAD	38.71066	-104.826	19
10	US1COEPC	COLORAD	38.71066	-104.826	19
11	US1COEPC	COLORAD	38.71066	-104.826	19
12	US1COEPC	COLORAD	38.71066	-104.826	19
13	US1COEPC	COLORAD	38.71066	-104.826	19
14	US1COEPC	COLORAD	38.71066	-104.826	19
15	US1COEPC	COLORAD	38.71066	-104.826	19
			3.71066	-104.826	19
			3.71		

	J	K	L	M	N	O	P	Q	R	S	T	U
1	PLACE_CO	FEATURES	CRITICAL_	FACILITY_	LOCATION	MIN_VERT	KILOPOINT	BASE_HW	LRS_INV_F	SUBROUTI	LAT_016	LONG_017
2	80850	Cimarron creek		Highway E	About mil	99.99	0	0	0	0	3844	10755
3	80850	Cimmeron Creek		Highway E	Cimmeror	99.99	0	0	0	0	3845	10755
4	51385	Cimmeron Creek		Highway E	Cimmmer	99.99	0	0	0	0	3845	10755
5	80770	Spillway		Bridge	Vega Dam	99.99	0	0	0	0	3923	10781
6	80770	Canal		Road	Vaga Dam	99.99	0	0	0	0	3922	10781
7	80450	Spillway		Highway 3	Highway 3	99.99	0	0	0	0	3963	10776
8	8077	Buzzard Creek		Bridge at (	At Collbra	99.99	0	0	0	0	3924	10797
9	32100	Canal		Bridge- Ch	San Luis L	99.99	0	0	0	0	3767	10573
10	32100	Canal		Bridge- Ch	Road D of	99.99	0	0	0	0	3779	10584
11	32100	Canal		Bridge-Ch	Road G of	99.99	0	0	0	0	3784	10586
12	80830	Dove Creek Canal - R		County Hi	Canal Sta.	99.99	0	0	0	0	3753	10964
13	80830	Dove Creek Canal - R		County Hi	Canal Sta.	99.99	0	0	0	0	3754	10867
14	80830	Dove Creek Canal - R		County Hi	Dove Cree	99.99	0	0	0	0	3754	10868
15	80830	Dove Creek - Reach 1		County Hi	Dove Cree	99.99	0	0	0	0	3754	10870
16	80830	Dove Creek Canal - R		County Hi	Dove Cree	99.99	0	0	0	0	3755	10870
17	80830	Dove Creek Canal - R		County Hi	Dove Cree	99.99	0	0	0	0	3755	10872
18	80830	Dove Creek Canal - R		County Hi	Dove Cree	99.99	0	0	0	0	3755	10874
19	80830	Dove Creek Canal - R		County Hi	Sta. 661+3	99.99	0	0	0	0	3756	10874
20	27075	Florida Canal		County rd	Florida Ca	99.99	0	0	0	0	3729	10779
21	80830	Dove Creek Canal- R		Road	On Dove C	99.99	0	0	0	0	3752	10859
22	80670	Spillway		State Roac	Vallicito D	99.99	0	0	0	0	3738	10758
23	80830	Canal		Bridge	Mancos	99.99	0	0	0	0	3738	10827
24	80670	Canal		Bridge	Florada C	99.99	0	0	0	0	25246087	4.13E+08
25	80670	Dove Creek Canal - R		Bridge	Dove Cree	99.99	0	0	0	0	3752	10863
26	80330	Dolores River		Williams C	Dolores	99.99	0	0	0	0	3765	10874
27	80830	Canal		County Ro	Towaoc	99.99	0	0	0	0	3728	10860
28	27075	Towaoc canal		Dirt road	Towaoc	99.99	0	0	0	0	3728	10859
29	0	ARKANSAS RIVER		NORTH SH	4 miles W.	99.99	0	0	0	0	38161698	1.04E+08
				E	4 miles W	99.99	0	0	0	0	38171480	1.04E+08
				R	Approx. 2	99.99	0	0	0	0	39523297	1.06E+08
				PL	Approx. 2	99.99	0	0	0	0	39524850	1.06E+08
				R	Approx. 5	99.99	0	0	0	0	40085260	1.06E+08
				Ro	The bridg	99.99	0	0	0	0	40085161	1.06E+08
				Ro	Approx. 1	99.99	0	0	0	0	40220357	1.05E+08
				R	22 MILES	99.99	0	0	0	0	39375610	1.02E+08
				RT	1.3KM NO	99.99	1.6	0	0	0	37074410	1.07E+08
38	392	DEVIL CREEK		IRR BIA RT	1.6 KM SW	99.99	1.3	0	0	0	37121760	1.07E+08
									0	0	37014650	1.07E+08
									0	0	37051140	1.07E+08
									0	0	37034670	1.08E+08

Latitude and Longitude for NBI data in the following format

Latitude and Longitude coordinates provided for both NOAA and NBI data

EXAMPLE:		Code
Latitude is 35°27.3'	(current precision)	35271800
	(acceptable coding)	35270000
35°27'18.55"	(GPS reading)	35271855

Figure 3. Illustration of NOAA and NBI Datasets Showing Coordinates used for Localization

2.2.2 Data Source Service Architecture

The **Data Source Service** operates as the backend engine that connects the i-BM frontend with the system's two core databases:

- **Microsoft SQL Server**, which stores static data tables such as StructureStaticTbl, GroupStructureMapping, and GroupTable containing bridge inventory and metadata.
- **TimescaleDB (PostgreSQL-based)**, which stores dynamic and temporal datasets including bridge condition ratings (structuredynamictbl) and weather observations (noaadatatbl).

The Data Source Service processes API requests from the frontend, queries both databases through the appropriate data connectors, and returns structured responses to the user interface. This dual-database configuration enables efficient querying of both static and time-series data, ensuring fast access and synchronization across analytical modules. **Figure 4** illustrates the backend data source architecture connecting the frontend interface to the SQL Server and TimescaleDB environments.

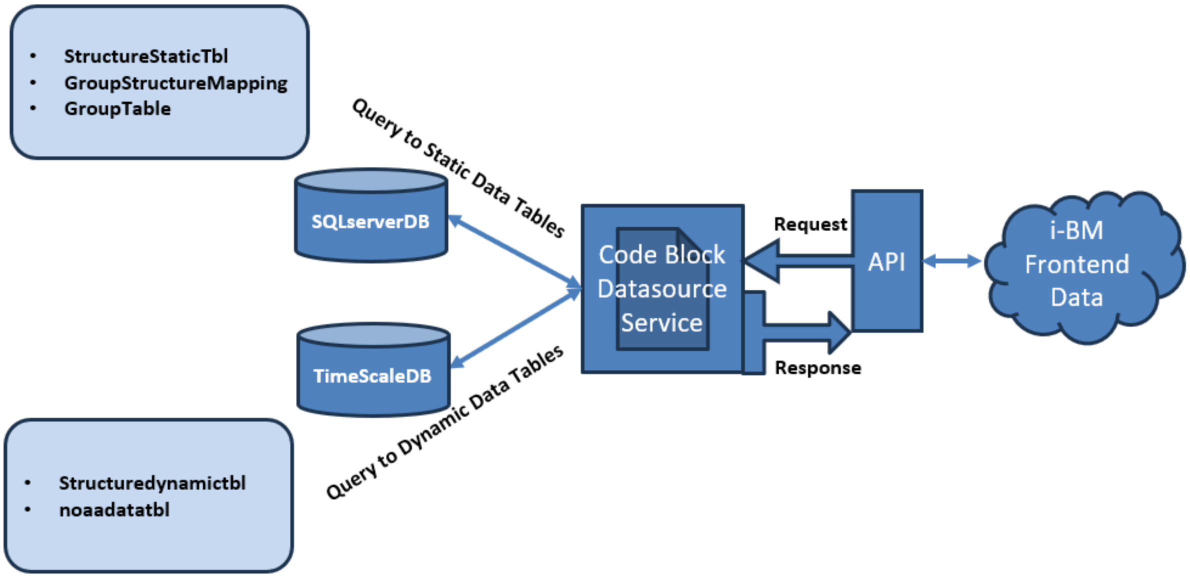


Figure 4. Data Source Service Backend Architecture

2.2.3 Database and Data Table Design

The i-BM system connects both relational and time-series databases to manage bridge, culvert, and weather datasets efficiently. The details database schema is explained in the next subsections.

2.2.3.1 Backend Schema and Relationships

The database architecture follows a modular and relational schema that connects multiple functional tables to maintain organized data flow across the system. The design incorporates three major categories of tables:

1. **Static Data Tables** – store non-changing bridge inventory attributes derived from the National Bridge Inventory (NBI). These include:
    - StructureStaticTbl – contains location (LAT_016, LONG_017), construction (YEAR_BUILT_027), and geometric attributes (TRAFFIC_LANES_ON_028A, DESIGN_LOAD_031).
    - GroupTable – stores user-defined grouping information with fields such as Group_Name, Construction_Year, and Structure_Category.
    - GroupStructureMapping – serves as a bridge between GroupTable and StructureStaticTbl, linking each structure to one or more analysis groups.
2. **Dynamic Data Tables** – store annual, time-varying bridge and environmental data retrieved from NBI and NOAA. These include:
    - structuredynamictbl – records yearly measurements such as Average Daily Traffic (ADT), deck and substructure condition ratings, roadway width, and operating ratings.
    - noaadatatbl – contains environmental parameters such as precipitation, snowfall, and temperature averages corresponding to each bridge location and year.
3. **Analytical and Support Tables** – store model and analysis-related information generated within the i-BM system. These include:
    - registered_models and trained_models – maintain model registry, configurations, and training metadata.

- condition_ratings – stores forecasted condition outputs.
- repairlisttbl – records repair history and condition updates.

Both static and dynamic tables are linked through the primary key Structure_number_008, ensuring that static bridge identifiers are synchronized with yearly inspection and weather datasets. This relational integrity allows data queries to be joined appropriately between SQL Server and TimescaleDB.

Figure 5 illustrates the complete backend database schema, including relationships among static, dynamic, and analytical tables that support the forecasting and anomaly detection modules.

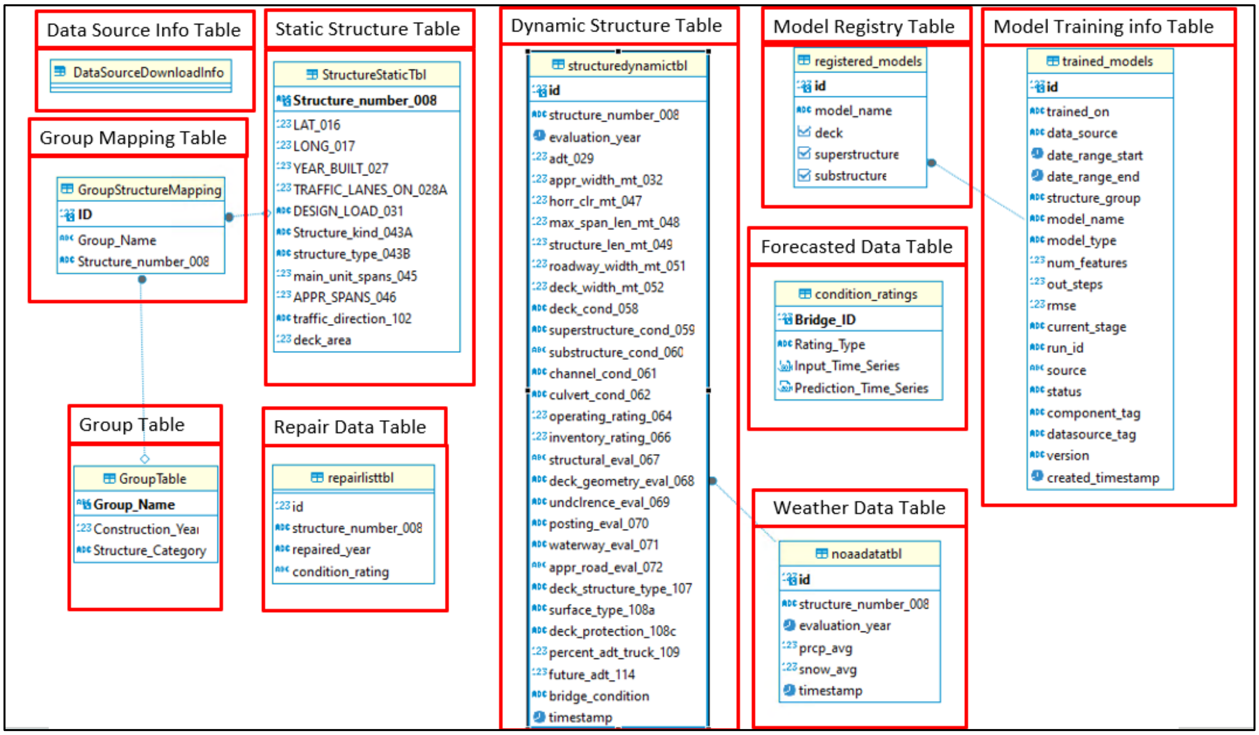


Figure 5. i-BM Backend Database Design

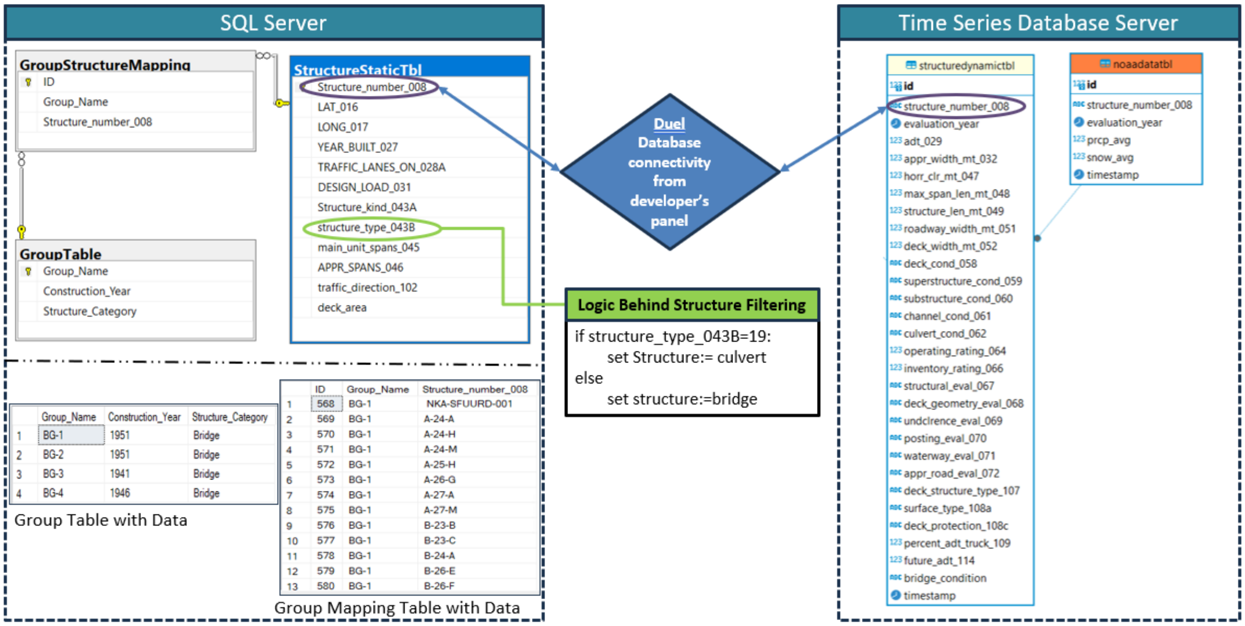
2.2.3.2 Bridge and Culvert Distinction

A key backend function in the Data Source Service is automatic structure categorization, which determines whether a record belongs to a bridge or a culvert. The classification logic uses the field structure_type_043B from the static table:

```
if structure_type_043B == 19:
    set Structure := "Culvert"
                        else:
    set Structure := "Bridge"
```

This conditional logic is applied during data import and retrieval, ensuring consistent filtering across all modules. It also allows users to manage datasets separately for bridges and culverts while maintaining shared architecture and the same backend code.

The logic operates at the service layer, immediately after querying the SQL Server and before sending structured responses to the frontend or analytical modules such as Forecasting or Anomaly Detection. **Figure 6** shows the flow of structure classification and data filtering logic within the Data Source Service.



```
if structure_type_043B=19:
    set Structure:= culvert
else
    set structure:=bridge
```

	Group_Name	Construction_Year	Structure_Category
1	BG-1	1951	Bridge
2	BG-2	1951	Bridge
3	BG-3	1941	Bridge
4	BG-4	1946	Bridge

Group Table with Data

	ID	Group_Name	Structure_number_008
1	568	BG-1	NKA-SFUURD-001
2	569	BG-1	A-24-A
3	570	BG-1	A-24-H
4	571	BG-1	A-24-M
5	572	BG-1	A-25-H
6	573	BG-1	A-26-G
7	574	BG-1	A-27-A
8	575	BG-1	A-27-M
9	576	BG-1	B-23-B
10	577	BG-1	B-23-C
11	578	BG-1	B-24-A
12	579	BG-1	B-26-E
13	580	BG-1	B-26-F

Group Mapping Table with Data

Figure 6. Bridge and Culvert Classification and Filtering Logic

2.2.4 Data Module Functionalities

The **Data Module** provides four primary functionalities accessible through the user dashboard.

2.2.4.1 Data Import

Enables users to import bridge and weather datasets directly from NBI and NOAA. Users can select between two update options:

- Update—adds missing data entries.
- Overwrite—replaces existing records completely.

The import interface ensures that data is validated and stored in the correct database (SQL Server for static data and TimescaleDB for dynamic data).

2.2.4.2 Data Grouping

Allows users to create, edit, or delete structure groups individually for bridges and culverts. Users can select structures directly from an interactive map interface using rectangle, circle, or point tools, or upload a list of structures via CSV. Group mapping tables link each structure to its assigned group for downstream analysis.

2.2.4.3 Data Exploration

Provides visualization and tabular access to both historical and non-historical bridge data. Users can view bridge condition trends over time, examine structure attributes, and compare performance across selected groups. Historical data are presented through time-series plots, while non-historical data appear in an interactive grid view.

2.2.4.4 Data Export

Enables users to export selected datasets—including bridge evaluation, traffic, and weather data—to standard formats such as .csv or .xlsx. Users can specify the desired time range, select relevant attributes, and choose to export all structures or grouped subsets. This ensures flexible reporting and integration with external analysis tools.

Figure 7 shows the key data management functionalities developed in the i-BM tool, which allow users to import, export, and explore the bridge dataset.

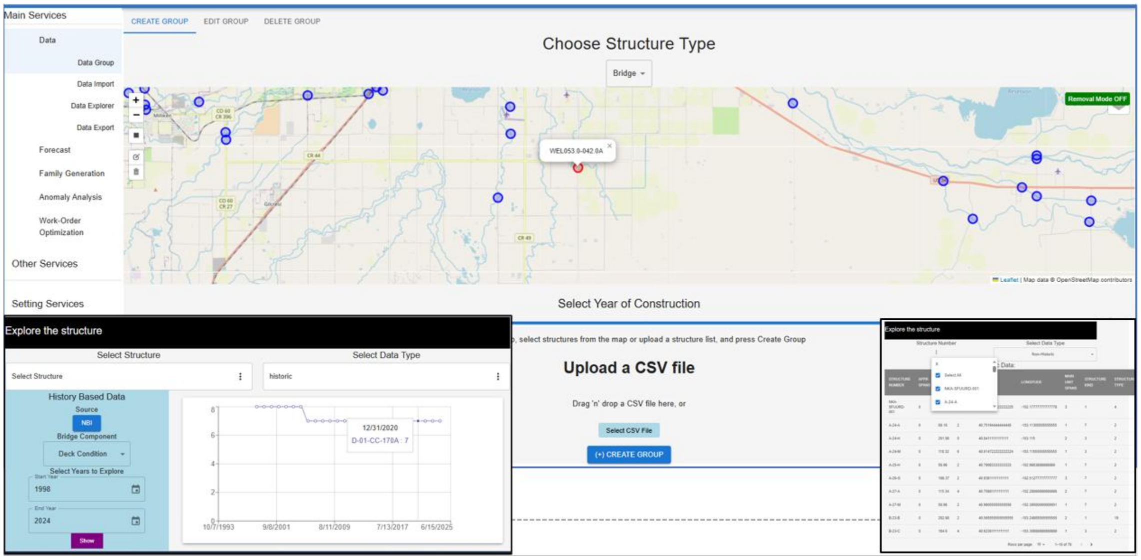


Figure 7 Data Management Module of the i-BM Tool

2.3 Forecast Module

The Forecast Module in the Intelligent Bridge Management (i-BM) system is designed to perform predictive analysis of bridge and culvert condition ratings by utilizing both data-driven and physics-informed machine learning models. This module enables CDOT engineers to proactively estimate future deterioration states and plan maintenance interventions based on quantitative evidence.

2.3.1 Forecast Service Architecture

The Forecast Service provides the functionality to train, manage, and deploy predictive models that estimate future bridge condition ratings (deck, superstructure, substructure, and culvert). It interacts directly with the backend data sources—**SQL Server** and **TimescaleDB**—to retrieve structural, traffic, and weather data required for model training and inference.
Users can configure forecasting parameters, such as input features, time range, and prediction horizon, directly from the i-BM user interface. The forecast results can then be stored in the system database for further analysis or exported as external files for reporting purposes. Figure

8 illustrates the overall backend architecture of the forecasting workflow, showing the interaction between the data sources, model management system, and user interface.

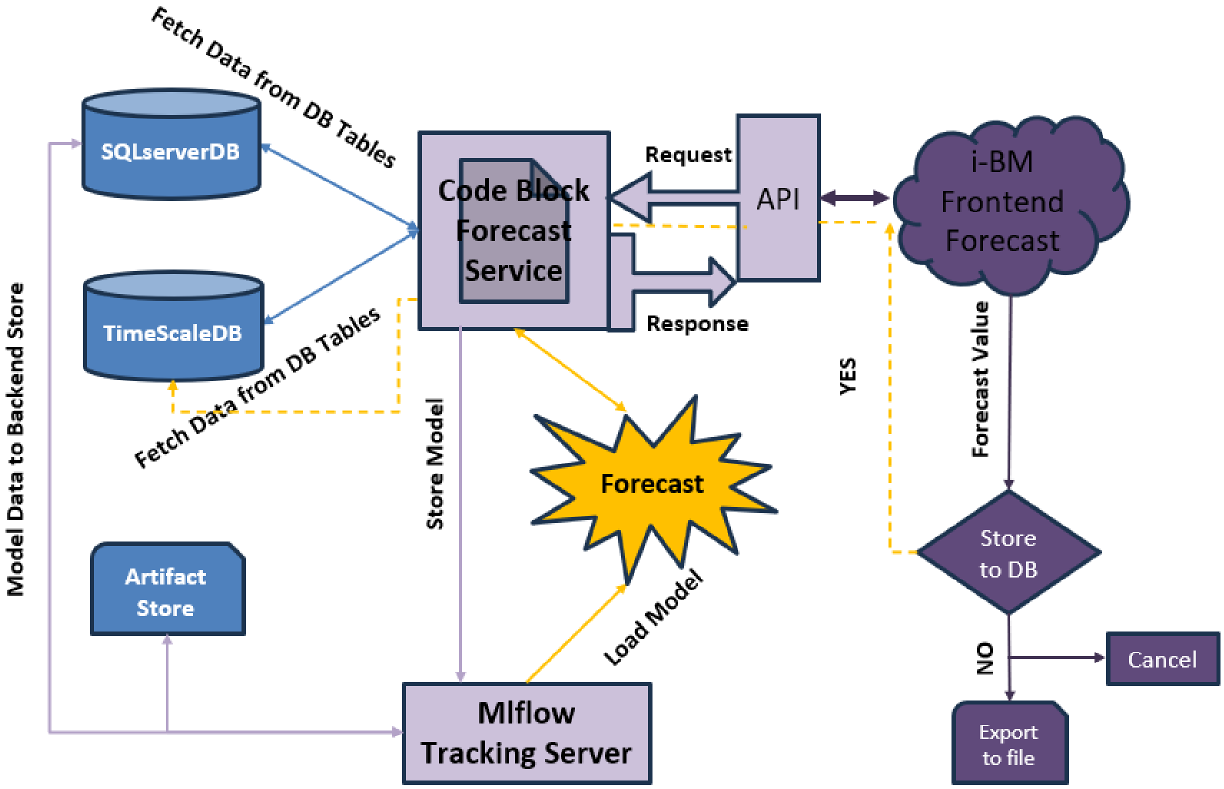


Figure 8. Forecast Service Back-end Architecture

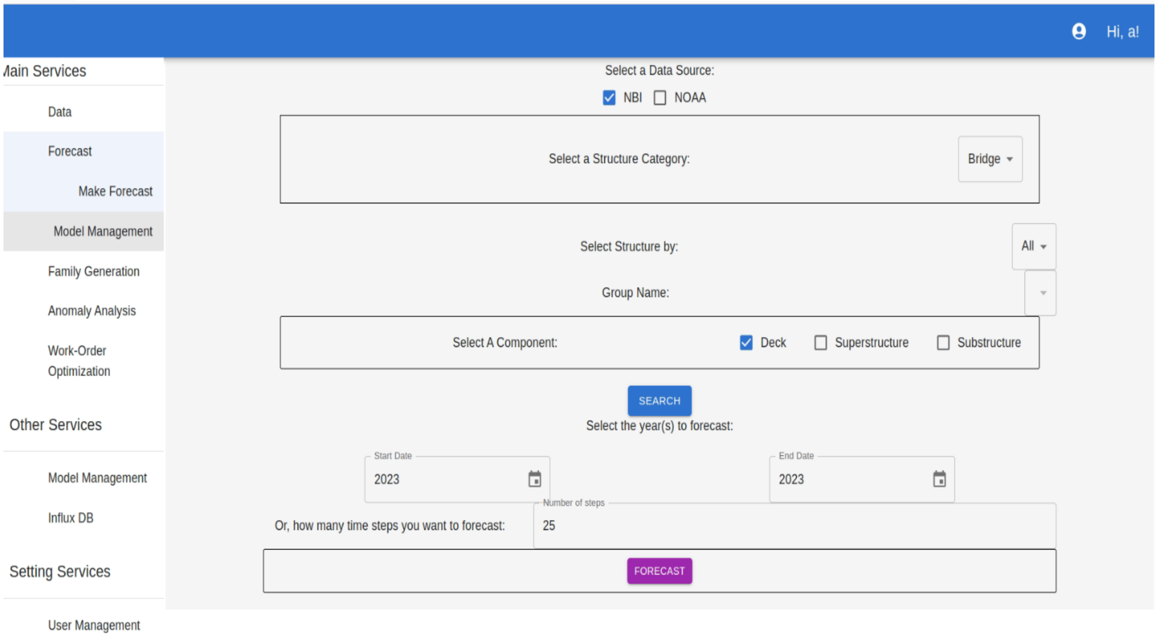
2.3.2 Core Functionalities

The Forecast Module operates as a service-oriented subsystem that connects the backend data sources, machine learning framework, and visualization layer. It integrates all stages of predictive modeling—training, management, execution, and result delivery (e.g., Figure 9 shows the Forecast module of the i-BM tool). The core functionalities include the following:

2.3.2.1 Model Management and Training

- The module handles model training, registration, and version control through MLflow, which tracks configurations, hyperparameters, performance metrics, and model artifacts for each experiment.

- Machine learning and Deep learning algorithms such as LSTM, Bi-LSTM, GRU, CNN, CNN-BiLSTM, Multi-channel CNN, TCN, and Linear Regression, along with their physics-guided extensions (detailed in Chapter 3. Physics-guided Bridge Deterioration Forecasting), are supported for forecasting future bridge element condition ratings—deck, superstructure, substructure, and culvert.
- Users can define training parameters such as input features, training range, model, and Iterations directly through the i-BM interface.
- All trained models are recorded in the backend tables trained_models and registered_models, ensuring complete traceability and reproducibility.
- Each model entry includes metadata such as model_name, data_source, structure_group, run_id, and evaluation metrics (e.g., RMSE), enabling comprehensive version tracking and auditability.



Select a Data Source:
☑ NBI □ NOAA

Select a Structure Category: Bridge

Select Structure by: All

Group Name:

Select A Component: ☑ Deck □ Superstructure □ Substructure

Select the year(s) to forecast:

Start Date 2023

End Date 2023

Or, how many time steps you want to forecast: Number of steps 25

Figure 9. Bridge/Culvert Forecasting Interface in the i-BM Tool

2.3.2.2 Forecast Execution and Result Generation

- The service dynamically loads trained models from the MLflow registry and executes them on user-selected datasets drawn from:
  - ▪SQL Server – for static bridge inventory, geometry, and categorical data, and

    - TimescaleDB – for dynamic, time-series data such as annual inspection ratings, traffic volume, and environmental parameters.
- The forecasting workflow computes multi-year condition predictions for the selected structure category (bridge or culvert) and structural components (deck, superstructure, substructure).
- Users can define forecasting parameters such as groups, models, and prediction horizons directly through the i-BM interface.
- Forecast results can be:
    - Stored in the database for visualization, monitoring, and analysis, or
    - Exported as structured files (CSV/Excel) for documentation and external analysis.

2.4 Anomaly Detection Module

The **Anomaly Detection Module** in the Intelligent Bridge Management (i-BM) system is designed to automatically identify abnormal deterioration patterns and inconsistencies within bridge and culvert condition data. It supports data validation, model reliability assessment, and decision-making by ensuring that forecasting and analysis are performed on accurate and trustworthy datasets.

2.4.1 Anomaly Detection Service Architecture

The anomaly detection framework in the i-BM system is designed based on the label-efficient interactive time series anomaly detection (LEIAD) architecture [59] (Detail in Chapter 4. Bridge Anomaly Detection) that integrates unsupervised anomaly detection (UAD), weak supervision, and active learning. The goal is to reduce manual labeling effort while improving the accuracy of bridge performance anomaly identification. The workflow, illustrated in Figure 10, operates through the following key components:

1. UAD Methods (Unsupervised Anomaly Detection): The process begins with a set of unsupervised algorithms that identify abnormal temporal patterns in bridge condition data without relying on pre-labeled samples. These algorithms (e.g., Isolation Forest, Spectral Residual, STL, RC-Forest, and Luminol) serve as the foundation for generating initial anomaly candidates.

2.Labeling Functions (LF): A set of rule-based or model-based labeling functions is applied to the UAD outputs. Each labeling function encodes heuristic knowledge or statistical criteria (e.g., sudden condition drop, deviation from trend) to automatically assign weak labels to time-series data points.

3.Weak Supervision Layer: The weakly labeled data are aggregated and combined to produce weak labels, which serve as the initial pseudo-ground truth for model training. This allows the model to learn general anomaly patterns without manual annotation.

4.End Model Training: A discriminative anomaly detection model (e.g., LightGBM) is trained on the weakly labeled dataset to distinguish between normal and abnormal deterioration behaviors. The trained model produces refined anomaly scores and binary predictions.

5.Active Learning: Following the initial training, the system identifies samples with high uncertainty or model disagreement and presents them to the user through an interactive labeling interface. This human-in-the-loop feedback mechanism enhances model accuracy where it is most uncertain.

6.Golden Label Generation: Verified user feedback is incorporated to create golden labels—trusted, high-confidence annotations that improve model precision and provide a reliable benchmark for retraining.

7.Label Function Generator (LF Generator): Using insights from the newly generated golden labels, the system automatically creates new or refined labeling functions. These functions are added back into the LF pool, reinforcing the weak supervision process and enabling continual improvement.

8.Iterative Refinement: The framework operates as a closed feedback loop, where new labeling functions and retrained models iteratively enhance anomaly detection accuracy over time, achieving progressive improvement with minimal human supervision.

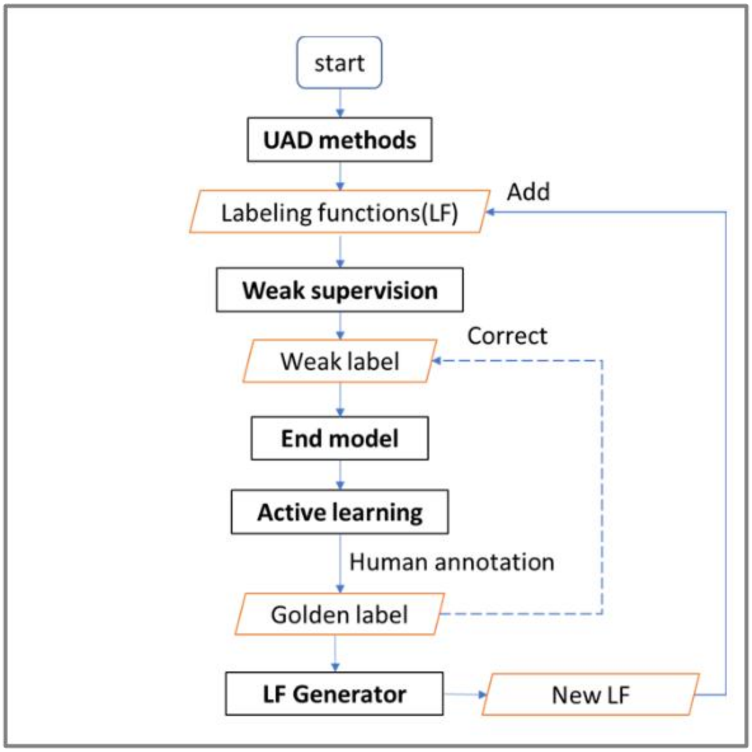


Figure 10. Architecture of the Anomaly Detection Framework [59]

2.4.2 Core Functionalities

The Anomaly Detection Module provides two principal backend modules: **Model Management** and **Detection Execution**.

2.4.2.1 Model Management

- Handles model training, tracking, and lifecycle management through **MLflow**, which records configurations, hyperparameters, thresholds, and performance metrics.
- Each trained model and its run metadata are stored in SQL Server tables (trained_models, registered_models) to ensure full version control, traceability, and reproducibility.
- The model metadata include parameters such as model_name, component, data_source, structure_group, and run_id, ensuring complete transparency and traceability across all detection experiments.

2.4.2.2 Detection Service

- Loads trained models from MLflow and applies them to time-series condition data retrieved from **TimescaleDB** to identify anomalous data points for individual bridge components' time series sequence (Deck, Superstructure, Substructure, and Culvert).
- Supports both bridge and culvert datasets using inspection data from NBI.
- The detection results include yearly anomaly labels (0 = normal (blue), 1 = anomaly (red)) with associated timestamps and structure IDs.
- Results can be either:
  - **Stored** in the database for visualization and analysis, or
  - **Exported** as structured files for external validation and reporting.

2.5 Functional Workflow of the i-BM System

Figure 11 illustrates the functional workflow of the Intelligent Bridge Management (i-BM) system, showing how users interact with different modules such as data management, forecasting, model training, and anomaly detection. The diagram outlines the logical sequence of actions within the application—from data import and exploration to model training, prediction, and anomaly detection—demonstrating how the system integrates analytical tasks through an intuitive, user-centered interface.

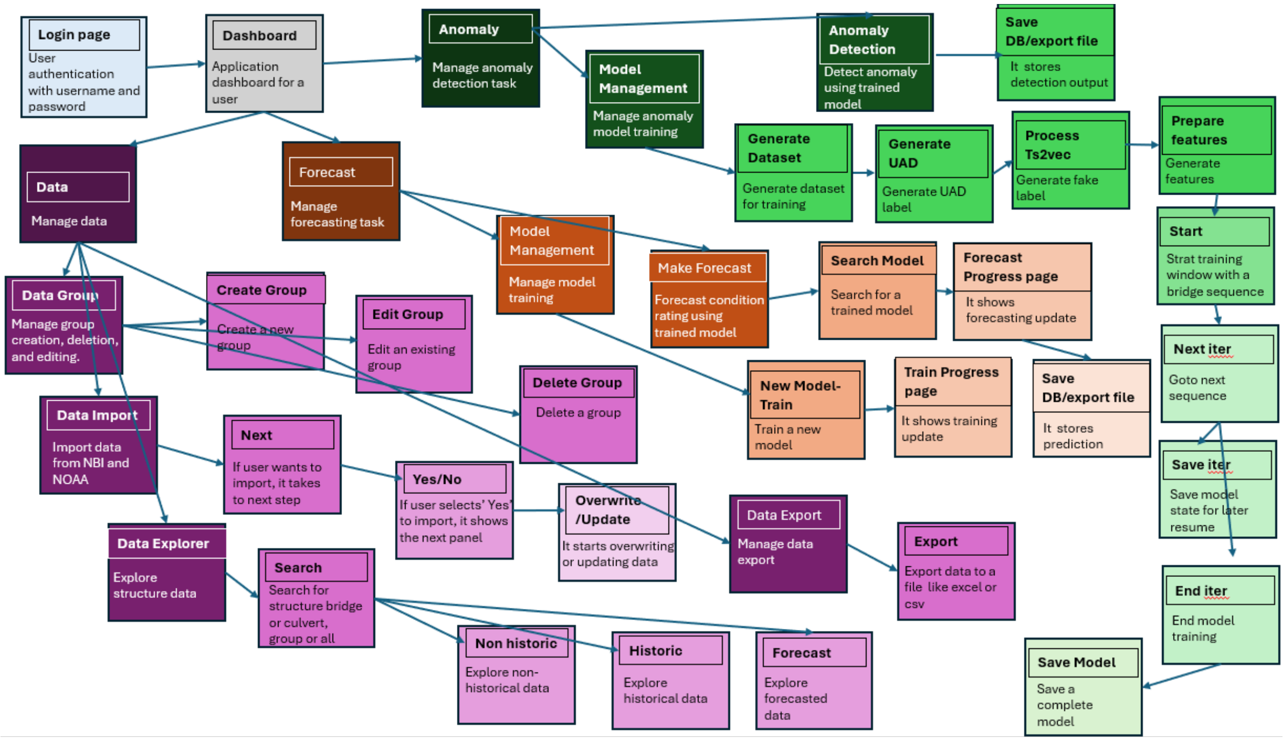


Figure 11. Functional Workflow of the Intelligent Bridge Management (i-BM) System
Different colors highlight key system functionalities: user management (light blue), data exploration, import, and grouping (dark magenta to light pink), model management and forecasting (dark to light orange), and anomaly training and detection (dark to light green) workflows.

Chapter 3. Physics-guided Bridge Deterioration Forecasting

In this section, we delve into physics-guided bridge deterioration forecasting. We begin with the problem definition, followed by a review of relevant literature, methodology, and finally conclude with a comprehensive experimental evaluation of the models and i-BM tool results.

3.1 Problem Definition

The focus of this work is on effective **multivariate time-series forecasting** of bridge condition ratings over time using **physics guidance**. Given a set of consecutive time steps **t**, each of which contains multiple variables of length **m**, the objective is to predict **N** future time steps.

The input for this problem can be represented as:

$$X = \begin{bmatrix} x_{11} & x_{12} & & x_{1n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & & x_{mn} \end{bmatrix},$$

where **n** denotes the total number of input time steps, and **m** represents the total number of variables for each time step. Each variable $\mathbf{X_{ij}}$ corresponds to the condition rating or supplementary feature associated with that time and component. A **multivariate time-series forecasting model**, denoted as $\mathbf{f_{\theta}}$ predicts one future multivariate time step based on the past **n** time steps, where both the input and prediction steps contain **m** features. The input and output variables share the same structure. In addition to this, the model is conditioned using some form of physics-based grounding, which is embedded into the model architecture or training process to incorporate domain-specific physics-based knowledge. To accommodate sequences of varying lengths during training, a **masking layer** with **zero-padding** can be employed within the model. Furthermore, a **sliding-window technique** is used during prediction to generate multiple future time steps, as shown in Figure 12. The figure also demonstrates how variables of interest can be selectively extracted from the predicted output vector.

The success of the forecasting model is evaluated using the **Root Mean Squared Error (RMSE)**, defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}} \qquad (1)$$

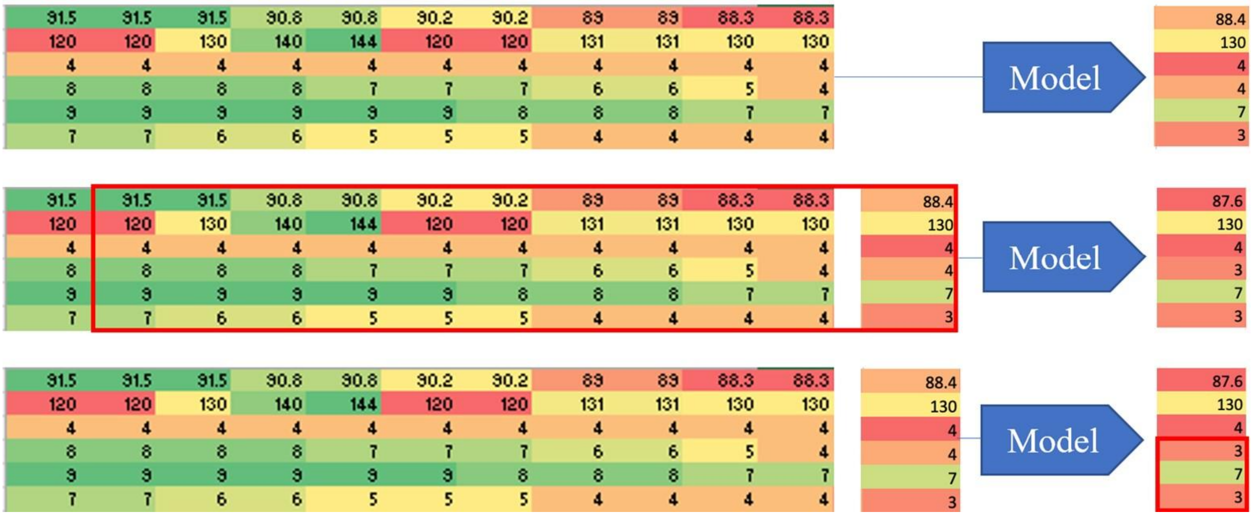
where:

x_i (x sub i) represents the ground truth value,

$\hat{x}_i$ (x hat sub i) denotes the predicted value, and

N is the total number of data points.

The evaluation process measures the error in the produced outputs during the evaluation of the test split of the data after model training. Each sample in the test split of the data is used to obtain an RMSE score for each rating. An average of RMSE scores is taken over all test split samples to obtain an aggregate RMSE measure. An example of this is shown in Figure 13**,** with a test split size of 2215 data samples, 21 total time steps, and 3 variables.



91.5	91.5	91.5	90.8	90.8	90.2	90.2	89	89	88.3	88.3
120	120	130	140	144	120	120	131	131	130	130
4	4	4	4	4	4	4	4	4	4	4
8	8	8	8	7	7	7	6	6	5	4
9	9	9	9	9	9	8	8	8	7	7
7	7	6	6	5	5	5	4	4	4	4

88.4
130
4
4
7
3

91.5	91.5	91.5	90.8	90.8	90.2	90.2	89	89	88.3	88.3	88.4
120	120	130	140	144	120	120	131	131	130	130	130
4	4	4	4	4	4	4	4	4	4	4	4
8	8	8	8	7	7	7	6	6	5	4	4
9	9	9	9	9	9	8	8	8	7	7	7
7	7	6	6	5	5	5	4	4	4	4	3

87.6
130
4
3
7
3

91.5	91.5	91.5	90.8	90.8	90.2	90.2	89	89	88.3	88.3	88.4
120	120	130	140	144	120	120	131	131	130	130	130
4	4	4	4	4	4	4	4	4	4	4	4
8	8	8	8	7	7	7	6	6	5	4	4
9	9	9	9	9	9	8	8	8	7	7	7
7	7	6	6	5	5	5	4	4	4	4	3

87.6
130
4
3
7
3

Figure 12. Illustration of Model Prediction Process

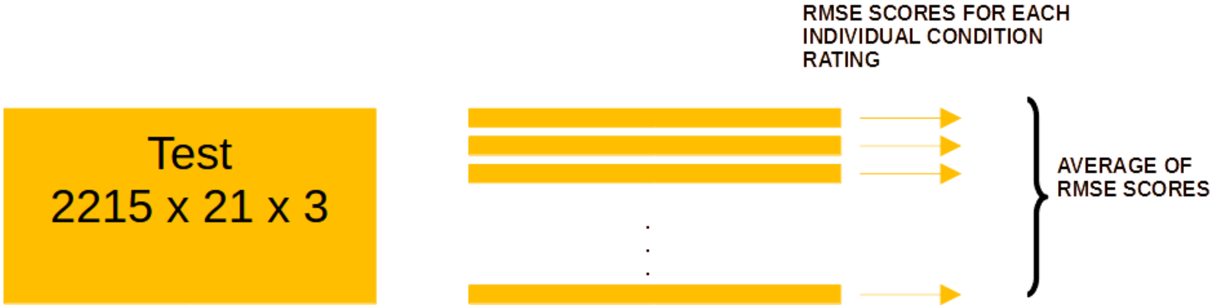


Figure 13. RMSE Measurement Process

To reiterate, we wish to train a multivariate time-series forecasting model $\mathbf{f_\theta}$ such that it generates a single time step output with length **m** representing the number of variables in the output time step. This model is to be conditioned on the input matrix **X,** where its dimensions are **n** and **m,** where **n** is the total number of input time steps, and **m** is the number of variables in each time step. This model also needs to be conditioned using some form of physical grounding via the physics-informed neural network approaches. The success of this model will be evaluated using RMSE as the metric.

3.2 Literature Review

In this section, we review the literature on bridge and culvert deterioration forecasting. In the current literature, existing deterioration forecasting models are either mechanistic or data-driven, such as deterministic, stochastic, or artificial intelligence (AI)-based. Toward this end, we categorize the literature into three key areas: (1) physics-based methods which account for mechanistic insights (2) data-driven approaches, which leverage statistical, artificial intelligence (AI), machine learning (ML), and deep learning (DL) techniques for predictive modeling, and (3) physics-guided neural network, which are hybrid solutions that integrate engineering principles and physics-based parameters to enhance model accuracy. The following subsections discuss these approaches in detail.

3.2.1 Physics-based Bridge Deterioration Forecasting

Physics-based approaches represent traditional systematic efforts to forecast bridge deterioration by modeling the material and environmental mechanisms that drive structural degradation. Before the emergence of data-driven methods, researchers developed mechanistic formulations that mathematically describe deterioration as a function of material properties, stress conditions, and exposure environments. These models—also referred to as mechanistic or simulation-based—derive from engineering mechanics and material science principles, focusing primarily on corrosion-induced degradation of reinforced concrete (RC) bridge decks.

One of the earliest mechanistic frameworks was proposed by Tuutti [60], who divided the corrosion process into two distinct stages: initiation, governed by chloride ingress and carbonation, and propagation, driven by electrochemical reactions and cracking. This conceptual model became the foundation for later service-life prediction tools such as Life-365.

The Life-365 Service Life Prediction Model [61–62] introduced a diffusion-based approach for estimating the time to corrosion initiation in RC structures exposed to deicing salts or marine environments. By utilizing measurable parameters such as chloride diffusion coefficient, surface concentration, and concrete cover depth, the model provided engineers with a practical means to forecast deterioration over a structure's life cycle. Its integration into engineering practice and industry standards marked a major advancement in durability design for reinforced concrete bridges.

Building on this foundation, Hu et al. [63] reviewed several prominent commercial mechanistic modeling tools—STADIUM, CONCLIFE, and Life-365—each offering distinct physical formulations for predicting deterioration. STADIUM and Life-365 primarily simulate chloride-induced corrosion using ion-diffusion models, while CONCLIFE extends its analysis to sulfate attack and freeze–thaw damage. These frameworks collectively illustrate the state of practice in physics-based durability modeling for RC bridge systems. Although these models have demonstrated substantial predictive power, they are generally limited by simplified assumptions, reliance on laboratory-calibrated parameters, and computational complexity when applied at the network scale.

Several analytical studies have also contributed to understanding specific deterioration mechanisms in reinforced concrete. Bažant et al. [64] developed a model to describe freeze–thaw damage in concrete, while Isgor and Razaqpur [65] formulated a carbonation-based corrosion model linking CO_2 diffusion and environmental exposure to steel-corrosion initiation. Bažant and Baweja [66] further extended this analytical work to include creep and shrinkage models, enhancing understanding of time-dependent concrete behavior. Collectively, these analytical formulations form the theoretical backbone for many of the deterioration mechanisms represented in later service-life prediction models.

A key contribution to corrosion modeling came from Liu and Weyers [67], whose experimental study established relationships between corrosion rate, temperature, chloride concentration, concrete resistivity, and exposure duration. Balafas and Burgoyne [68] subsequently developed a mathematical model to predict internal pressure buildup from corrosion products leading to concrete cover cracking. These submodels were later incorporated into the comprehensive time-to-failure framework developed by Hu et al. [63] for RC bridge decks, which captured both corrosion and carbonation processes under realistic environmental and design conditions. While this model provided an important step toward integrating mechanistic modeling in bridge-asset management, it still omitted several modern aspects of bridge design and maintenance practice.

To address these constraints, Nickless [54] conducted one of the most comprehensive applications of mechanistic modeling for predicting bridge deterioration. The study, carried out for the Office of Applied Research(OAR) of the Colorado Department of Transportation (CDOT), developed a multi-stage mechanistic model for corrosion-induced cracking in reinforced concrete bridge decks. The model estimated the time to corrosion initiation, cracking onset, and crack propagation by integrating sub-models for corrosion rate, concrete resistivity, and cracking pressure. It also considered practical bridge-specific factors such as epoxy-coated rebar, waterproofing membranes, asphalt overlays, joint deterioration, and deck maintenance. By combining laboratory-based parameterized formulations (see Figure 14) with realistic environmental data, the study demonstrated how physics-based deterioration modeling can guide bridge design and maintenance planning. However, these models often face development complexities and are typically focused on straightforward result generation rather than accounting for actual structural conditions, which are susceptible to environmental noise and estimation errors. For this reason, they are generally considered more appropriate for project-level analysis or as complementary tools alongside other deterioration-modeling approaches, for example, serving as supportive components within data-driven or ML/DL models.

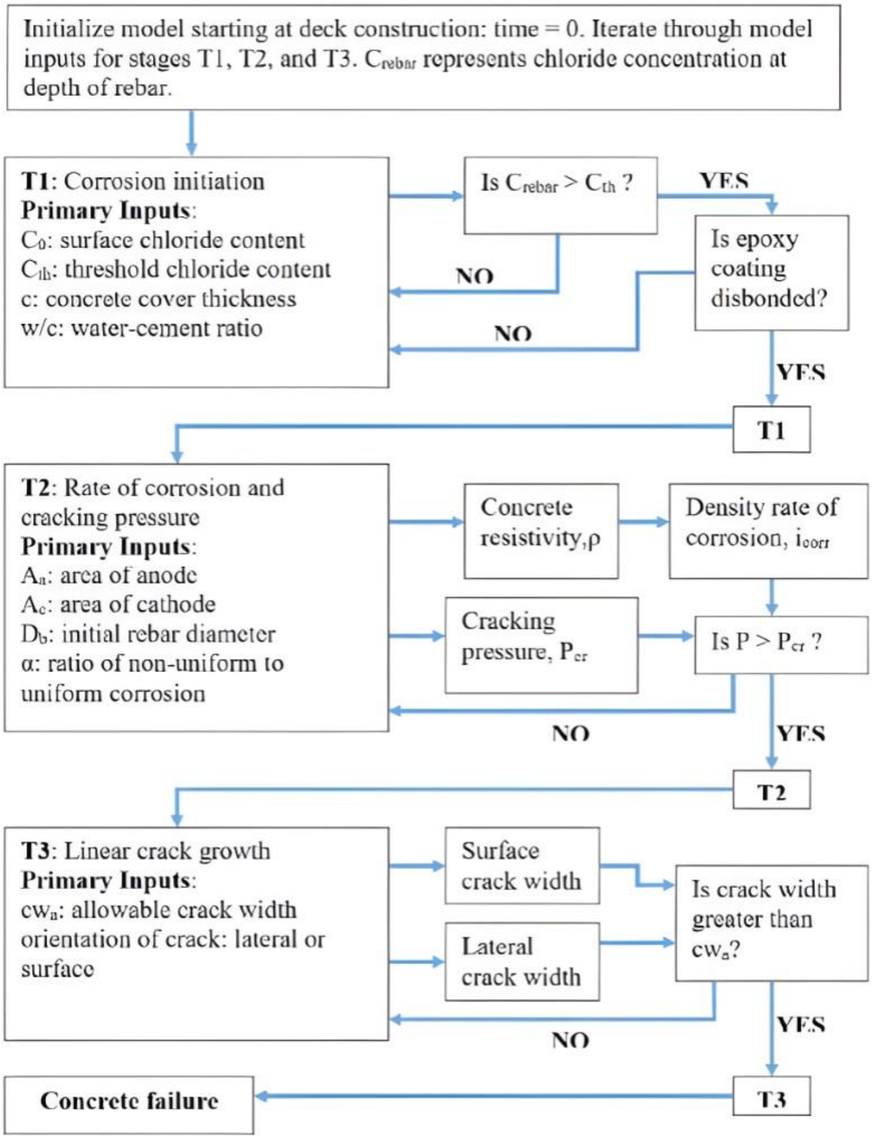

Initialize model starting at deck construction: time = 0. Iterate through model inputs for stages T1, T2, and T3. C_rebar represents chloride concentration at depth of rebar.
T1: Corrosion initiation
Primary Inputs:
C0: surface chloride content
Cth: threshold chloride content
c: concrete cover thickness
w/c: water-cement ratio
Is Crebar > Cth ?
YES
NO
Is epoxy coating disbonded?
NO
YES
T1
T2: Rate of corrosion and cracking pressure
Primary Inputs:
Aa: area of anode
Ac: area of cathode
Db: initial rebar diameter
α: ratio of non-uniform to uniform corrosion
Concrete resistivity,ρ
Density rate of corrosion, icorr
Cracking pressure, Pcr
Is P > Pcr ?
NO
YES
T2
T3: Linear crack growth
Primary Inputs:
cwa: allowable crack width
orientation of crack: lateral or surface
Surface crack width
Lateral crack width
Is crack width greater than cwa?
NO
YES
T3
Concrete failure

Figure 14. Multi-Level Model Process for Predicting Concrete Failure [53]

3.2.2 Data-Driven Bridge Deterioration Forecasting

Over the past three decades, bridge deterioration forecasting has shifted from traditional mechanistic approaches to increasingly data-driven and learning-based models. A wide range of deterministic, stochastic, and AI-based methods have been explored, from early research stages to recent years. Deterministic models assume that the tendency of bridge deterioration processes is certain and are based on regression analysis of condition data [69–71]. Stochastic models consider the bridge deterioration process as one or more random variables [72] and employ probabilistic techniques such as the Markov-chain model to capture uncertainty [73–76]. However, deterministic and stochastic models often fail to include all of the influential factors directly, except for dependence on engineering judgments and assumptions.

In this regard, researchers have focused on demonstrating that artificial intelligence (AI) techniques can mimic deterioration trends directly from inspection data, moving beyond rigid statistical formulations. The first use of Artificial Neural Networks (ANN) in bridge deterioration forecasting was explored by Sobanjo [77]. Early AI models demonstrated the potential of data-driven learning to capture nonlinear deterioration patterns. Building on this idea, Tokdemir et al. [78] compared ANN and Genetic Algorithms (GA) for predicting bridge sufficiency ratings using explanatory variables such as geometrical attributes, structure age, traffic volume, and structural attributes. This work found that ANNs outperformed genetic algorithms when different models were constructed for varying levels of sufficiency ratings, and genetic algorithms outperformed ANNs when using the entire dataset. Morcous [79] further compared ANNs with case-based reasoning for predicting future bridge conditions. The study highlighted that case-based reasoning achieved high accuracy but was difficult to calibrate and ineffective for unseen data, while ANNs were more adaptable but required greater initial effort for model development and updates.

While the artificial intelligence–based models, particularly neural networks, show strong potential to overcome the limitations of existing methods, their application in Bridge Management Systems (BMS) remains in the nascent stage [80]. These foundational efforts defined the promise of data-driven learning but also exposed key barriers—small datasets, high tuning effort, and weak temporal reasoning. Bridge and culvert deterioration forecasting is challenging because the deterioration process is influenced by multiple factors such as materials, design, daily traffic, freeze and thaw cycles, and climate conditions [25-26, 81]. Although transportation agencies collect diverse datasets—such as NBI and NBE records, structural attributes, traffic volumes, and weather data—these data sources are often fragmented and inconsistent, making comprehensive modeling difficult. Structured bridge inventory data, particularly the NBI dataset, remains the primary data source used in most existing studies on bridge deterioration forecasting. Researchers began to test a range of statistical and data mining models—linear regression, decision trees, and neural network algorithms—to identify the most influential features in NBI inspection records, thereby improving predictive ability. Studies such as Contreras-Nieto et al. [82] and Jonnalagadda et al. [83] demonstrated that machine learning can analyze the effects of factors and capture regional deterioration trends for steel and concrete bridges. Mia and Kameshwar [84] further showed that Random-Forest-based ensembles could produce highly accurate short-term forecasts with quantified uncertainty, while Rashidi and Elzarka [85] revealed that feature optimization plays a decisive role in prediction quality.

Regardless of these advances, most machine learning models remain constrained by region-specific datasets, short-term prediction horizons, and oracle-identified or a limited set of feature representations, as well as applicability restricted to specific bridge types. That hinders their ability to generalize or capture nonlinear and temporal deterioration behaviors. Recent studies have focused on Deep Learning architectures capable of capturing spatial and temporal dependencies in bridge deterioration. Liu et al. [86, 87] developed deep learning–based approaches for forecasting bridge component conditions. In two back-to-back studies, they showcased CNN-based deterioration forecasting that incorporated historical condition data and uncertainty quantification through stochastic Markov integration. Subsequent efforts by Zhu and Wang [88] and Rajkumar et al. [89] demonstrated that hybrid models—combining CNN-RNN architectures and autoencoders with random forest (RF) algorithms—can effectively capture sequential deterioration dynamics with improved performance. These works broadly represent a transition from static to spatiotemporal modeling, where deterioration is treated as a continuous sequence influenced by both structure and environment. In addition, recent works by Jing et al. [90], Miao et al. [91], and Abu Dabous et al. [92] have further advanced this direction by employing other sequential architectures integrating with LSTMs to capture long-term temporal dependencies and sequential deterioration patterns in bridge components with accuracies exceeding 90% for near-term forecasts. Despite the importance of dealing with advanced deep learning architectures and achieving promising results, limitations such as reliance on single-source data, focus on individual components, exclusion of maintenance effects, and insufficient consideration of environmental, spatial, and temporal dependencies are limiting the ability of existing studies [88–92] to be effectively used in long-term bridge deterioration forecasting.

Beyond single-source sequential deep learning models, Liu and El-Gohary [86] proposed a multisource deep learning framework that integrates both structured and unstructured bridge data using a recurrent neural network enhanced with manifold and cost-sensitive learning. While the approach effectively leveraged heterogeneous data to improve deterioration prediction accuracy, its unidirectional architecture predicted only one component type at a time, limiting scalability and bidirectional temporal learning capacity.

Gleaned from the literature, it is evident that DL-based models remain highly data-dependent, requiring comprehensive and high-quality multisource datasets, reliable feature selection, and explicit treatment of repair events to achieve robust predictive performance and overcome existing limitations. Moreover, they often lack explicit physics-based learning and transferability across bridge types and deterioration mechanisms, motivating the integration of domain knowledge into data-driven modeling.

3.2.3 Physics-Guided Neural Networks

In recent years, in other domains, physics-guided approaches have been introduced to enhance the learning capabilities of ML/DL models by integrating physics-based principles, thereby reducing dependence on purely historical data. To the best of our knowledge, this research direction has not yet been explored within the field of bridge deterioration forecasting. Toward this end, we examine the emerging domain of Physics-Guided Neural Networks (PGNNs), an interdisciplinary framework that fuses the computational power of neural networks with mechanistic modeling grounded in physics-based laws and domain knowledge to improve model intelligibility and predictive performance.

A key area within this field is the design and application of physics-guided loss functions, which incorporate behavioral physics laws directly into neural network training. These loss functions improve the predictive accuracy and generalization capabilities of the neural networks by aligning the learning objectives with known physics-based principles. For instance, Huang et al. [56] introduced a Physics-Guided Deep Neural Network (PGDNN) for structural damage identification by integrating finite element (FE) model outputs with measured vibration data through a cross-domain physics-based loss function. This approach significantly improved damage localization accuracy and robustness against modeling noise. Similarly, Yousefpour and Wang [58] developed Scour Physics-Inspired Neural Networks (SPINNs), a hybrid framework that couples empirical scour equations with LSTM and CNN architectures. Their models reduced prediction errors by up to 70% compared to purely data-driven methods, highlighting the value of incorporating governing physics into learning. However, none of these studies directly address bridge or culvert deterioration forecasting, leaving a clear gap in the current research scope. Building on the identified research gap and inspired by the demonstrated success of Physics-Guided Neural Networks (PGNNs), we explore the landscape of this research domain and provide a taxonomy of the three primary areas of focus within physics-guided neural networks: (1) Physics-guided loss functions, (2) Physics-guided architecture, (3) Extended physics-guided machine learning [93]. The following subsection reviews the focus areas in detail.

3.2.3.1 Physics-Guided Loss Functions

One widely used approach for implementing physics-guided neural networks (PGNNs) is to introduce an additional term into the loss function of a data-driven model. This term incorporates a physics-based penalty, guiding the deep learning model to learn patterns that are consistent with established physical laws and normal system behavior [93].

The general form of a physics-guided loss function can be expressed as:

$$\arg\ \min_{f} \left[\underbrace{\mathrm{Loss}(\hat{\mathrm{Y}}, \mathrm{Y})}_{\text{Empirical Error}} + \underbrace{\lambda \mathrm{R}(\mathrm{f})}_{\text{Structural Error}} + \underbrace{\lambda_{\mathrm{PHY}}\ \mathrm{Loss.PHY}(\hat{\mathrm{Y}})}_{\text{Physical Inconsistency}} \right] \tag{2}$$

Here:

The **Empirical Error** represents the difference between predicted and observed values.

The **Structural Error** regularizes the model to prevent overfitting.

The **Physical Inconsistency** term penalizes outputs that violate known physical relationships.

Karpatne et al. [57] demonstrated this principle by leveraging the relationship between lake depth, temperature, and density to ensure that the model predictions adhere to known physical laws. They enforced a penalty whenever predicted water density values violated the physical constraint that density increases with depth.

The physical relationship is expressed as:

$$\rho[\mathrm{d}_1,\mathrm{t}] - \rho[\mathrm{d}_2,\mathrm{t}] \leq 0 \text{ if } \mathrm{d}_1 < \mathrm{d}_2 \tag{3}$$

For any pair of consecutive depth values d_i and d_{i+1}, where $d_i < d_{i+1}$, the model computes the difference between predicted densities as:

$$\Delta[\mathrm{d}_\mathrm{i},\mathrm{t}] = \hat{\rho}[\mathrm{d}_\mathrm{i},\mathrm{t}] - \hat{\rho}[\mathrm{d}_{\mathrm{i}+1},\mathrm{t}] \tag{4}$$

A positive value of $\Delta[d_i, t]$ represents a violation of the above constraint.
To penalize such violations, a rectified linear function (ReLU) is applied:

$$\mathrm{ReLU}(\Delta[\mathrm{d}_\mathrm{i},\mathrm{t}]) = \max(0, \Delta[\mathrm{d}_\mathrm{i},\mathrm{t}]) \tag{5}$$

The total physics-based loss across all samples and time steps is then formulated as:

$$\mathrm{Loss.PHY}(\hat{\mathrm{Y}}) = \frac{1}{\mathrm{n_t}(\mathrm{n_d}-1)} \sum_{\mathrm{t}=1}^{\mathrm{n_t}} \sum_{\mathrm{i}=1}^{\mathrm{n_d}-1} \mathrm{ReLU}(\Delta[\mathrm{d}_\mathrm{i},\mathrm{t}]) \tag{6}$$

This additional loss term ensures that the network learns outputs that obey physical consistency while minimizing empirical error. The approach has been shown to improve both interpretability and robustness in lake temperature modeling tasks.

3.2.3.2 Physics-Guided Architecture

Due to the modularity of neural networks, these architectures can be customized to encode physical properties directly into their structure. This allows for the design of physics-guided neural network architectures that can impose hard constraints, whereas physics-guided loss functions generally impose soft constraints.

An example of such an architecture is the Turbulent-Flow Net proposed by Wang et al. [94], a physics-guided neural network developed for turbulent flow prediction. The model is inspired by the RANS–LES coupling method but replaces fixed spectral filters with trainable convolutional layers. The turbulent flow input is decomposed into three components, each processed by a specialized convolutional U-Net to preserve multiscale flow features. A shared decoder learns interactions among these components to produce the final prediction.

While architectures like Turbulent-Flow Net have demonstrated strong performance in fluid dynamics, their use in bridge deterioration forecasting remains limited. This gap underscores the need for domain-specific adaptations of physics-guided architectures in structural engineering, especially for long-term condition prediction under physical constraints.

3.2.3.3 Extended Physics-Guided Machine Learning

Extended physics-guided machine learning leverages both pure physics-based models as well as pure deep learning models by developing them independently and combining them separately (see Figure 15).

In essence, we have a neural network model which takes as input data D and outputs Y (f_{NN}: D → Y) and a physics model which does the same (f_{PHY}: D → Y). In this case of extended physics-guided machine learning, the output from f_{PHY} is used as an additional feature input to f_{NN} in addition to the original data input D. This is formalized as, f_{HPD}: $X = [D, Y_{PHY}] \rightarrow \hat{Y}$

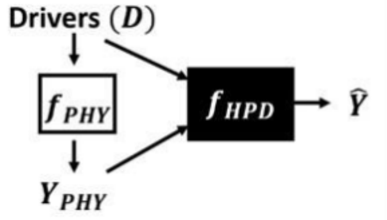


Figure 15. High-Level Diagram of Extended Physics-Guided Machine Learning [57]

DeepGLEAM is a piece of work that focuses on a type of extended modeling called residual learning, where a neural network learns to predict the errors or residuals made by pure physics-based models [95]. This work uses a mechanistic epidemic simulation model, Global Epidemic and Mobility Model (GLEAM), with deep learning. It uses a Diffusion Convolutional Recurrent Neural Network (DCRNN) to learn the correction terms from GLEAM, leading to improved performance.

3.3 Methodology

In this section, we present the proposed methodology for bridge deterioration forecasting using a physics-guided deep learning framework. The overall approach combines data-driven learning with domain knowledge of structural behavior to enhance the interpretability and physical consistency of the predictions. The methodology integrates three major components: (1) the development of Physics-Guided Neural Networks (PGNNs) that embed physical constraints through physics-guided loss functions, (2) Input selection for the deep learning models using feature importance analysis and (3) a Repair-Agnostic Forecasting Framework designed to separate normal deterioration trends from maintenance-induced anomalies in bridge condition ratings. Together, these components provide a robust and generalizable strategy for forecasting structural performance while maintaining fidelity to real-world physical behavior.

3.3.1 Physics-Guided Neural Networks (PGNN) for Bridge Deterioration Forecasting

In this section, we present physics-guided neural networks for bridge deterioration forecasting. The key concept of the Physics-Guided Neural Network (PGNN) builds upon prior work by Karpatne et al. [57] as explained in Section 3.3.2.3, which integrates domain physics with data-driven modeling. To incorporate bridge-specific domain knowledge into the physics-guided modeling, the foundational idea in this study is derived from the mechanistic deterioration modeling approach proposed by Nickless et al. [54], which simulated bridge deck deterioration using physics-based equations that describe chloride diffusion and adhesion loss. While the original model was purely physics-driven and did not incorporate machine learning, its simulation framework provides the conceptual basis for generating physics-informed features, as shown in Figure 16.

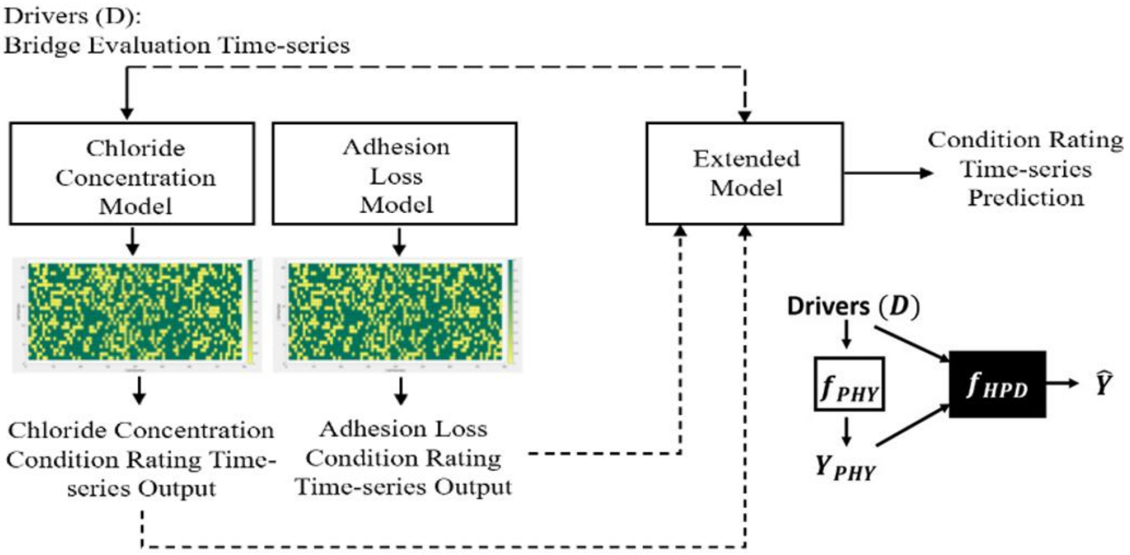

Drivers (D):
Bridge Evaluation Time-series
Chloride Concentration Model
Adhesion Loss Model
Extended Model
Condition Rating Time-series Prediction
Chloride Concentration Condition Rating Time-series Output
Adhesion Loss Condition Rating Time-series Output
Drivers (D)
f_{PHY}
Y_{PHY}
f_{HPD}
$\hat{Y}$

Figure 16. Hybrid physics-guided deep learning forecast model

In our work, this foundation is adapted within a physics-guided learning context, where physical consistency is introduced into the machine learning process through a physics-guided loss function that minimizes empirical loss during model training. The following subsection focuses on the implementation of this physics-guided loss function to enhance physical consistency in the deterioration patterns of prediction.

3.3.1.1 Physics-Guided Loss Functions in Bridge Deterioration Forecasting

The physics-guided loss function used in this work will be detailed in this section. First, a formalization of the idea will be presented. Following the formalization, the physics property of interest as well as how it was incorporated into the loss function of our neural networks will be examined.

3.3.1.1.1 Formalization

The goal of introducing a physics-guided loss term into the learning objective is to determine whether including a physically consistent penalty improves prediction accuracy for deterioration forecasting. Specifically, the objective is to assess whether embedding a physical constraint directly in the optimization function enables the model to generate more realistic deterioration forecasts.

Recall, the general structure of a physics-guided loss function is expressed as:

$$\arg \min_{f} [\underbrace{\text{Loss}(\widehat{Y}, Y)}_{\text{Empirical Error}} + \underbrace{\lambda R(f)}_{\text{Structural Error}} + \underbrace{\lambda_{PHY} \text{Loss.PHY}(\widehat{Y})}_{\text{Physical Inconsistency}}] \tag{7}$$

In the case of this work, only an empirical error and physical inconsistency were used from (6). For the empirical error, Mean Squared Error (MSE) was used, and for the physical inconsistency, a relationship pertaining to the ground truth and predictions for three condition ratings in the bridge evaluation data: Deck Condition, Superstructure Condition, and Substructure Condition, as shown in (8).

$$\text{ReLU}(x) = \max(0, x) \tag{8}$$

3.3.1.1.2 Physics Property in Loss Function

First, we introduce some notation:

$$\hat{y} = [\hat{y}_{\text{deck}}, \hat{y}_{\text{superstructure}}, \hat{y}_{\text{substructure}}] (\text{predicted values}) \tag{9}$$

$$y = [y_{\text{deck}}, y_{\text{superstructure}}, y_{\text{substructure}}] (\text{ground truth values}) \tag{10}$$

We formulate the relationship between (9) and (10) as described in the previous section (see equation 11):

$$\Delta[\hat{y}, y] = \hat{y} - y \tag{11}$$

A positive value of Δ[ŷ, y] indicates a violation of the physical constraint, as it implies that the predicted rating ŷ exceeds the true (observed) rating y. The result of this function can be fed into a ReLU (Rectified Linear Unit) as such, to capture the penalty. ReLU (Rectified Linear Unit) is a simple, fast, and well-worked function to train deep neural networks. As shown in equation (12), this violation is penalized using the function:

$$\text{ReLU}(\Delta[\hat{y}, y]) = \max(0, \Delta[\hat{y}, y]) \tag{12}$$

The physics-based loss we incorporate into our loss function, in addition to Mean Squared Error (MSE) is the following, where N represents the total number of samples. Note that the loss is calculated across all samples to obtain an aggregate calculation of loss for the model:

$$\text{Loss.PHY}(\hat{y}) = \frac{1}{N}\sum_{i=1}^{N} \text{ReLU}(\Delta[\hat{y}_i, y_i]) \tag{13}$$

The final loss, including both empirical loss and physical inconsistency, which is optimized using backpropagation, is as follows:

$$\text{Loss}(\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \frac{1}{N}\sum_{i=1}^{N} \text{ReLU}(\Delta[\hat{y}_i, y_i]) \tag{14}$$

3.3.1.2 Physics-Guided Neural Network Specifications

This subsection presents the neural network architecture employed in the physics-guided framework. A total of eight models were implemented and trained using the physics-informed loss function:

- Convolutional Neural Network (CNN)
- Temporal Convolutional Network (TCN)
- Long Short-Term Memory (LSTM)
- Bidirectional LSTM (BiLSTM)
- Gated Recurrent Unit (GRU)
- CNN–BiLSTM hybrid
- Multi-channel CNN
- Linear regression model

These architectures were selected to capture both spatial and temporal patterns in the bridge condition data. While all models benefit from the integration of the physics-guided loss function, their internal mechanisms differ in their ability to learn long-term deterioration patterns.

3.3.1.3 Physics-Guided Model Input Selection Using Feature Importance Analysis

Before directly diving into the deep-learning and physics-guided deterioration forecasting models training, a detailed feature importance analysis (FIA) was performed to identify the most relevant predictors for each structural component—deck, superstructure, substructure, and culvert. This preparatory step ensured that model inputs reflected the most physically meaningful and statistically significant variables derived from the National Bridge Inventory (NBI) and National Oceanic and Atmospheric Administration (NOAA) datasets.

We employed the SHapley Additive exPlanations (SHAP) methodology to quantify the relative influence of input features on the model output. SHAP, a model-agnostic and game-theoretic framework, assigns each feature a Shapley value representing its marginal contribution to the model's prediction. By averaging these contributions across all possible feature combinations, SHAP provides fair, consistent, and interpretable importance scores.

For bridge-specific modeling, SHAP values were computed for 37 domain-informed features for the deck, superstructure, and substructure components and 26 features for the culvert models. A threshold was applied to filter the most impactful features for each structure type. Features exceeding this threshold—such as Structural Evaluation, Approach Road Evaluation, Channel Condition, Traffic Lanes On, and Design Load—were retained as the final model inputs for subsequent training.

This systematic feature-selection process reduced redundancy, improved computational efficiency, and ensured that the models captured the dominant physical drivers of bridge deterioration. The effectiveness of SHAP-based feature selection and its quantitative impact on model performance are further analyzed in Section 3.4 (Experimental Evaluation) in ablation analysis.

Component-Wise Feature Importance

Figures 17 through 20 visualize the top-ranked features for each component. These summary plots reveal the most influential predictors associated with deterioration patterns. Several common features—such as structural evaluations, approach road evaluations, and geometric attributes—appear across multiple components, indicating their remarkable impact on the structural condition ratings prediction.

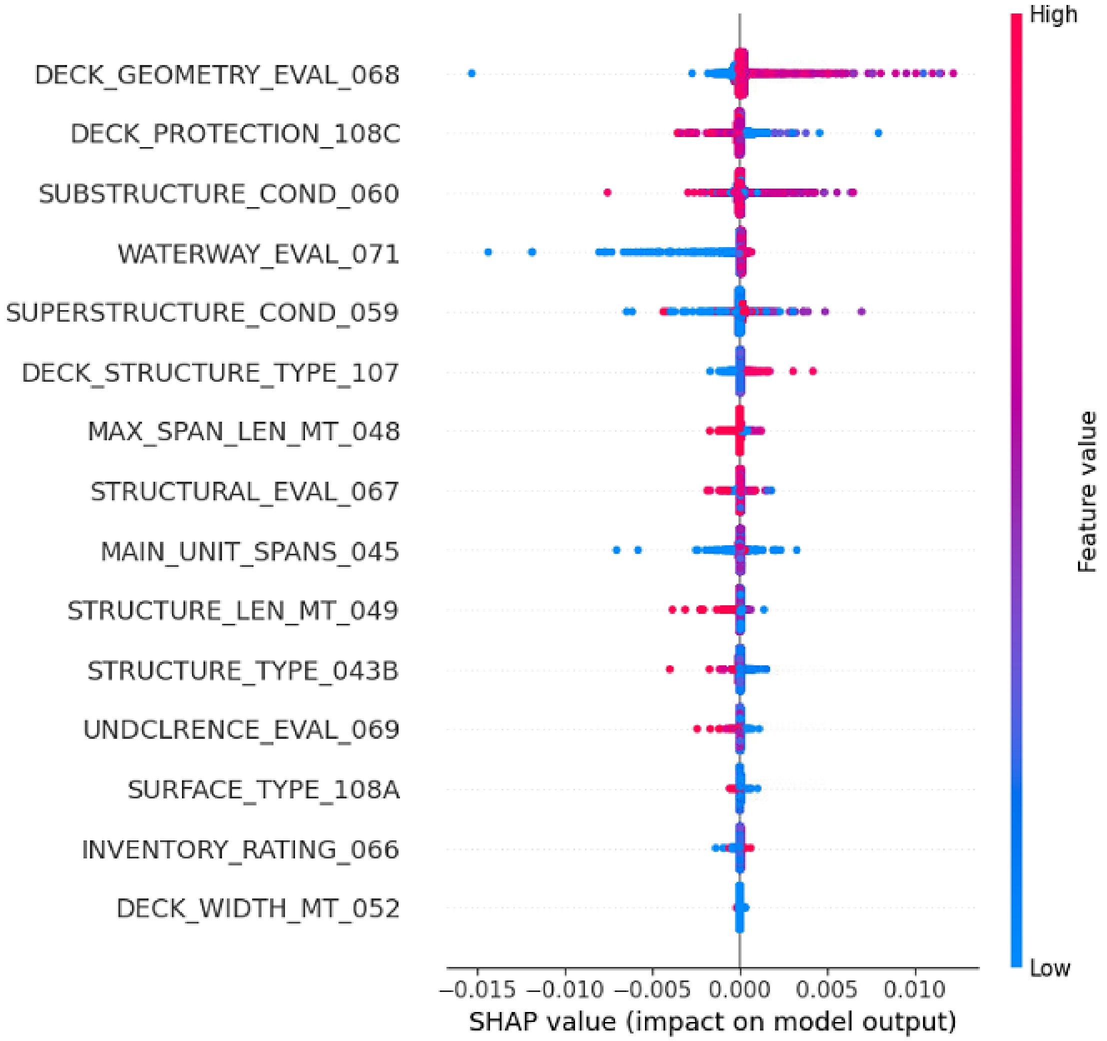


Figure 17. SHAP-based feature importance plot showing the relative influence of input variables on deck condition prediction

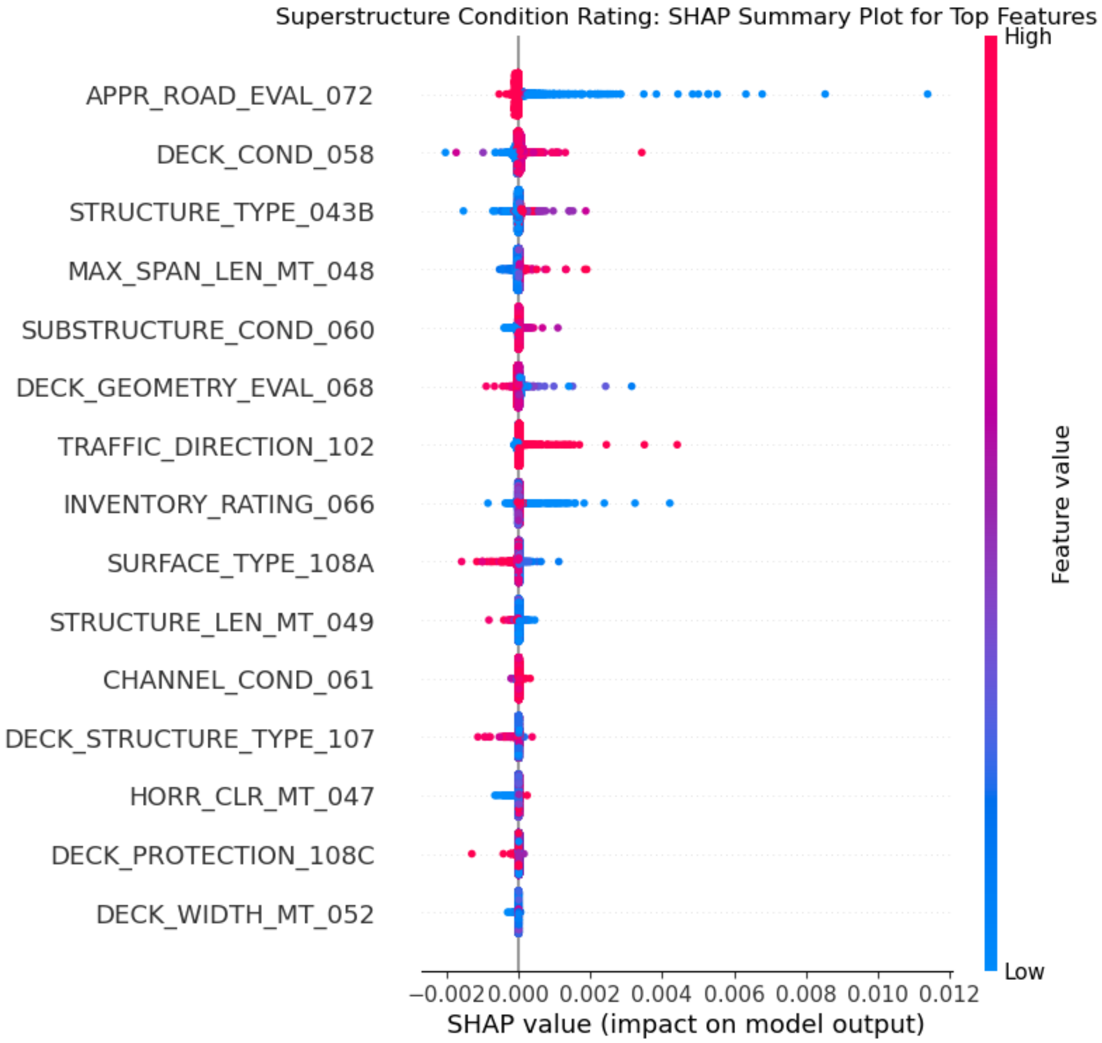


Figure 18. SHAP-based feature importance plot showing the relative influence of input variables on superstructure condition prediction

To streamline the modeling process, we applied a SHAP value threshold of 0.002 for the deck, superstructure, and substructure models, and 0.04 for the culvert model, to identify the most impactful features for each structure type. For instance, in the case of the culvert component (Figure 18), the most influential predictors—Structural Evaluation (067), Approach Road Evaluation (072), Channel Condition (061), Traffic Lanes On (028A), and Design Load

(031)—all exceeded the streamline threshold. A similar selection strategy was employed for the other components using their respective SHAP-based thresholds.

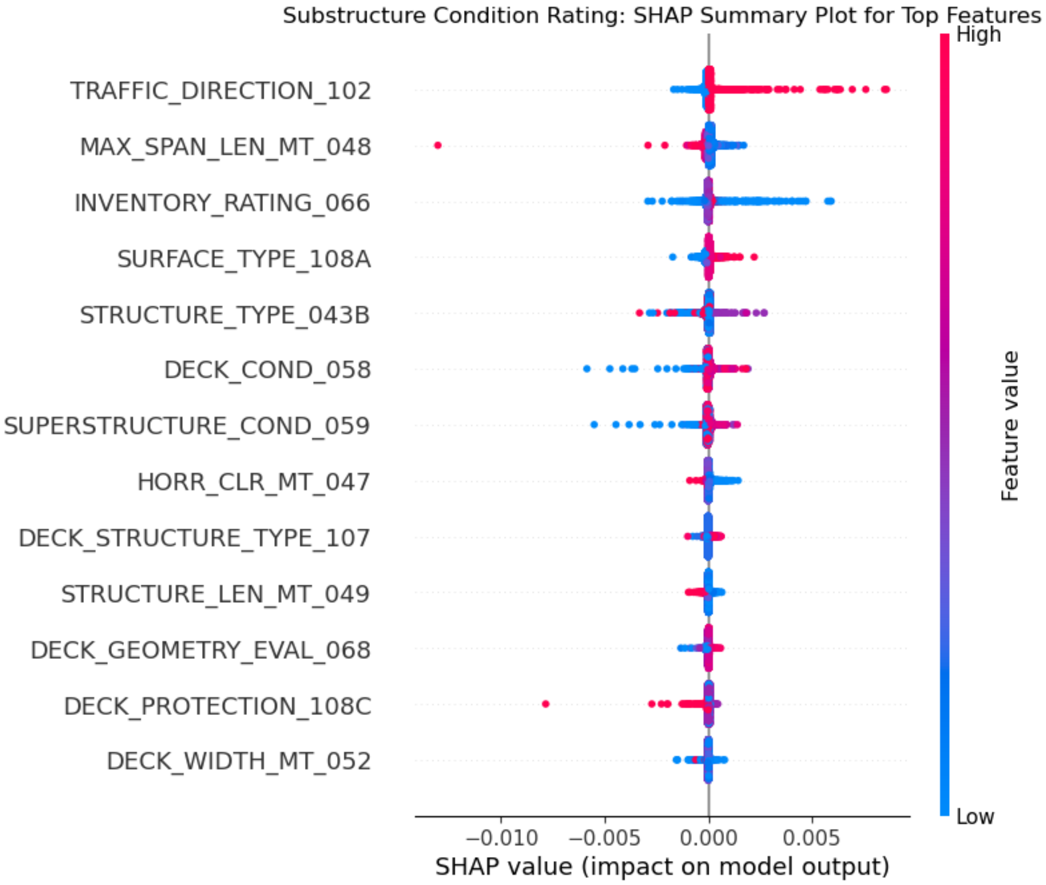


Figure 19. SHAP-based feature importance plot showing the relative influence of input variables on substructure condition prediction

Table 2 presents the top SHAP-identified features for each structural component, representing the most influential predictors of deterioration. These selected features form the optimized input set used for developing and evaluating all subsequent forecasting and physics-guided models.

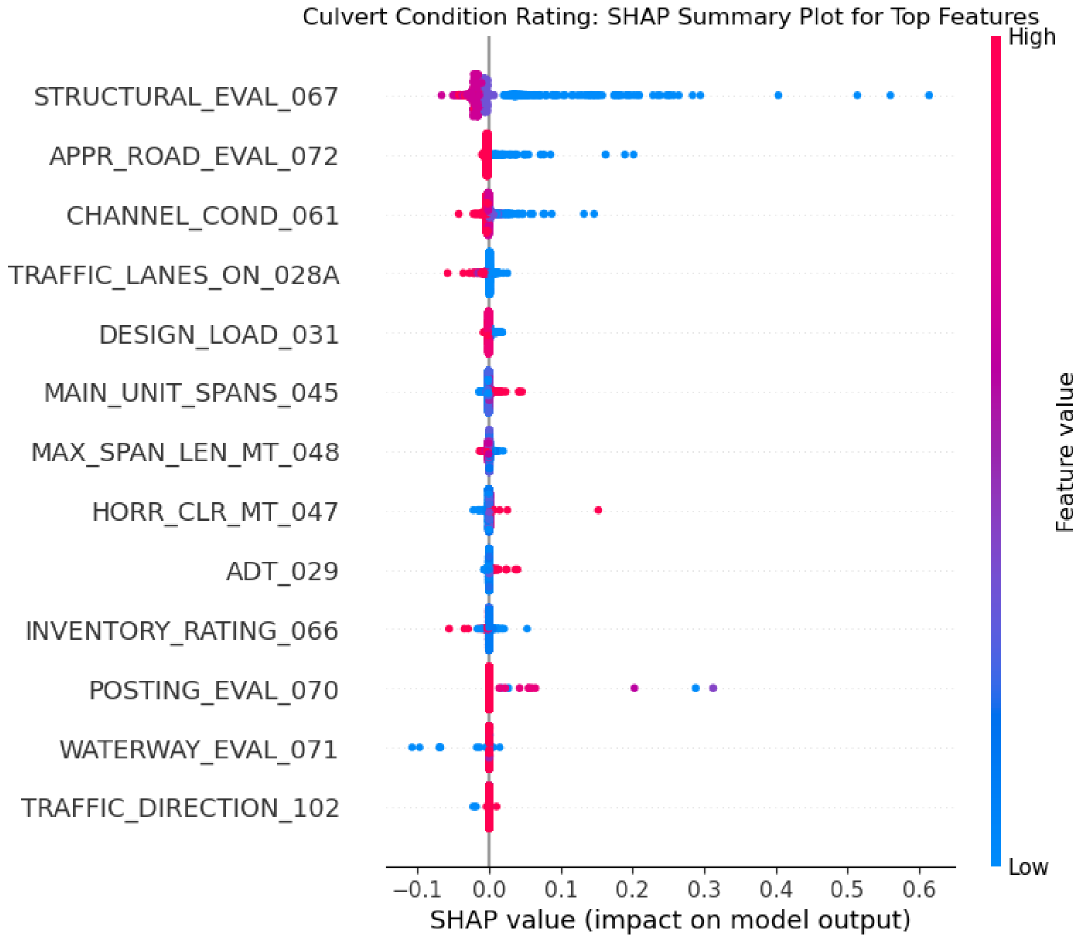


Figure 20. SHAP-based feature importance plot showing the relative influence of input variables on culvert condition prediction

Table 2. Most Impactful SHAP-Selected Features for Each Structural Component

Component	**Top Features Identified by SHAP Analysis**	**Description / Influence on Deterioration**
Deck	Deck Geometry Evaluation (068); Deck Protection (108C); Substructure Condition Rating (060); Waterway Evaluation (071); Superstructure Condition Rating (059); Deck Structure Type (107); Maximum Span Length (048); Structural Evaluation (067); Main Unit Spans (045); Structure Length (049)	These features capture geometric adequacy, material protection, and structural configuration affecting deck performance. Deck geometry and protection control drainage and corrosion resistance, while span length and structure type influence load distribution and stress propagation across connected elements.
Superstructure	Approach Road Evaluation (072); Deck Condition Rating (058); Structure Type (043B); Maximum Span Length (048);	These features represent load transfer, geometry, and material interactions governing superstructure performance. Approach road

	Substructure Condition Rating (060); Deck Geometry Evaluation (068); Traffic Direction (102); Inventory Load Rating (066); Surface Type (108A); Structure Length (049)	and traffic direction affect dynamic loading, while span length and structure type influence bending and fatigue. Condition ratings and surface type capture environmental and drainage effects that accelerate deterioration
Substructure	Traffic Direction (102); Maximum Span Length (048); Inventory Rating (066); Surface Type (108A); Structure Type (043B); Deck Condition Rating (058); Superstructure Condition Rating (059); Horizontal Clearance (047); Deck Structure Type (107); Structure Length (049)	These features reflect hydraulic and load-related factors influencing foundation performance. Traffic direction and span length govern load transfer, while surface type and clearance affect moisture and scour. Condition ratings indicate deterioration transmission from upper structural elements.
Culvert	Structural Evaluation (067); Approach Road Evaluation (072); Channel Condition (061); Traffic Lanes On (028A); Design Load (031); Main Unit Spans (045); Maximum Span Length (048); Horizontal Clearance (047); Average Daily Traffic (029); Inventory Load Rating (066)	These features capture hydraulic, structural, and traffic-driven effects on culvert performance. Structural and approach evaluations represent overall stability, while channel condition and design load define hydraulic and load capacity. Traffic, span geometry, and clearance contribute to wear and deformation.

3.3.2 Repair-Agnostic Methodology for Bridge Condition Forecasting

This subsection introduces a repair-agnostic methodology for forecasting bridge condition ratings. Unlike conventional approaches that rely on labeled repair data, this method identifies and segments repair-like events (see Figure 21) directly from condition rating trends, allowing forecasting to proceed without explicit repair annotations.

Repairs are identified by detecting significant increases (“bumps”) in component condition ratings, which indicate maintenance or rehabilitation actions.

3.3.2.1 Repair-Aware Segmentation: Identification and Integration of Repair Events

Let:

$$\mathbf{X}_{\mathrm{i}} = \{\mathrm{x}_{\mathrm{i}1}, \mathrm{x}_{\mathrm{i}2}, \ldots, \mathrm{x}_{\mathrm{iT}}\}$$

represent the condition rating time series for bridge i, where x_{it} is the condition rating at time t. A repair event occurs at time t = r if:

$$x_{i(r)} - x_{i(r-1)} \geq \delta$$

Where δ > 0 is a predefined threshold representing a significant improvement in the condition rating.

Each time series is then segmented at the identified repair points as follows:

$$\mathbf{X}_i = \left\{\mathbf{X}_i^{(1)}, \mathbf{X}_i^{(2)}, \dots, \mathbf{X}_i^{(k)}\right\}$$

where, each segment $\mathbf{X}_i^{(j)}$ represents a continuous period of deterioration between two successive repairs. This segmentation enables focused modeling of natural deterioration while isolating maintenance-driven improvements.

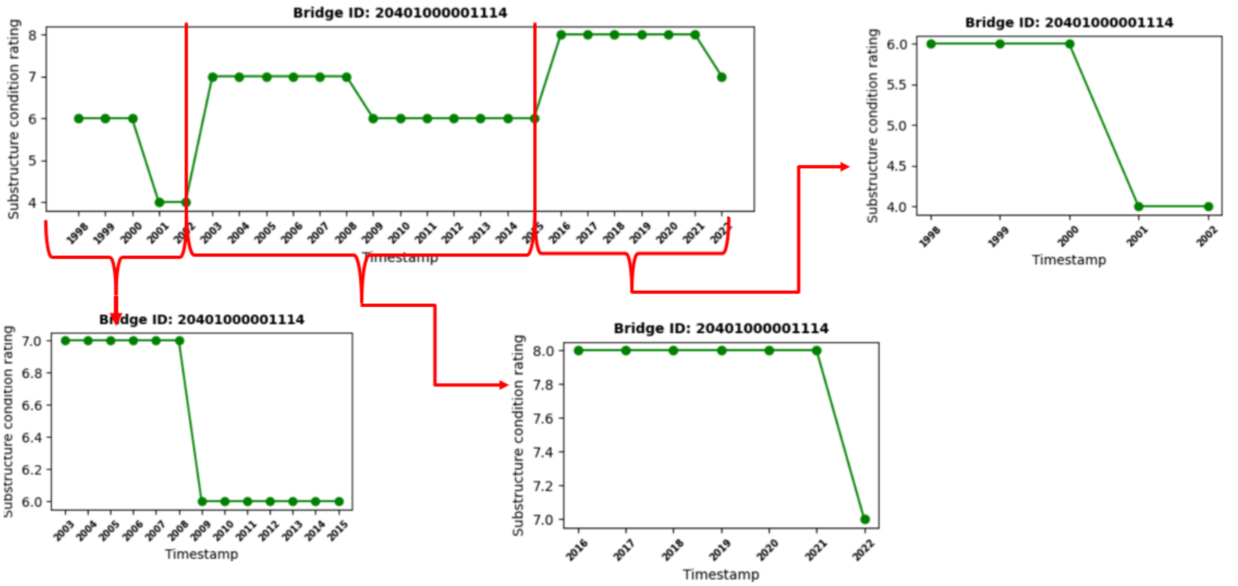


Figure 21. Segmentation process applied to bridge condition ratings, showing how repair years divide the series into continuous deterioration segments.

3.3.2.2 Repair-Aware Modeling: Training and Prediction

Each segment is treated as an independent training sample to capture distinct deterioration patterns. For forecasting, the model uses the most recent segment of each bridge's time series:

$$\hat{x}_{i(T+1)} = f_\theta(\mathbf{X}_i^{(k)}) \tag{15}$$

where f_Θ represents the forecasting model (e.g., LSTM, TCN), and $\mathbf{X}_i^{(k)}$ denotes the latest available segment. This design ensures that predictions reflect current deterioration conditions while being unaffected by earlier repair discontinuities.

3.3.2.3 Handling Variable-Length Sequences: Masking and Zero Padding

To accommodate variable sequence lengths caused by segmentation, zero padding and masking are implemented:

- **Zero Padding:** Each sequence is padded with zeros to match the maximum length in the dataset, enabling uniform batch processing.
- **Masking:** During computation, padded values are ignored to ensure that the model only processes valid data points.

This approach allows efficient training of deep learning models while maintaining accuracy across sequences of differing lengths. It was applied consistently across all models, including LSTM, BiLSTM, CNN–BiLSTM, CNN, GRU, Multi-channel CNN, and TCN.

3.4 Experimental Evaluation

In this section, we will go into experimental evaluation of the bridge and culvert deterioration forecasting models specified above, how they were used in this specific problem, and how the added complexity when moving from traditional regression-based models to novel physics-guided neural networks did indeed allow for better performance. The analysis is structured into two key components: (1) a comparative analysis of baseline data-driven models against enhanced physics-guided models and repair agnostics models; and (2) an ablation study examining the incremental contributions of physics-based loss functions and repair data integration. These evaluations use Root Mean Square Error (RMSE) as the primary performance metric to ensure consistent and interpretable comparisons across models and configurations. Overall, this evaluation provides critical insights into how physical constraints and maintenance interventions enhance deterioration forecasting models' robustness, accuracy, and reliability.

3.4.1 Experimental Setup and Baseline Definition

Our experimental framework evaluates baseline data-driven models, physics-guided models, and repair-aware variants. All experiments are implemented in Python 3.9, using TensorFlow for neural network models and Scikit-Learn for linear baselines. We benchmark eight architectures: **LSTM**, **BiLSTM**, **GRU**, **CNN**, **CNN-BiLSTM**, **Multi-channel CNN**, **TCN**, and **Linear Regression**. Neural networks are trained on an NVIDIA GeForce RTX 2070 Super GPU.

Training details:

- Optimizer: Adam
- Batch size: 64
- Max epochs: 300 with early stopping (patience = 50)
- Objective (baseline): minimize Mean Squared Error (MSE)
- Physics-guided models: add a physical-inconsistency penalty to the loss (Section 3)

Performance is reported as RMSE (see Equation 1) on a held-out 30% test split (70% training). Table 3 summarizes key hyperparameters for the evaluated models.

Table 3. Training Hyperparameters for Evaluated Models

Model	**Training Parameters / Hyperparameters**
LSTM	Units = 32; Return Sequence = False; Dense Units = out_steps × num_features; Loss = MSE; Optimizer = Adam
GRU	Units = 32; Return Sequence = False; Dense Units = out_steps × num_features; Loss = MSE; Optimizer = Adam
TCN	Filters = variable; Dropout = [0.0–0.8]; Return Sequence = False; Dense Units = out_steps × num_features
BiLSTM	Bidirectional Units = 32; Return Sequence = False; Dense Units = out_steps × num_features; Loss = MSE; Optimizer = Adam
CNN	Conv1D Filters = 64; Activation = ReLU; Dropout = 0.2; Dense Units = out_steps × num_features
CNN-BiLSTM	CNN layer followed by BiLSTM; tuned similar to CNN and BiLSTM individually
Multi-Channel CNN	Conv1: Filters = 8, Kernel = 5, Activation = ReLU; MaxPool = 2; Conv2: Filters = 4, Kernel = 5; Dense Units = 732
Linear Regression	Scikit-Learn LinearRegression (default parameters)

Finally, we trained the above models with four different combinations (see Figure 22) to conduct a comparative analysis. The models are as follows:

I. Data-Driven Models

II. Physics-Guided Models (i.e., Data Driven + Physics-based)

III. Repair Agnostic Models (i.e., Data-Driven models that are Repair Agnostic)

IV. Repair Agnostic Physics-Guided Models (i.e., Physics-Guided models that are Repair Agnostic)

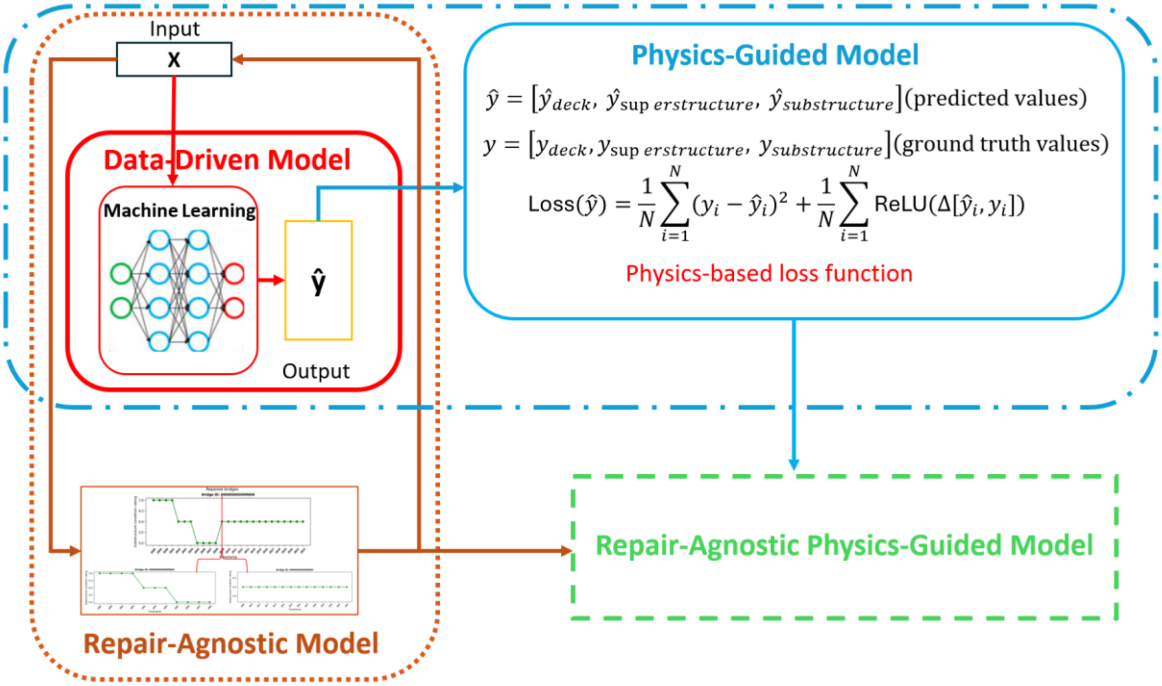

Input
X
Data-Driven Model
Machine Learning
ŷ
Output
Physics-Guided Model
$\hat{y} = [\hat{y}_{deck}, \hat{y}_{\sup erstructure}, \hat{y}_{substructure}]$(predicted values)
$y = [y_{deck}, y_{\sup erstructure}, y_{substructure}]$(ground truth values)
$\text{Loss}(\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \frac{1}{N}\sum_{i=1}^{N}\text{ReLU}(\Delta[\hat{y}_i, y_i])$
Physics-based loss function
Repair-Agnostic Physics-Guided Model
Repair-Agnostic Model

Figure 22. Overview of the four model categories used in bridge deterioration forecasting
The framework illustrates the relationships between (1) the **Data-Driven Model (Red box)**, which learns deterioration patterns directly from inspection data; (2) the **Physics-Guided Model (Blue box)**, which augments data-driven predictions with mechanistic constraints using a physics-based loss term; (3) the **Repair-Agnostic Model (Orange box)**, which learns deterioration behavior with consideration of maintenance events; and (4) the **Repair-Agnostic Physics-Guided Model (Green box)**, which integrates both physics-based constraints and repair-agnostic learning. Together, these models progressively combine empirical learning, domain knowledge, and maintenance-independent representations to enhance predictive performance and interpretability in bridge condition forecasting.

3.4.2 Comparative Analysis

In this section, we perform a detailed comparative analysis involving four model types—data-driven, physics-guided, repair-agnostic, and repair-agnostic physics-guided—organized into three distinct comparative studies.

3.4.2.1 Comparative Study I: Data-Driven Models vs. Physics-Guided Models

This subsection presents a detailed comparative analysis between purely data-driven models and their extended equivalent models incorporating physics-guided loss functions. The objective is to evaluate the impact of integrating physics-informed constraints on model performance across four structural components: deck, superstructure, substructure, and culvert.

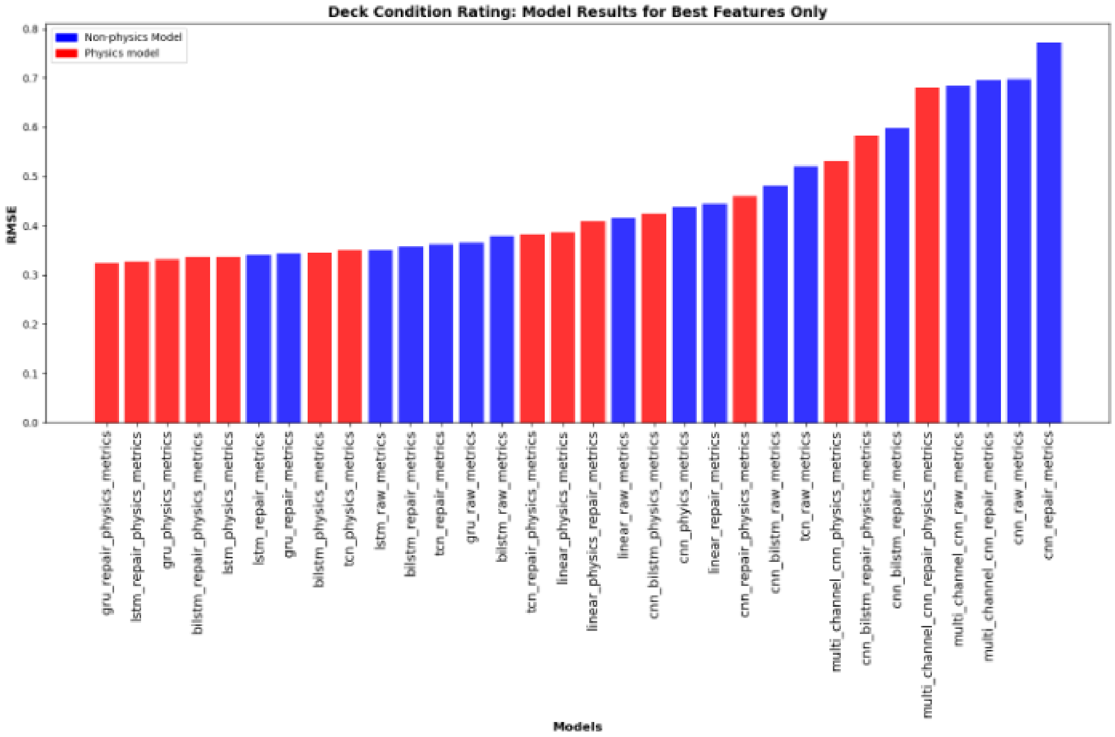


Figure 23. Deck Results for Physics-guided vs Purely Data-driven Models

In the baseline setup, models are trained using standard data-driven learning, while the amended versions incorporate domain-specific physical knowledge into the loss function. Model performance is evaluated using Root Mean Square Error (RMSE), where lower values indicate higher predictive accuracy. Figures 23 to 26 summarize the RMSE results for each model under both configurations. As shown in the figures, physics-guided models consistently outperform their data-driven peers across nearly all architectures.

For deck deterioration forecasting, the LSTM model achieves the lowest RMSE among the data-driven models (0.3501). When the physics-guided loss function is introduced, the GRU model obtains the best performance, reducing RMSE to 0.3319. The LSTM model also shows an RMSE drop to 0.3368—very close to the GRU’s score. These results suggest that both GRU and LSTM architectures are particularly effective in incorporating physical constraints during training.

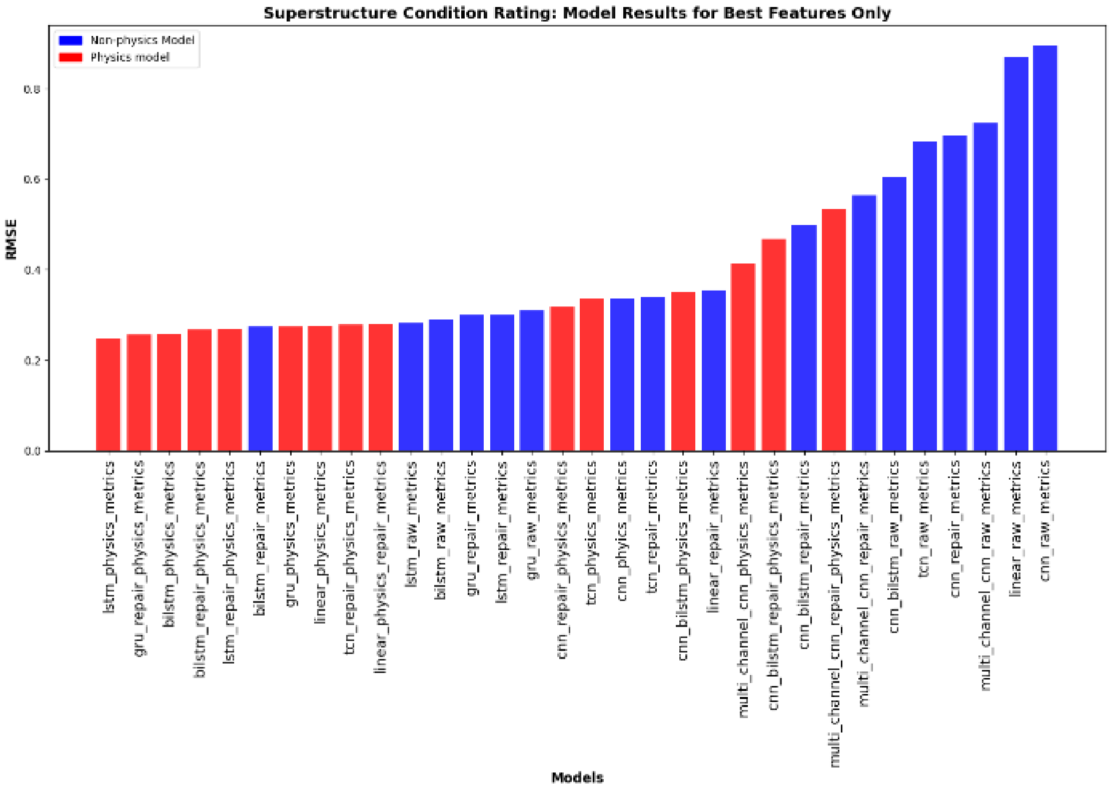


Figure 24. Superstructure Results for Physics-guided vs Purely Data-driven Models

Among the superstructure models, the data-driven LSTM achieves a winning RMSE of 0.2826. With physics guidance, the same model reduces its RMSE to 0.2482, beating all others in this category. The Physics-Guided BiLSTM also performs competitively, attaining an RMSE of 0.2583, confirming its effectiveness under physics-aware configurations.

In substructure condition prediction, the Physics-Guided BiLSTM model demonstrates the most observable improvement, reducing RMSE from 0.2712 to 0.2516 when trained with knowledge-informed loss functions. This reflects the model's capacity to capture long-term temporal patterns influenced by physical degradation behavior.

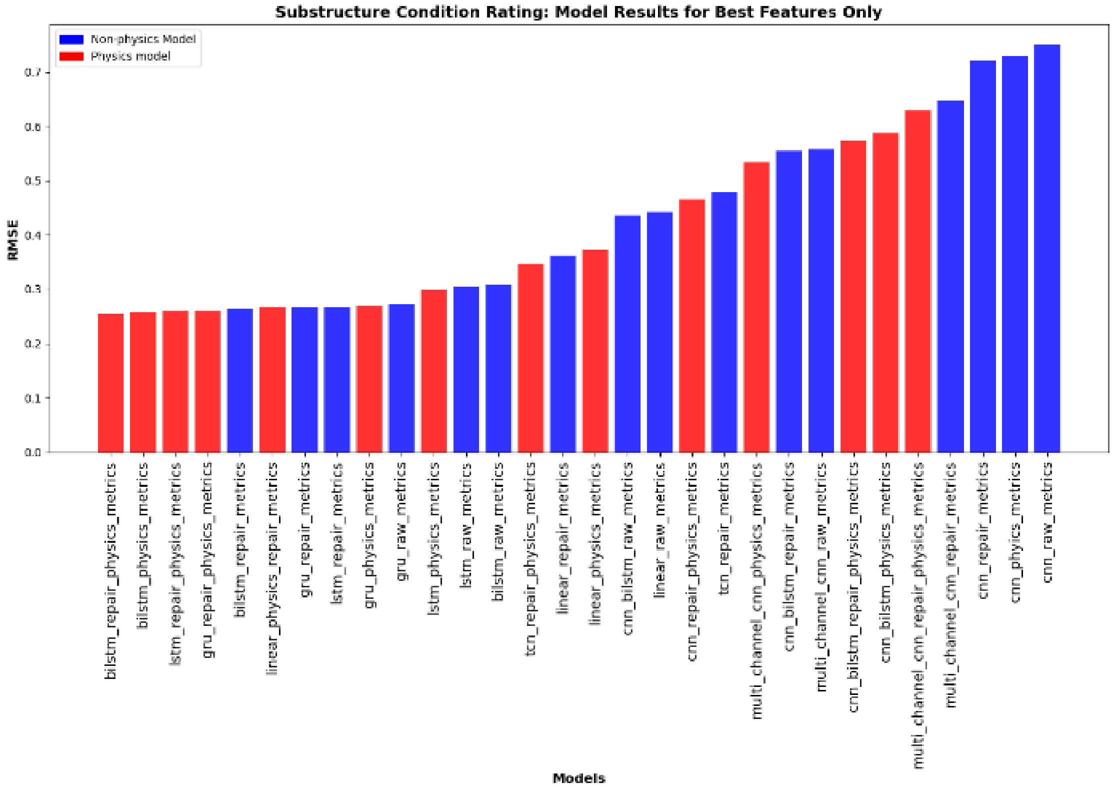


Figure 25. Substructure Results for Physics-guided vs Purely Data-driven Models

In culvert condition forecasting, the Linear Regression model initially performs best among traditional approaches (RMSE = 0.7102). However, with physics-guided loss, the CNN model significantly improves performance, achieving a lower RMSE of 0.6802, outperforming all other model variants. The GRU model follows closely with an RMSE of 0.6855, further demonstrating its effectiveness.

Among all evaluated configurations, the LSTM, GRU, and BiLSTM models show the greatest performance improvements, with RMSE reductions of up to 0.3 when transitioning from purely data-driven to physics-guided learning. These results confirm the advantage of embedding physical knowledge into the model's training objective to improve accuracy and real-world explainability.

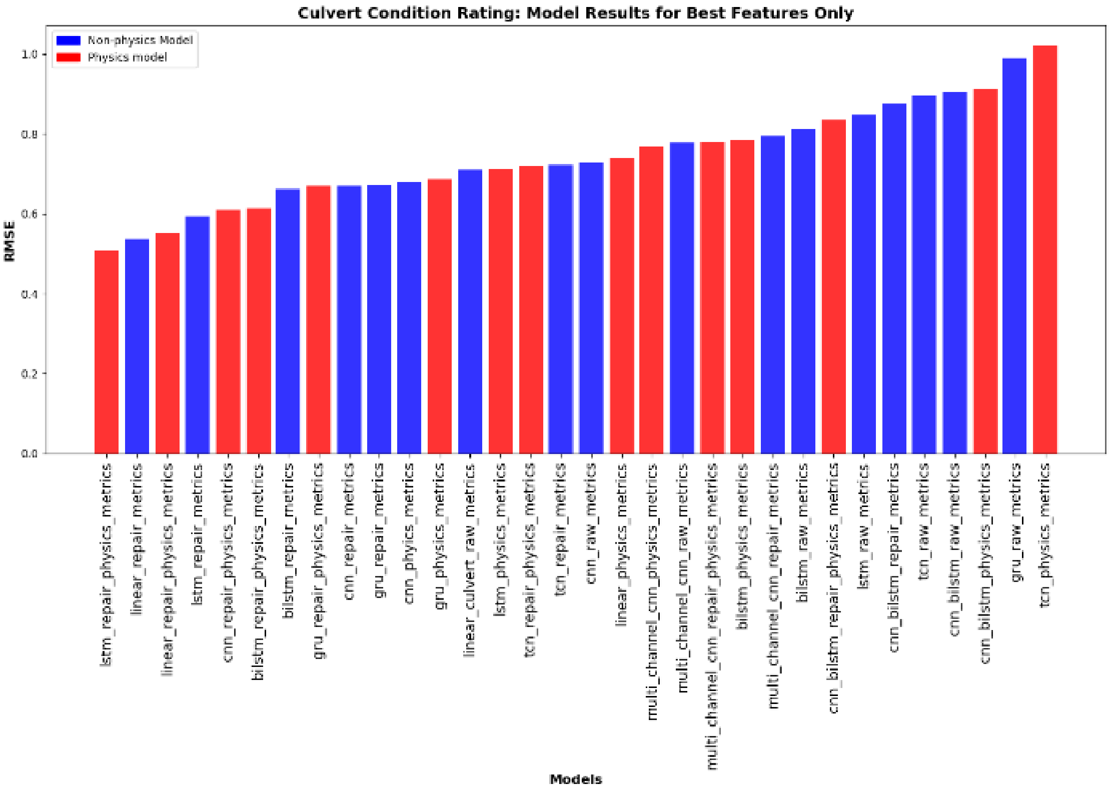


Figure 26. Culvert Results for Physics-guided vs Purely Data-driven Models

Comparative Result Summary:

Physics-guided models consistently outperform their data-driven counterparts across all structural components. These results highlight the importance of integrating domain knowledge into the learning process and identify physics-guided GRU, BiLSTM, and LSTM as promising candidates for further evaluation. A comparative result analysis of data-driven and physics-guided models is presented in Table 4. The subsequent subsection extends this investigation by assessing the impact of incorporating real-world repair data on model performance.

Table 4. Comparison of Experimental results (RMSE) for Data-Driven Models vs. Physics-Guided Models Across Structural Components

Model	**Deck Data-Driven**	**Deck Physics-Guided**	**Super-structure Data-Driven**	**Super-structure Physics-Guided**	**Sub-structure Data-Driven**	**Sub-structure Physics-Guided**	**Culvert Data-Driven**	**Culvert Physics-Guided**

LSTM	**0.3501***	0.3368	**0.2826***	**0.2482***	**0.2652***	0.2519	0.8461	0.7117
GRU	0.365	**0.3319***	0.3097	0.2741	0.3418	0.2635	0.9898	0.6855
BiLSTM	0.3793	0.3457	0.2914	0.2583	0.2712	**0.2516***	0.8129	0.7845
Linear	0.4155	0.3862	0.8685	0.2757	0.4493	0.3123	**0.7102***	0.741
CNN	0.6968	0.4383	0.8949	0.3374	0.7842	0.3772	0.7286	**0.6802***
CNN-BiLSTM	0.481	0.4242	0.6041	0.3503	0.4873	0.3602	0.9057	0.9125
TCN	0.5203	0.35	0.6831	0.7565	0.702	0.3073	0.8979	1.02
Multi-Channel-CNN	0.6831	0.5311	0.7229	0.413	0.5232	0.5224	0.7788	0.7683

Note: The rows represent the different machine- and deep-learning models. The columns are organized in pairs for side-by-side comparison: the first column in each pair shows the data-driven model results, and the second column shows the physics-guided model results. Highlighted Bold values (marked with an asterisk) indicate the lowest RMSE within each component. Lower RMSE values correspond to better predictive performance. For example, in the deck column, the physics-guided GRU model achieves the winning RMSE of 0.3319 (cyan with an asterisk). Colors are provided only for visual emphasis; all results can be interpreted directly from the numerical values.

3.4.2.2 Comparative Study II: Data-Driven Models vs. Repair-Agnostic Models

In this analysis, we compare the baseline data-driven models that were trained without considering repair cases with models trained using repair-agnostic data, which consider real-world maintenance events. This setup evaluates the benefit of using historical repair information in model training. Figures 27 to 30 summarize the RMSE results for each model under both configurations.

For deck condition rating, the data-driven LSTM reports an RMSE of 0.3501, while the repair-agnostic LSTM achieves a reduced RMSE of 0.3414, confirming the lowest score among all other models. A similar pattern is observed for GRU, BiLSTM, and TCN models. In the superstructure domain, the repair-agnostic BiLSTM model achieves the best performance, and other models also show impressive RMSE reductions under the repair-agnostic setting. Substructure condition ratings prediction models follow the same trend, with models like TCN reducing RMSE from 0.702 to 0.4254, and repair-agnostic LSTM as the winner model. Culvert prediction also benefited from repair-agnostic model training, with the best-performing model, LSTM, improving its RMSE from 0.8461 to 0.5929.

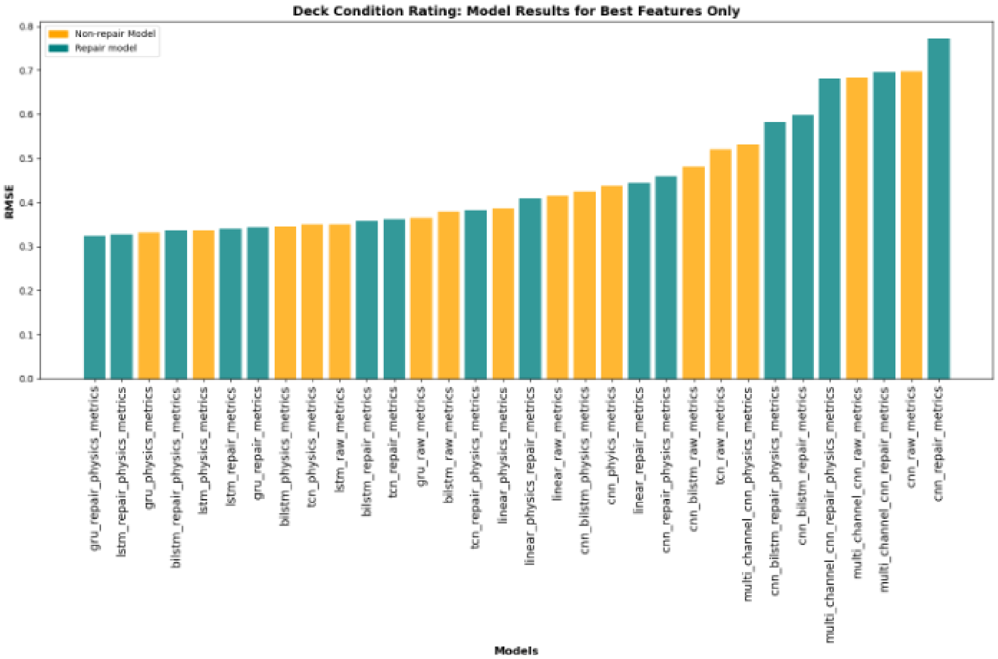


Figure 27. Deck Results for Repair Events vs Non-Repair

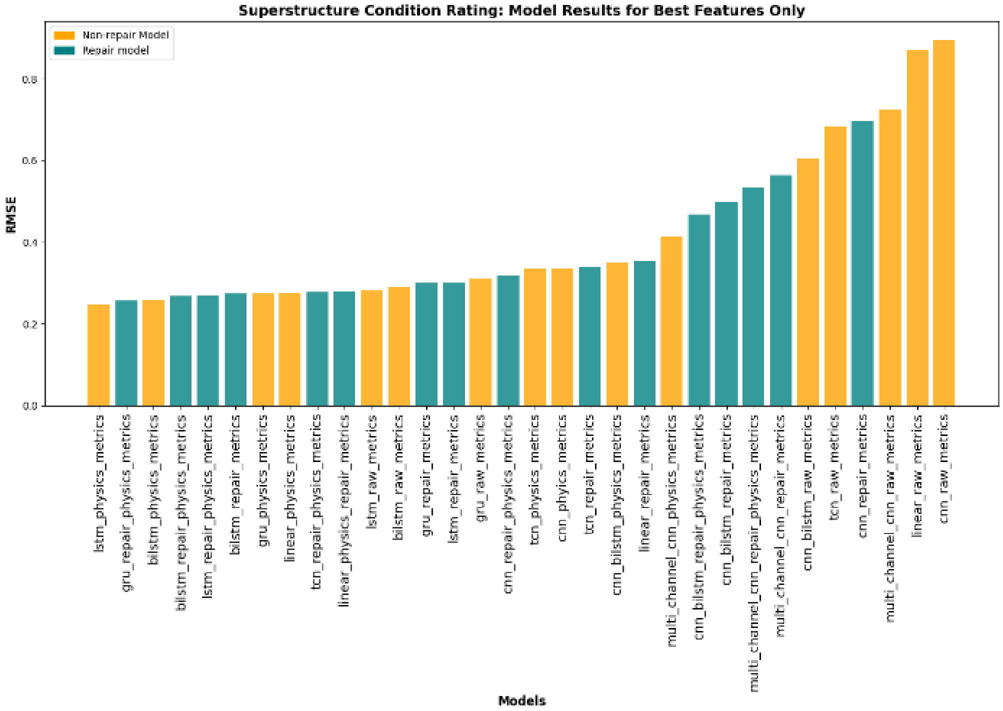


Figure 28. Superstructure Results for Repair Events vs Non-Repair

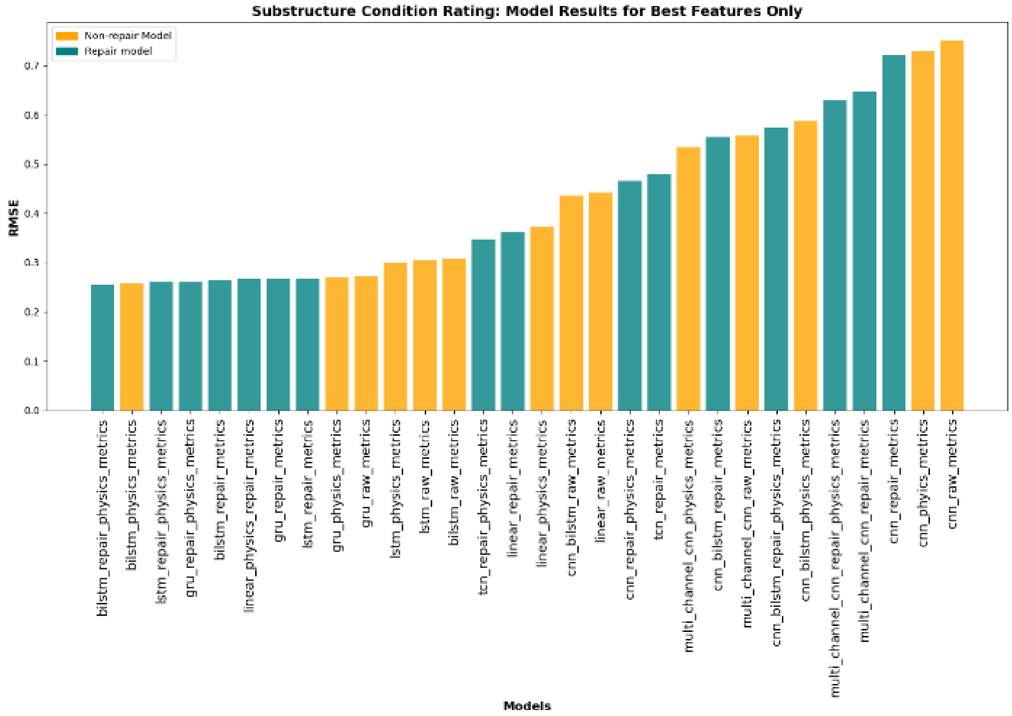


Figure 29. Substructure Results for Repair Events vs Non-Repair

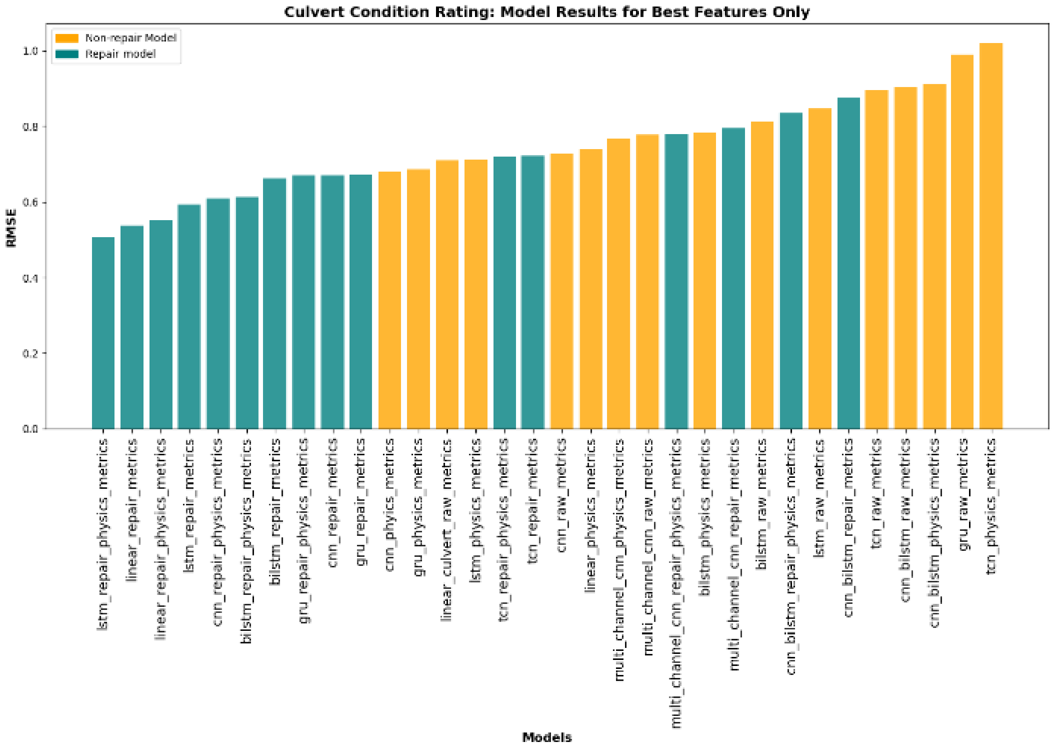


Figure 30. Culvert Results for Repair Events vs Non-Repair

These results confirm that incorporating bridge and culvert maintenance information in the training procedure can optimize the models' performance and reduce overfitting, and allow the network to generalize better from available data.

Comparative Result Summary:

Repair-agnostic models consistently outperform their data-driven versions across all structural components, as shown in Table 5. Incorporating real-world repair events during the training process leads to reduced RMSE values, indicating that models trained with repair-agnostic data are less prone to overfitting and better at capturing underlying deterioration trends. More specifically, LSTM and BiLSTM models exhibit substantial gains in deck, superstructure, substructure, and culvert prediction, while models like TCN and GRU show marked improvement in all categories. These results emphasize the value of using clean, intervention-considered datasets to enhance the generalization capability of condition rating models.

Table 5. Comparison of experimental results (RMSE) for Data-Driven vs. Repair-Agnostic models across structural components

Model	**Deck Data-Driven**	**Deck Repair-Agnostic**	**Superstructure Data-Driven**	**Superstructure Repair-Agnostic**	**Substructure Data-Driven**	**Substructure Repair-Agnostic**	**Culvert Data-Driven**	**Repair-Agnostic**
LSTM	**0.3501***	**0.3414***	**0.2826***	0.3001	**0.2652***	**0.2623***	0.8461	0.5929
GRU	0.365	0.3435	0.3097	0.2997	0.3418	0.2902	0.9898	0.671
BiLSTM	0.3793	0.3573	0.2914	**0.2739***	0.2712	0.2652	0.8129	0.6624
Linear	0.4155	0.4443	0.8685	0.3551	0.4493	0.4099	**0.7102***	**0.5361***
CNN	0.6968	0.7724	0.8949	0.6973	0.7842	0.7498	0.7286	0.67
CNN-BiLSTM	0.481	0.5983	0.6041	0.4981	0.4873	0.5696	0.9057	0.8744
TCN	0.5203	0.3618	0.6831	0.3389	0.702	0.4254	0.8979	0.7215
Multi-Channel-CNN	0.6831	0.6955	0.7229	0.564	0.5232	0.8402	0.7788	0.7948

Note: The rows represent the different machine- and deep-learning models. The columns are organized in pairs for side-by-side comparison: the first column in each pair shows the data-driven model results, and the second column shows the repair-agnostic model results. Highlighted Bold values (marked with an asterisk) indicate the lowest RMSE within each component. Lower RMSE values correspond to better predictive performance. For example, in the deck column, the repair-agnostic LSTM model achieves the winning RMSE of 0.3414 (gray with an asterisk). Colors are provided only for visual emphasis; all results can be interpreted directly from the numerical values.

3.4.2.3 Comparative Study III: Physics-Guided Models vs. Repair-Agnostic Physics-Guided (RAPG) Models

This comparative analysis evaluates the added value of integrating repair-agnostic training strategies into physics-guided deep learning models. These models already benefit from physical constraints in the loss function; the addition of repair-agnostic data helps determine whether further generalization is achievable by considering repair history.

In the deck component, LSTM improves from 0.3319 to 0.3267, and GRU from 0.3319 to 0.3241, showing consistent enhancement in predictive accuracy. BiLSTM improves slightly from 0.3457 to 0.3355. For superstructure condition forecasting, GRU improves from 0.2741 to 0.2565, and CNN from 0.3374 to 0.3167. Substructure models follow a similar trend, where LSTM improves from 0.2519 to 0.2472, and BiLSTM from 0.2516 to 0.2472. Culvert predictions show the most dramatic improvement in LSTM, with RMSE dropping from 0.7117 to 0.5057.

Comparative Result Summary:

Across all components, repair-agnostic physics-guided models consistently outperform their physics-only equivalent, as shown in Table 6. This configuration leverages both domain knowledge and robust generalization from repair-agnostic data, leading to the most reliable and physically consistent condition rating forecasts. These results affirm that the repair-agnostic physics-based approach is the most effective strategy among all evaluated configurations.

Table 6. Comparison of experimental results (RMSE) for Physics-Guided (PG) vs. Repair-Agnostic Physics-Guided (RAPG) models across structural components

Model	**Deck PG**	**Deck RAPG**	**Super-structure PG**	**Super-structur RAPG**	**Sub-structur PG**	**Sub-structure RAPG**	**Culvert PG**	**Culver RAPG**
LSTM	0.3368	0.3267	**0.2482***	0.2689	0.2519	**0.2472***	0.7117	**0.5057***
GRU	**0.3319***	**0.3241***	0.2741	**0.2565***	0.2635	0.2534	0.6855	0.6698
BiLSTM	0.3457	0.3355	0.2583	0.2679	**0.2516***	**0.2472***	0.7845	0.6134
Linear	0.3862	0.4093	0.2757	0.2791	0.3123	0.2555	0.741	0.5509
CNN	0.4383	0.4603	0.3374	0.3167	0.3772	0.3581	**0.6802***	0.6092
CNN-BiLSTM	0.4242	0.5834	0.3503	0.4675	0.3602	0.5542	0.9125	0.8341
TCN	0.35	0.3822	0.3349	0.2787	0.3073	0.3043	1.02	0.7198

Multi-Channel-CNN	0.5311	0.6812	0.413	0.5331	0.5224	0.7018	0.7683	0.7814

Note: The rows represent the different machine- and deep-learning models. The columns are organized in pairs for side-by-side comparison: the first column in each pair shows the physics-guided model results, and the second column shows the repair-agnostic physics-guided model results. Highlighted Bold values (marked with an asterisk) indicate the lowest RMSE within each component. Lower RMSE values correspond to better predictive performance. For example, in the superstructure column, the physics-guided GRU model achieves the winning RMSE of 0.2482 (green highlighted with an asterisk). Colors are provided only for visual emphasis; all results can be interpreted directly from the numerical values.

3.4.3 Ablation Analysis: Stepwise Performance Improvement from Feature Selection to Repair-Agnostic Physics Models

To provide a comprehensive understanding of the performance evolution, this section presents a stepwise ablation analysis demonstrating how structural condition rating predictions progressively improve across five model stages:

1.The purely data-driven model trained with all available features,

2.The SHAP-selected data-driven model,

3.The Repair-Agnostic (RA) model,

4.The Physics-Guided (PG) model, and

5.The combined Repair-Agnostic Physics-Guided (RAPG) model.

This progressive evaluation highlights the individual and combined contributions of feature selection, domain-informed learning, and repair-agnostic training to overall model accuracy and generalization.

Performance Impact of Feature Selection:

As a preliminary step described in section 3.3, feature importance analysis using SHAP identified the most influential input variables for each component.

Figure 31 and Table 7 illustrate the performance improvement patterns of models when using all available features compared to only SHAP-selected best features. This comparison serves as a preliminary ablation analysis, highlighting the performance gains from feature reduction. This analysis not only improves model performance but also facilitates effective feature selection. By prioritizing the most informative predictors, we reduce model complexity and training time while improving generalization.

Table 7. Improvement in Root Mean Squared Error (RMSE) Achieved by Using SHAP-Selected Best Features over All Available Features

Model	Deck (%)	Superstructure (%)	Substructure (%)	Culvert (%)
LSTM	66	74	76	27
GRU	65	71	68	15
BiLSTM	63	73	75	29
Linear	99	99	99	98
CNN	91	86	93	74
CNN–BiLSTM	54	45	56	22
TCN	51	52	37	74
Multi-Channel CNN	84	79	90	82

Using only the most important predictors, we achieved an average RMSE improvement of 63% across all structural components, demonstrating the effectiveness of SHAP-based feature selection as the foundation for subsequent model development. These optimized feature sets were then used for all following experiments, including the Repair-Agnostic, Physics-Guided, and RAPG configurations.

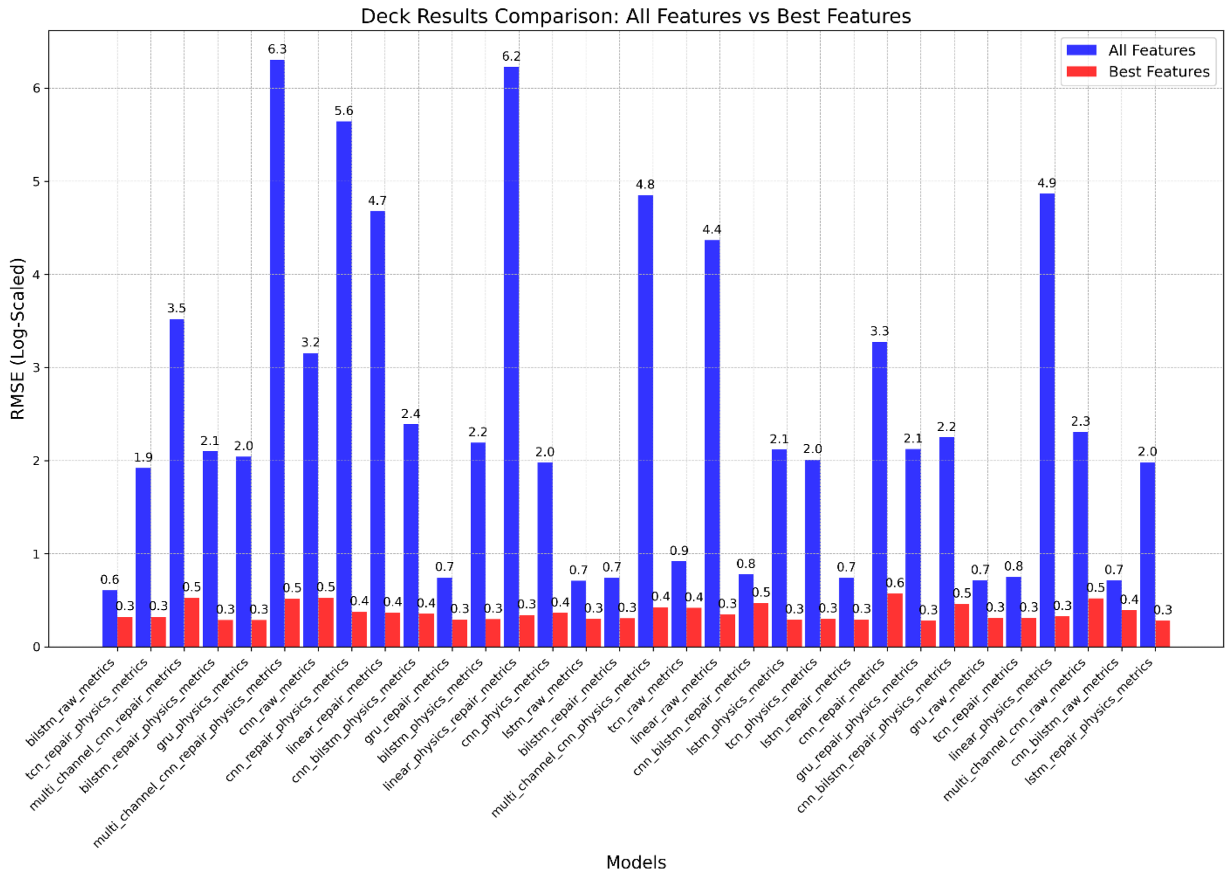


Figure 31. Model Performance Using All Features vs. SHAP-Selected Best Features

Progressive Performance Improvement from Data-Driven Models to Repair-Agnostic Physics-Guided Models

Building on the SHAP-optimized feature inputs, further enhancements were introduced through physics-guided and repair-agnostic mechanisms.

The optimal subset of features for each component, identified using SHAP values, is applied to all following models—from the Data-Driven to the RAPG configuration. The Data-Driven model, built on these selected features, already demonstrates a significant RMSE improvement compared to the All-Features data-driven models, as mentioned before. The incorporation of physics-guided loss functions further reinforces performance by implementing physical consistency, thereby mitigating overfitting to noisy or spurious training signals. The most substantial improvements, however, are achieved with the RAPG model, which integrates both physics-based learning and repair-agnostic training. Table 8 summarizes the stepwise RMSE improvements observed throughout the model development pipeline.

Table 8. RMSE Improvement Over Training Stages from the All-Features Data-Driven Model to Repair-Agnostic Physics-Guided (RAPG) Model

Component	**AFDDM → Data-Driven**	**Data-Driven → RA**	**Data-Driven → PG**	**Data-Driven → RAPG**	**AFDDM → RAPG**
Deck	66%	2%	5%	7%	69%
Superstructure	74%	3%	12%	9%	76%
Substructure	75%	1%	5%	7%	77%
Culvert	38%	25%	4%	29%	56%
Avg. Improvement	**63%**	**8%**	**7%**	**13%**	**69%**

Starting with an average 63% gain across all components from feature selection using SHAP analysis, model performance improves by an additional 7% with the integration of a physics-guided loss function to data-driven models with SHAP-selected features and 13% upon incorporating repair-agnostic training data with physics loss, resulting in a total performance gain of up to 69% (an average improvement of all components) over models trained with all features. This configuration effectively minimizes bias introduced by historical repair interventions while retaining the benefits of domain-aware modeling.

Table 9. Stepwise Ablation Analysis: Comparison of RMSE across Data-Driven, Repair-Agnostic, Physics-guided, and Repair-Agnostic Physics-guided (RAPG) models

Model	**LSTM**	**GRU**	**BiLSTM**	**Linear**	**CNN**	**CNN-BiLSTM**	**TCN**	**Multi-Channel-CNN**
Deck Data-Driven	**0.3501**	0.3650	0.3793	0.4155	0.6968	0.4810	0.5203	0.6831
Deck Repair-agnostic	**0.3414**	0.3435	0.3573	0.4443	0.7724	0.5983	0.3618	0.6955
Deck Physics-guided	0.3368	**0.3319**	0.3457	0.3862	0.4383	0.4242	0.3500	0.5311
Deck Repair-agnostic Physics-guided	0.3267	**0.3241**	0.3355	0.4093	0.4603	0.5834	0.3822	0.6812
Superstructure Data-Driven	**0.2826**	0.3097	0.2914	0.8685	0.8949	0.6041	0.6831	0.7229
Superstructure Repair-agnostic	0.3001	0.2997	**0.2739**	0.3551	0.6973	0.4981	0.3389	0.5640

Superstructure Physics-guided	**0.2482**	0.2741	0.2583	0.2757	0.3374	0.3503	0.3349	0.4130
Superstructure Repair-agnostic PG	0.2689	**0.2565**	0.2679	0.2791	0.3167	0.4675	0.2787	0.5331
Substructure Data-Driven	**0.2652**	0.3418	0.2712	0.4493	0.7842	0.4873	0.7020	0.5232
Substructure Repair-agnostic	**0.2623**	0.2902	0.2652	0.4099	0.7498	0.5696	0.4254	0.8402
Substructure Physics-guided	0.2519	0.2635	**0.2516**	0.3123	0.3772	0.3602	0.3073	0.5224
Substructure Repair-agnostic PG	**0.2472**	0.2534	**0.2472**	0.2555	0.3581	0.5542	0.3043	0.7018
Culvert Data-Driven	0.8461	0.9898	0.8129	**0.7102**	0.7286	0.9057	0.8979	0.7788
Culvert Repair-agnostic	0.5929	0.6710	0.6624	**0.5361**	0.6700	0.8744	0.7215	0.7948
Culvert Physics-guided	0.7117	0.6855	0.7845	0.7410	**0.6802**	0.9125	1.0200	0.7683
Culvert Repair-agnostic Physics-guided	**0.5057**	0.6698	0.6134	0.5509	0.6092	0.8341	0.7198	0.7814

Note: Highlighted cells indicate the lowest (best) RMSE values at each ablation step.Yellow, orange, green, and blue denote data-driven, repair-agnostic, physics-guided, and repair-agnostic physics-guided (RAPG) models, respectively. Highlighted cells for each component are as follows:

- Deck: LSTM (0.3501), LSTM (0.3414), GRU (0.3319), GRU (0.3241)
- Superstructure: LSTM (0.2826), BiLSTM (0.2739), LSTM (0.2482), GRU (0.2565)
- Substructure: LSTM (0.2652), LSTM (0.2623), BiLSTM (0.2516), LSTM and BiLSTM (0.2472)
- Culvert: Linear (0.7102), Linear (0.5361), CNN (0.6802), LSTM (0.5057)

Table 9 shows a detailed comparison among four category model configurations trained with SHAP-selected features. The performance trend strongly demonstrates the effectiveness of this layered-training strategy. For example, in deck condition prediction, the LSTM model progressively improves from 0.3501 (Data-Driven) to 0.3414 (Repair-Agnostic), then to 0.3368 (Physics-guided), and finally achieves the best performance at 0.3267 (RAPG). Similarly, the GRU model follows this pattern with RMSEs of 0.365, 0.3435, 0.3319, and 0.3241, respectively. This trend is consistent across superstructure, substructure, and culvert predictions, where models like BiLSTM and GRU demonstrate substantial performance gains under the RAPG configuration. These findings affirm that integrating both physics-informed loss functions and repair-agnostic data yields the most accurate and generalizable forecasts for structural condition ratings.

Ablation Study Outcome:

This stepwise ablation clearly illustrates that the RAPG-DL models consistently outperform both data-driven and physics-only configurations. The combination of physics-based modeling and repair-independent training data represents the most effective strategy for structural deterioration forecasting. The ablation analysis results, presented in Figures 32 to 35, show that most models follow a consistent pattern of RMSE reduction, highlighting progressive performance improvements from Data-Driven to Repair-Agnostic Physics-Guided deep learning models.

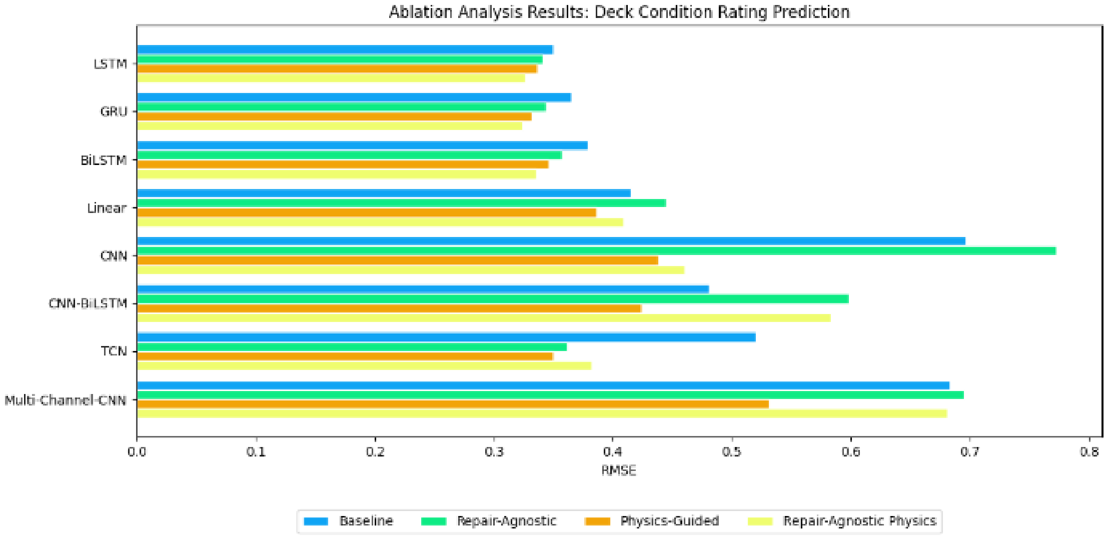


Figure 32. Deck: Comparison of Ablation Analysis

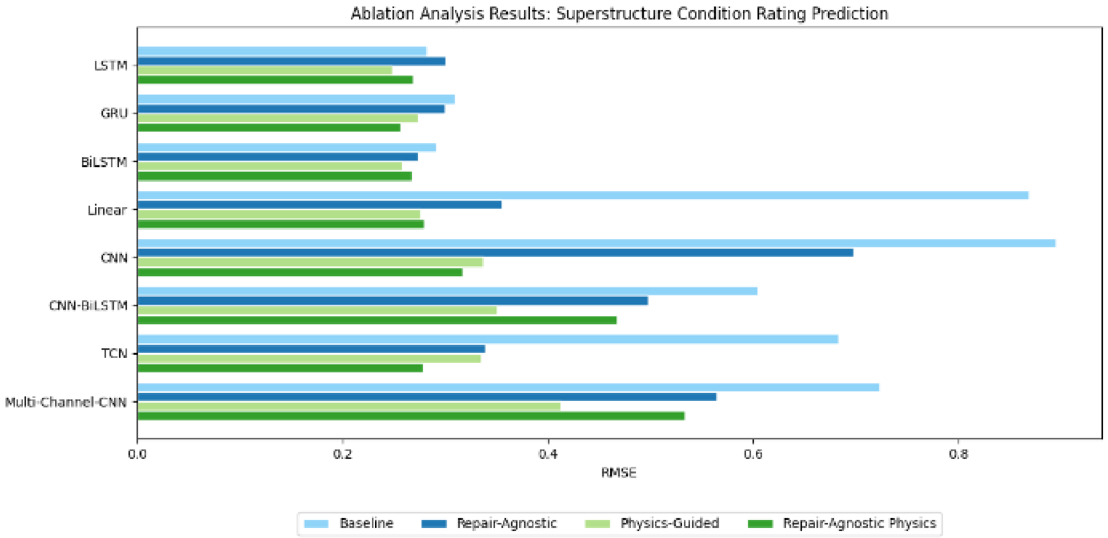


Figure 33. Superstructure: Comparison of Ablation Analysis

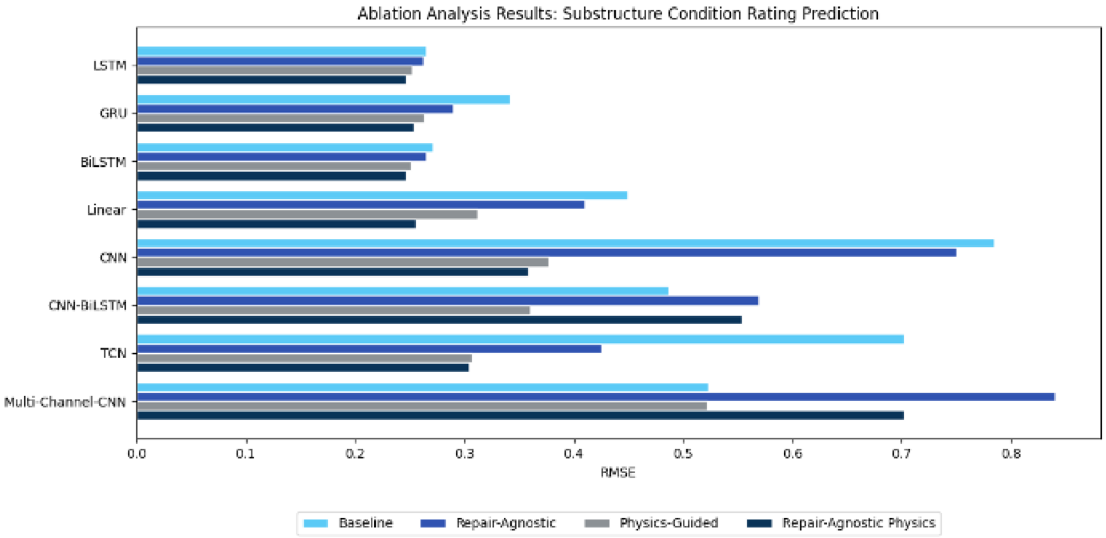


Figure 34. Substructure: Comparison of Ablation Analysis

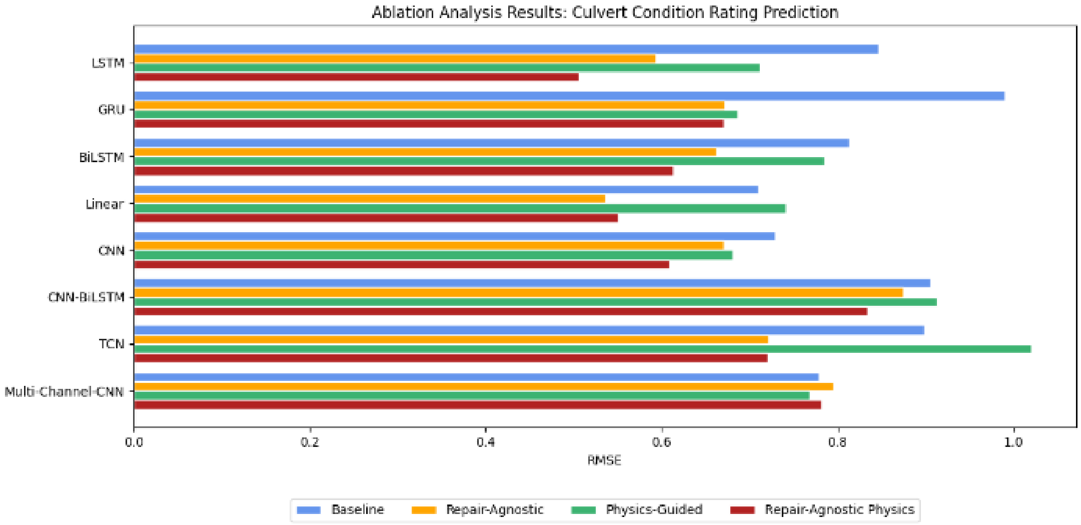


Figure 35. Culvert: Comparison of Ablation Analysis

3.4.4 Discussion

Our comparative and ablation analysis illustrates that LSTM, BiLSTM, and GRU are the most prominent models for forecasting the condition of bridge components. Meanwhile, Linear Regression remains a satisfactory choice for culvert modeling under baseline configurations, as summarized in Table 10. The superior performance of these recurrent models can be attributed to their recurrent gated architecture, which effectively captures complex, time-dependent relationships. The input, forget, and output gates enable the models to dynamically retain or discard information across time steps, making them particularly well-suited for temporal deterioration data.

Furthermore, appending physics-based loss functions provides an inductive bias aligned with real-world behavior, accelerating the model's learning ability. Repair data complements this by injecting historical intervention context into the learning process. Using temporal memory, physical constraints, and repair history significantly improves prediction accuracy and robustness, supporting informed maintenance planning and resource allocation.

Table 10. Best-Performing Models for Structural Condition Rating by Component and Configuration

Component	**Data-Driven**	**Repair-Agnostic**	**Physics-Guided**	**RAPG**
Deck	LSTM	GRU	GRU	GRU
Superstructure	LSTM	BiLSTM	LSTM	GRU

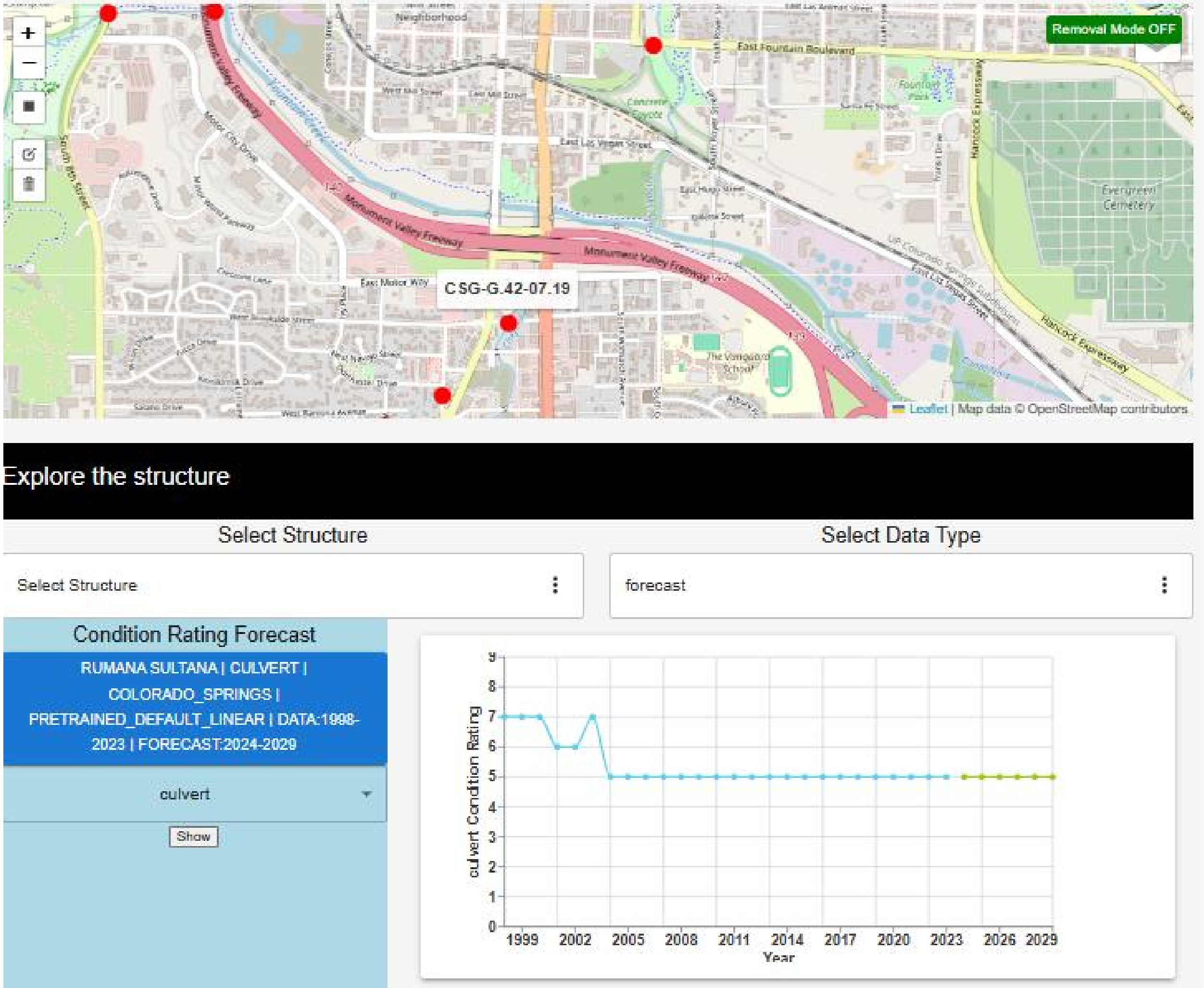
Substructure	BiLSTM	GRU	BiLSTM	LSTM
Culvert	Linear	LSTM	GRU	LSTM

3.5 Sample Forecasting Results

The forecasting results presented in Figures 36–38 are generated using the Make Forecast functionality of the i-BM platform. This functionality enables users to interactively produce multi-year condition forecasts by selecting a structure, structural component, trained model, and prediction horizon through the web-based interface.

When a user initiates the Make Forecast action, the system dynamically retrieves the relevant historical inspection, traffic, and environmental data from the backend databases based on the selections made through the frontend interface. The corresponding trained forecasting model is then loaded from the MLflow model registry and executed to generate future condition ratings over the specified prediction horizon. The forecasting workflow supports both data-driven and physics-guided models, as well as repair-aware and non-repair configurations, enabling flexible scenario analysis. The resulting forecasts can be examined directly within the i-BM data explorer interface or exported as Excel files for offline analysis and documentation.

Figure 36 illustrates sample forecasting results visualized directly within the i-BM tool after execution. In this case, the predicted condition ratings for a selected culvert structure are displayed as a time-series plot, where historical inspection data are shown alongside future predictions. Forecasted values have persisted in the database, enabling interactive visualization and comparison across years for further analysis.



Explore the structure

Select Structure: Select Structure

Select Data Type: forecast

Condition Rating Forecast

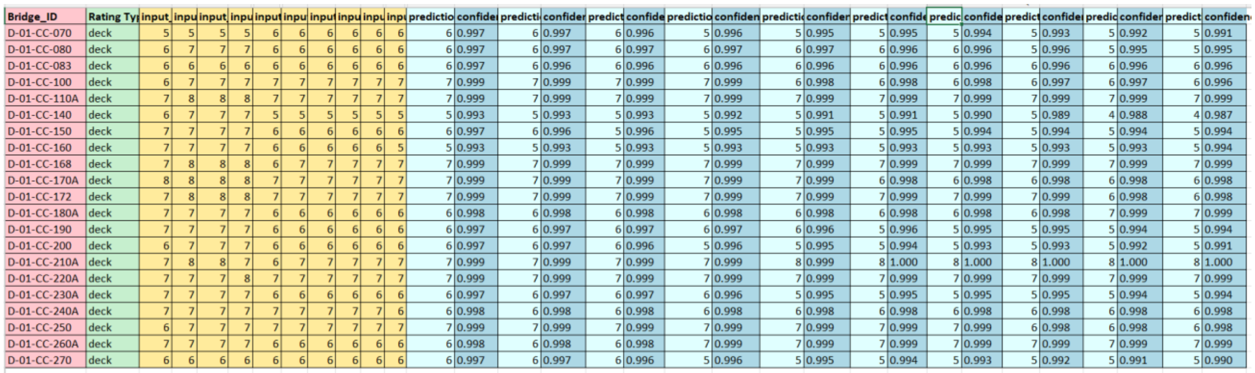
RUMANA SULTANA | CULVERT | COLORADO_SPRINGS | PRETRAINED_DEFAULT_LINEAR | DATA:1998-2023 | FORECAST:2024-2029

culvert

Show

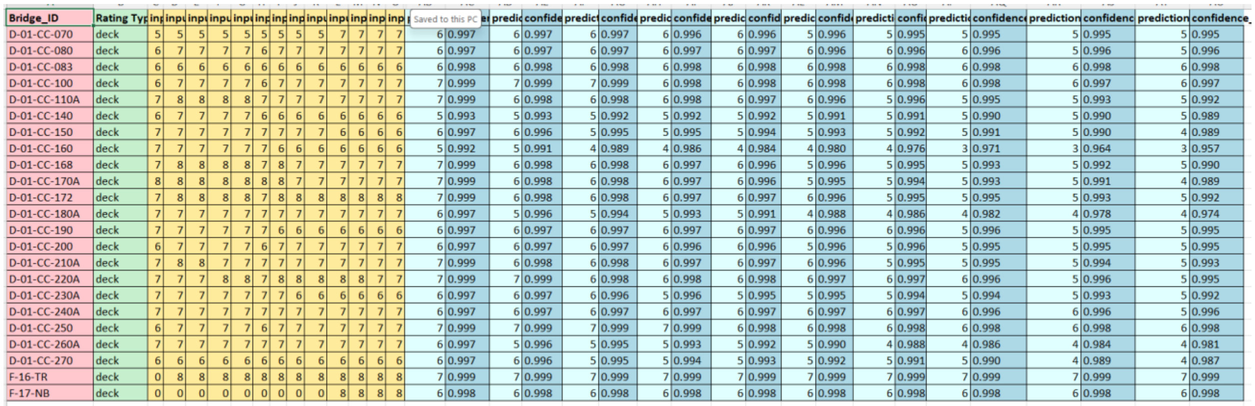
Figure 36. Sample Forecasting Results Visualized from the Database in the i-BM Tool

In addition to in-platform visualization, the i-BM system allows forecast results to be exported as structured files for offline analysis and documentation. Figures 37 and 38 present examples of long-term forecast results exported to files for two different modeling configurations. Figure 37 presents long-term condition rating forecasts generated using a non-repair, non-physics (purely data-driven) model. The figure illustrates how future condition ratings are predicted based solely on historical deterioration patterns observed in the input data, without explicitly accounting for repair actions or enforcing physics-based constraints. As reflected in the tabular representation, each row corresponds to an individual bridge deck identified by a unique Bridge_ID. The left portion of the table contains historical condition ratings across multiple years (highlighted in yellow), which serve as model inputs, while the right portion shows the predicted future condition ratings along with their associated confidence values for each forecasted year (displayed as alternating prediction and confidence columns shaded in light blue). This visualization demonstrates how the data-driven model extrapolates long-term deterioration trends directly from past observations. In contrast, Figure 38 demonstrates results produced using a repair-agnostic physics-guided model, which incorporates domain knowledge to regulate deterioration trends and enforce physically meaningful behavior over extended forecast horizons.



Bridge_ID	Rating Typ	input_	inpu	input_	inpu	input	inpu	input	inpu	inpu	inpu	predictio	confider	predicti	confider	predict	confide	predictio	confiden	predicti	confider	predict	confide	predic	confide	predict	confider	predic	confider	predict	confiden
D-01-CC-070	deck	5	5	5	5	6	6	6	6	6	6	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995	5	0.995	5	0.994	5	0.993	5	0.992	5	0.991
D-01-CC-080	deck	6	7	7	7	6	6	6	6	6	6	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	5	0.996	5	0.995	5	0.995
D-01-CC-083	deck	6	6	6	6	6	6	6	6	6	6	6	0.997	6	0.996	6	0.996	6	0.996	6	0.996	6	0.996	6	0.996	6	0.996	6	0.996	6	0.996
D-01-CC-100	deck	6	7	7	7	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998	6	0.997	6	0.997	6	0.996
D-01-CC-110A	deck	7	8	8	8	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999
D-01-CC-140	deck	6	7	7	7	5	5	5	5	5	5	5	0.993	5	0.993	5	0.993	5	0.992	5	0.991	5	0.991	5	0.990	5	0.989	4	0.988	4	0.987
D-01-CC-150	deck	7	7	7	7	6	6	6	6	6	6	6	0.997	6	0.996	5	0.996	5	0.995	5	0.995	5	0.995	5	0.994	5	0.994	5	0.994	5	0.994
D-01-CC-160	deck	7	7	7	7	6	6	6	6	6	5	5	0.993	5	0.993	5	0.993	5	0.993	5	0.993	5	0.993	5	0.993	5	0.993	5	0.993	5	0.994
D-01-CC-168	deck	7	8	8	8	6	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999
D-01-CC-170A	deck	8	8	8	8	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998
D-01-CC-172	deck	7	8	8	8	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998
D-01-CC-180A	deck	7	7	7	7	6	6	6	6	6	6	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	7	0.999	7	0.999
D-01-CC-190	deck	7	7	7	7	6	6	6	6	6	6	6	0.997	6	0.997	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995	5	0.995	5	0.994	5	0.994
D-01-CC-200	deck	6	7	7	7	6	6	6	6	6	6	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995	5	0.994	5	0.993	5	0.993	5	0.992	5	0.991
D-01-CC-210A	deck	7	8	8	7	6	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	8	0.999	8	1.000	8	1.000	8	1.000	8	1.000	8	1.000
D-01-CC-220A	deck	7	7	7	8	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999
D-01-CC-230A	deck	7	7	7	7	6	6	6	6	6	6	6	0.997	6	0.997	6	0.997	6	0.996	5	0.995	5	0.995	5	0.995	5	0.995	5	0.994	5	0.994
D-01-CC-240A	deck	7	7	7	7	7	7	7	7	7	6	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998
D-01-CC-250	deck	6	7	7	7	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998
D-01-CC-260A	deck	7	7	7	7	6	6	6	6	6	6	6	0.998	6	0.998	6	0.998	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999
D-01-CC-270	deck	6	6	6	6	6	6	6	6	6	6	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995	5	0.994	5	0.993	5	0.992	5	0.991	5	0.990

Figure 37. Sample Long-Term Forecast Results Using a Non-Repair, Non-Physics Model (Exported to File)



Bridge_ID	Rating Typ	inp	inpu	inpu	inpu	inpu	inp	inp	inp	inpu	inpu	inp	inp	inp	p Saved to this PC	er	predic	confide	predict	confide	predic	confide	predic	confid	predic	confid	predicti	confi	predicti	confidence	prediction	confidenc	prediction	confidence_
D-01-CC-070	deck	5	5	5	5	5	5	5	5	5	7	7	7	7	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	5	0.996	5	0.995	5	0.995	5	0.995	5	0.995
D-01-CC-080	deck	6	7	7	7	7	6	7	7	7	7	7	7	7	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	6	0.996	5	0.996	5	0.996
D-01-CC-083	deck	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998
D-01-CC-100	deck	6	7	7	7	7	6	7	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.997	6	0.997
D-01-CC-110A	deck	7	8	8	8	8	7	7	7	7	7	7	7	7	7	0.999	6	0.998	6	0.998	6	0.998	6	0.997	6	0.996	5	0.996	5	0.995	5	0.993	5	0.992
D-01-CC-140	deck	6	7	7	7	7	6	6	6	6	6	6	6	6	5	0.993	5	0.993	5	0.992	5	0.992	5	0.992	5	0.991	5	0.991	5	0.990	5	0.990	5	0.989
D-01-CC-150	deck	7	7	7	7	7	7	7	7	7	6	6	6	6	6	0.997	6	0.996	5	0.995	5	0.995	5	0.994	5	0.993	5	0.992	5	0.991	5	0.990	4	0.989
D-01-CC-160	deck	7	7	7	7	7	7	6	6	6	6	6	6	6	5	0.992	5	0.991	4	0.989	4	0.986	4	0.984	4	0.980	4	0.976	3	0.971	3	0.964	3	0.957
D-01-CC-168	deck	7	8	8	8	8	7	8	7	7	7	7	7	7	7	0.999	6	0.998	6	0.998	6	0.997	6	0.996	5	0.996	5	0.995	5	0.993	5	0.992	5	0.990
D-01-CC-170A	deck	8	8	8	8	8	8	8	7	7	7	7	7	7	7	0.999	6	0.998	6	0.998	6	0.997	6	0.996	5	0.995	5	0.994	5	0.993	5	0.991	4	0.989
D-01-CC-172	deck	7	8	8	8	8	7	8	8	8	8	8	8	8	7	0.999	6	0.998	6	0.998	6	0.997	6	0.997	6	0.996	5	0.995	5	0.995	5	0.993	5	0.992
D-01-CC-180A	deck	7	7	7	7	7	7	7	7	7	7	7	7	7	6	0.997	5	0.996	5	0.994	5	0.993	5	0.991	4	0.988	4	0.986	4	0.982	4	0.978	4	0.974
D-01-CC-190	deck	7	7	7	7	7	7	6	6	6	6	6	6	6	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	5	0.996	5	0.995	5	0.995
D-01-CC-200	deck	6	7	7	7	7	6	7	7	7	7	7	7	7	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	5	0.996	5	0.996	5	0.995	5	0.995	5	0.995
D-01-CC-210A	deck	7	8	8	7	7	7	7	7	7	7	7	7	7	7	0.999	6	0.998	6	0.998	6	0.997	6	0.997	6	0.996	5	0.995	5	0.995	5	0.994	5	0.993
D-01-CC-220A	deck	7	7	7	8	8	7	8	8	8	8	8	7	7	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995
D-01-CC-230A	deck	7	7	7	7	7	7	7	6	6	6	6	6	6	6	0.997	6	0.997	6	0.996	5	0.996	5	0.995	5	0.995	5	0.994	5	0.994	5	0.993	5	0.992
D-01-CC-240A	deck	7	7	7	7	7	7	7	7	7	7	7	7	7	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.997	6	0.996	6	0.996	5	0.996
D-01-CC-250	deck	6	7	7	7	7	6	7	7	7	7	7	7	7	7	0.999	7	0.999	7	0.999	7	0.999	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998
D-01-CC-260A	deck	7	7	7	7	7	7	7	7	7	7	7	7	7	6	0.997	5	0.996	5	0.995	5	0.993	5	0.992	5	0.990	4	0.988	4	0.986	4	0.984	4	0.981
D-01-CC-270	deck	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0.997	6	0.996	5	0.995	5	0.994	5	0.993	5	0.992	5	0.991	5	0.990	4	0.989	4	0.987
F-16-TR	deck	0	8	8	8	8	8	8	8	8	8	8	8	8	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999	7	0.999
F-17-NB	deck	0	0	0	0	0	0	0	0	0	8	8	8	8	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998	6	0.998

Figure 38. Sample Long-Term Forecast Results Using a Repair-Agnostic Physics Model (Exported to File)

Together, these results highlight the flexibility of the Make Forecast functionality in supporting multiple modeling paradigms and output formats.By enabling both in-tool visualization and file-based export, the i-BM platform facilitates detailed examination of long-term deterioration trends and supports data-driven decision-making for infrastructure maintenance planning and management.

Chapter 4. Bridge Anomaly Detection

4.1 Problem Definition

Time-series anomaly detection aims to identify abnormal patterns or events that deviate significantly from normal temporal behavior. These anomalies often correspond to system faults, data errors, or rare operational events. While deep learning–based methods such as CNN, LSTM, and Transformer architectures have achieved strong detection performance, their success heavily depends on the availability of large, high-quality labeled datasets. Unfortunately, manual labeling of time-series anomalies is costly, time-consuming, and ambiguous, since abnormal behaviors vary in scale, duration, and semantics. Traditional unsupervised methods (e.g., Isolation Forest, Autoencoder, LOF) require no labels but often yield suboptimal and unstable results. Meanwhile, fully supervised models demand extensive labeling that is impractical in industrial contexts.

To address this labeling bottleneck, Haotian Guo et al. in LEIAD [59] defined a new research objective:

Develop a label-efficient anomaly detection framework that minimizes manual supervision while maintaining high detection accuracy.

Accordingly, we define the anomaly detection problem as follows:

Given a multivariate time-series dataset $X=\{x_1,x_2,\ldots,x_T\}$, detect anomalies $A=\{a_1,a_2,\ldots,a_T\}$ with minimal human-labeled samples, leveraging weak supervision, active learning, and heuristic labeling functions.

4.2 Literature Review

In this section, we review existing methods that are commonly applied in anomaly detection research and practice.

4.2.1 Unsupervised Anomaly Detection (UAD)

Unsupervised methods detect anomalies by modeling normal behavior and identifying deviations. Classic UAD approaches include:

- **Isolation Forest (I-Forest):** builds isolation trees that separate abnormal points with fewer splits, producing anomaly scores based on path lengths.

- **Spectral Residual (SR):** applies Fourier Transform to filter frequency-domain signals and reconstructs time-domain residuals for anomaly detection.
- **STL (Seasonal-Trend Decomposition):** separates trend and seasonal components, marking residual outliers as anomalies.
- **RC-Forest (Random Cut Forest):** builds ensembles of random cut trees to isolate anomalies probabilistically.
- **Luminol:** a lightweight library by LinkedIn used for streaming anomaly detection.

Each of these UAD models has specific assumptions about data distribution and works independently. However, no single model performs optimally across diverse datasets or scenarios. LEIAD [59] mitigates this limitation by aggregating multiple UAD outputs to produce initial pseudo-labels that are then refined interactively.

4.2.2 Active Learning

Active learning reduces annotation cost by querying labels for only the most informative samples. Instead of labeling all data, a model iteratively selects uncertain points and requests user feedback to improve performance.

However, existing active learning frameworks face two issues:

1. **Cold-start problem** – initial labeled data are needed to bootstrap the model.
2. **Limited scalability** – most approaches have not been adapted for time-series anomaly detection, where dependencies exist across timestamps.

LEIAD [59] addresses these limitations by integrating **weak supervision** to generate initial pseudo-labels, providing a warm start for the active learning process. Then, active queries refine the model using uncertainty-based feedback from human annotators.

4.2.3 Weak Supervision

Weak supervision provides an efficient means of generating approximate labels using heuristic rules, domain knowledge, or multiple noisy sources instead of manual annotation. Frameworks such as **Snorkel** combine several labeling functions (LFs) to synthesize probabilistic labels.

LEIAD [59] extends weak supervision for time-series data by using:

•Multiple UAD methods as initial labeling functions.

•A **label model** that aggregates these weak labels into probabilistic scores.

•Iterative refinement where new labeling functions are generated from user feedback.

This enables learning even without explicit ground truth, reducing dependency on large-scale manual annotation.

4.3 Proposed Methods

The proposed Label-Efficient Interactive Time-Series Anomaly Detection (LEIAD) [59] system integrates unsupervised detection, weak supervision, active learning, and automatic labeling-function generation into a uniform framework. Its goal is to iteratively improve anomaly detection performance with minimal human supervision by combining automated model outputs with selective expert feedback. The original LEIAD [59] framework was trained and tested using benchmark time-series datasets such as Yahoo, Microsoft, and KPI. In this study, we adapted the same framework to train and detect anomalies in the National Bridge Inventory (NBI) dataset.

4.3.1 System Overview

LEIAD [59] operates as an interactive pipeline that continuously refines its understanding of anomalies over time (see **Figure 8** in section 2.4.1). The workflow consists of four main stages:

The pipeline proceeds as follows:

1. **Initial Label Generation:** Apply multiple UAD models (e.g., I-Forest, SR, STL, RC-Forest, Luminol) to obtain initial pseudo-labels.
2. **Weak Supervision Module:** Aggregate pseudo-labels using a generative label model (Snorkel) that produces probabilistic soft labels.
3. **End Model Training:** Train a supervised anomaly detection model (e.g., LightGBM or DNN) using the aggregated weak labels.
4. **Active Learning Module:** Query human feedback on uncertain or conflicting segments to refine model predictions.
5. **Label Function Generator:** Convert new feedback into additional heuristic LFs, expanding the weak supervision pool.

This loop continues until model predictions stabilize and labeling cost is minimized.

4.3.2 Unsupervised Anomaly Detector (UAD)

Let X={x_1,x_2,…,x_T} denote a time-series. Each UAD method produces anomaly scores $s_t^{(m)}$, where m indexes different detectors. LEIAD normalizes and ensembles these outputs:

$$P(x_t) = \frac{1}{M}\sum_{m=1}^{M} s_t^{(m)} \tag{16}$$

The ensemble score P(x_t) provides initial pseudo-labels for the weak supervision module, enabling cross-model robustness.

4.3.3 Weak Supervision Module

Using **Snorkel**, the weak supervision model integrates multiple LFs to infer probabilistic labels without ground truth.

For each time point x_t, let L=[L_1(x_t),L_2(x_t),…,L_M(x_t)] be the LF votes. The generative model estimates:

$$P_w(Y \mid L) = \frac{1}{Z}\exp\left(\sum_{j=1}^{M} w_j \cdot \phi_j(L, Y)\right) \tag{17}$$

where w_j are model weights and ϕ_j represent label correlations.

This step produces denoised labels that serve as training data for the **end model**.

4.3.4 End Model

The end model is a binary classifier $f_\theta(x_t)$ predicting whether a point is anomalous. It minimizes:

$$\mathcal{L}_{end} = \frac{1}{|L|}\sum_{(x_t,y_t)\in L} CE(f_\theta(x_t), y_t) \tag{18}$$

where CE denotes cross-entropy loss.

LEIAD uses **LightGBM** for its efficiency and robustness on tabular and time-series data. Input features include:

- **Statistical metrics:** mean, variance, skewness, kurtosis, etc.
- **Transformations:** log, trend, and seasonality components.
- **Sliding window statistics:** across multiple scales (10, 50, 100, 200 timesteps).
- **Ratios/Differences:** relative deviation from previous intervals.

4.3.5 Active Learning Module

The active learning agent selects the most informative samples based on four metrics:

1.**Agreement of Labeling Functions**

$$A(x_t) = -\sum_{l=1}^{M} [p_l(x_t) \log\, p_l(x_t) + (1 - p_l(x_t)) \log(1 - p_l(x_t))] \tag{19}$$

2.**Abstention Count**

$$H(x_t) = \log(\mathrm{count}_{LF}(x_t) + 1) \tag{20}$$

3.**Uncertainty of End Model**

$$U(x_t) = -[p_\theta(x_t) \log\, p_\theta(x_t) + (1 - p_\theta(x_t)) \log(1 - p_\theta(x_t))] \tag{21}$$

4.**Diversity**

$$D(x_t) = 1 - \frac{1}{|S^l|} \sum_{x_k \in S^l} \mathrm{sim}(x_t, x_k) \tag{22}$$

The combined **query function** determining which samples to label is:

$$Q(x_t) = \alpha A(x_t) + \beta H(x_t) + \gamma U(x_t) + \delta D(x_t) \tag{23}$$

where coefficients α, β, γ, and δ control the trade-off among criteria.

4.3.6 Label Function Generation

Once human feedback is collected, LEIAD [59] automatically transforms it into new labeling functions using time-series embedding similarity.

Given an annotated pair (x_i,y_i), similar instances are found via cosine distance in a learned embedding space:

$$\mathrm{sim}(x_i, x_j) = \| r(x_i) - r(x_j) \|$$

If similarity exceeds a threshold, a new LF is created to generalize that labeling rule across the dataset.

This adaptive mechanism enables continual improvement of the weak supervision pool, enhancing scalability and reducing manual labeling costs.

4.3.7 Advantages of LEIAD

- **Label-efficient:** minimizes human annotation through interactive learning.
- **Hybrid learning:** combines unsupervised, weakly supervised, and active learning paradigms.
- **Adaptable:** automatically generates new heuristic rules from user feedback.
- **Industrial scalability:** validated on Microsoft’s monitoring datasets.

4.5 Tool Description and Sample Results

This section provides an overview of the implemented interface and illustrative results obtained from the i-BM tool.

4.5.1 Anomaly Analysis Model Management Interface

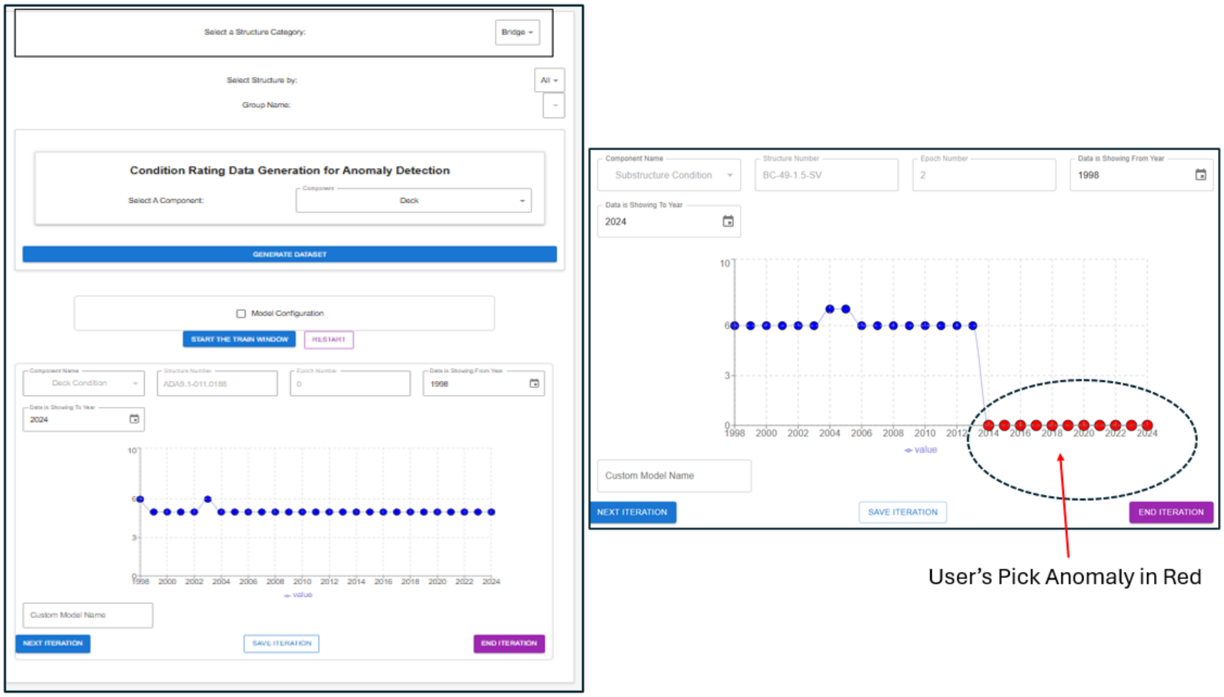
The Anomaly Analysis Model Management Window provides an interactive interface for generating datasets, configuring models, and training anomaly detection algorithms on bridge and culvert condition data. Users can begin by selecting the Structure Category (e.g., Bridge or Culvert) and the Group Name or structure selection criteria. The interface then allows specification of the Component to be analyzed (such as Deck, Superstructure, Substructure, or Culvert) under the Condition Rating Data Generation panel.

After selecting the desired component, users can click Generate Dataset to automatically retrieve condition rating data for the selected structure from the database. For advanced configuration, the Model Configuration option allows users to optionally adjust training parameters such as Epoch Number, Warm or Cold Start, or others. The **Start the Train Window button** initiates the anomaly detection model training, while the **Save Iteration** button stores the current training progress for future continuation. The **Restart** button enables users to resume training from a previously saved incomplete iteration, ensuring flexibility and continuity during multi-stage training processes.

During the iterative training process, one bridge time-series sequence is displayed per iteration. The user can visually inspect this sequence and mark potential anomalies based on the plotted data. Once an anomaly is identified and confirmed in the **Next Iteration** button, the model retrains in the backend using the user’s labeling feedback and then automatically presents the next most likely bridge candidate for anomaly review. This iterative process continues until the user is satisfied with the model's performance or when no anomalous patterns are detected after several consecutive iterations.

Once the training is complete, the user can finalize and save the trained model by clicking **End Iteration**, which stores the final model parameters for future use to detect anomalies from unseen data using the i-BM Anomaly Detection interface.

This module makes the anomaly detection workflow interactive, explainable, and data-driven—empowering bridge engineers to iteratively refine and train models that can automatically identify abnormal performance behaviors or unexpected deterioration patterns across bridge components. **Figure 39** shows the model management interface in the i-BM system.



Select a Structure Category: Bridge

Select Structure by: All

Group Name:

Condition Rating Data Generation for Anomaly Detection

Select A Component: Component: Deck

□ Model Configuration

Component Name: Deck Condition; Structure Number: ADA9.1-011.0188; Epoch Number: 0; Data is Showing From Year: 1998; Data is Showing To Year: 2024

Component Name: Substructure Condition; Structure Number: BC-49-1.5-SV; Epoch Number: 2; Data is Showing From Year: 1998; Data is Showing To Year: 2024

User's Pick Anomaly in Red

Figure 39. Model Management Interface to Train an Anomaly Model

4.5.2 Anomaly Detection Interface

The anomaly detection interface enables users to identify abnormal behavior in bridge component condition ratings using trained anomaly detection models. Users can begin by selecting the structure category (e.g., Bridge or Culvert) and choosing a group name or all structures for evaluation. Components available for anomaly analysis—Deck, Superstructure, Substructure, and Culvert—can be selected through the Component Selection panel.

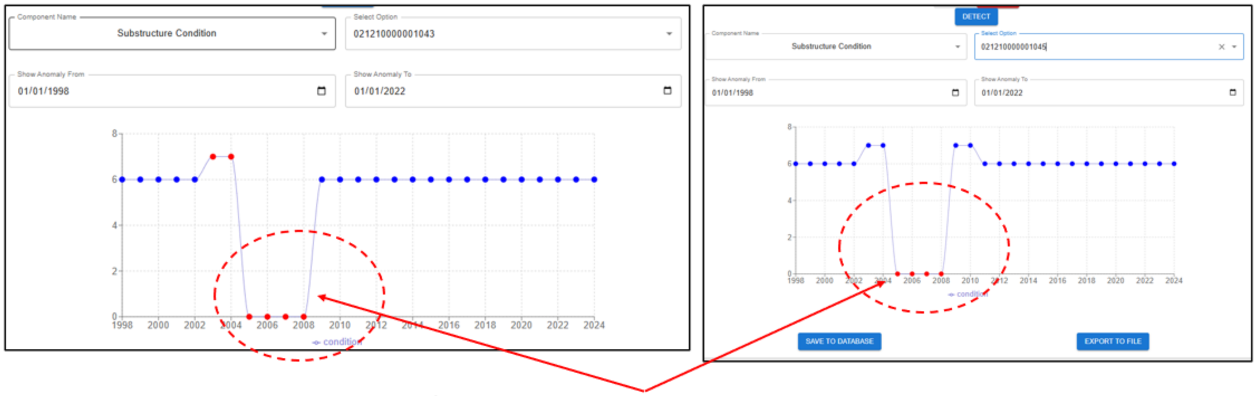
After component selection, users can choose a trained model and click on Detect. After clicking Detect, the system executes the selected anomaly detection model in the backend and visualizes the resulting condition rating time-series on a dynamic chart. Users can choose an individual bridge from a dropdown list to show and analyze.

Anomalous points are automatically highlighted in red, while normal data points are shown in blue, allowing bridge engineers to visually inspect periods of abnormal trends or data irregularities. Detected anomalies may indicate unusual deterioration patterns, measurement inconsistencies, or potential inspection/reporting errors that require further review.

Once the detection process is completed, users can either save results to the database for further analysis or export the results to a file for external documentation or reporting. This streamlined interface enhances the usability of the anomaly detection module, enabling engineers to quickly identify, validate, and record performance anomalies in bridge and culvert components across multiple structures.

4.5.3 Sample Results

Figure 40 shows some sample results generated by the i-BM anomaly detection model.



Component Name: Substructure Condition
Select Option: 021210000001043
Show Anomaly From: 01/01/1998
Show Anomaly To: 01/01/2022

Component Name: Substructure Condition
Select Option: 021210000001045
Show Anomaly From: 01/01/1998
Show Anomaly To: 01/01/2022

Automatically detected by the Model from unseen data

Figure 40. Sample Results from i-BM Anomaly Detection

Chapter 5. Conclusions and Future Work

In this study, we designed, developed, and deployed an end-to-end intelligent bridge management tool (i-BM) for bridge and culvert deterioration forecasting and anomaly detection to be used by CDOT bridge engineers as a tool for effective bridge management. I-BM integrates a series of data-driven deep learning models and their physics-guided extension DL models for bridge deterioration forecasting, as well as an interactive training framework for bridge anomaly detection. Before integrating the models into i-BM, we conducted extensive experimental evaluations using multimodal real-world datasets, including bridge performance, traffic, and weather data for all bridges in Colorado. The results demonstrated that our proposed physics-guided deep learning models significantly outperform existing purely data-driven models previously developed for bridge and culvert deterioration forecasting. Moreover, the standalone software package allows bridge engineers to either use pre-trained models or train their own models by selecting from approximately 32 model configurations with different combinations of input features, enabling comprehensive training, forecasting, and evaluation of predictive performance. This tool was developed by building upon and extending our previous work in three key ways: 1) it integrates the deep learning models into a user-friendly software tool with graphical user interface and improved operational features to improve usability and functionality, 2) it incorporates enhanced physics-guided deep learning models that integrate traditional physics based bridge deterioration forecasting models with data-driven deep learning models for further improved performance in prediction of deterioration, and 3) it develops bridge performance anomaly detection that allows for accurate prediction of bridge performance anomalies such as those that can lead to bridge failures/accidents.

In the future, we plan to integrate work-order optimization into the current Intelligent Bridge Management (i-BM) tool to enable cost-effective maintenance of bridges and culverts using both component-level and element-level deterioration forecasting. We also aim to extend this tool into a comprehensive structural management platform that serves as an intelligent assistant for bridge engineers, incorporating more advanced technologies.

6. References

[1] United States Department of Transportation Fed. Hw, "National Bridge Inventory," 1992. [Online]. Available: https://www.fhwa.dot.gov/bridge/nbi/ascii.cfm.

[2] National Center for Environmental Information, "Climate Data Online Search," 1981. [Online]. Available: https://www.ncdc.noaa.gov/cdo-web/datasets.

[3] I. Srikanth and M. Arocklasami, "Deterioration Models for Prediction of Remaining Useful Life of Timber and Concrete Bridges: A Review," Journal of Traffic and Transportation Engineering, 2019.

[4] H. Liu and Y. Zhang, "Bridge Condition Rating Data Modeling using Deep Learning Algorithm," Structure and Infrastructure Engineering, 2020.

[5] G. Morcous, "Comparing the Use of Artificial Neural Networks and Case-Based Reasoning in Modeling Bridge Deterioration," Annual Conference of the Canadian Society for Civil Engineering, 2002.

[6] O. Tokdemir, C. Ayvalik and J. Mohammadi, "Prediction of Highway Bridge Performance by Artificial Neural Networks and Genetic Algorithms," 2000 Proceedings of the 17th ISARC, Taipei, Taiwan, 2000.

[7] E. Zivot, "Vector Autoregressive Models for Multivariate Time Series," University of Washington Econ 584 Notes.

[8] K. Gulden and G. Nese, "A Study on Multiple Linear Regression Analysis," Procedia - Social and Behavioral Sciences, vol. 106, pp. 234-240, 2013.

[9] D. Rumelart, G. Hinton and R. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533-536, 1986.

[10] I. Goodfellow and Y. Bengio, Deep Learning, MIT Press, 2016.

[11] C. Olah, "Understanding lstm networks".

[12] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," IEE Transactions on Signal Processing, vol. 45, pp. 2673-2681, 1997.

[13] S. Bai, Z. Kolter and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv:1803.01271, 2018.

[14] J. Long, E. Shelhamer and D. Trevor, "Fully convolutional networks for semantic segmentation," CVPR, 2015.

[15] F. Lassig, "Temporal Convolutional Networks and Forecasting," Unit8, 2021.

[16] Y. Zheng, Q. Liu, E. Chen, Y. Ge and J. Zhao, "Time Series Classification using Multi-Channels Deep Convolutional Neural Networks," Lecture Notes in Computer Science, p. 298*310.

[17] P. Chetti and H. Ali, "Analyzing the Structural Health of Civil Infrastructures using Correlation Networks and Population Analysis," in The Eighth International Conference on Data Analytics, 2019.

[18] K. &. C. S. Chang, "Bridge Clustering for Systematic Recognition of Damage Patterns on Bridge Elements," Journal of Computing in Civil Engineering, vol. 33, no. 5, 2019.

[19] P. Diez, "A clustering approach for structural health monitoring on bridges," Journal of Civil Structural Health Monitoring, vol. 6, no. 3, pp. 429-445, 2016.

[20] M. &. C. B. Knight, "Infrastructure Investigation Using Latent Class Cluster Analysis," in International Conference on Computing in Civil Engineering, 2005.

[21] E. Woods, "Tslearn, A Machine Learning Toolkit for Time Series Data," Journal of Machine Learning Research, vol. 21, pp. 1-6, 2020.

[22] S. M. S. D. F. a. H. K. Naveen Sai Madiraju, "Deep temporal clustering: Fully unsupervised learning of time-domain features," arXiv:1802.01059, 2018.

[23] W. M. S. D. a. H. B. Toon Van Craenendonck, "COBRASTS: A new approach to Semi-Supervised Clustering of Time Series," arXiv:1805.00779, 2018.

[24] Hearn, G., "Deteriorating Modeling for Highway Bridges", Structural Reliability in Bridge Engineering, Workshop a

[25] Rens, K. L., Nogueira, C. L., Transue, D. J., "Bridge Management and Nondestructive Evaluation", Journal of Performance of Constructed Facilities, Vol. 19, N. 1, 3-16, EUA, February, 2005.

[26] Rens, K. L., Nogueira, C. L., Neiman Y. M., Gruber, T., and Johnson, L. E., "Bridge Management System for the City and County of Denver", Practice Periodical on Structural Design and Construction, Vol. 4, N. 4, 131-136, EUA, November, 1999.

[27] Hartle, R. A., Amrhein, W. J., Wilson III, K. E., and Baughman, D. R., "Bridge Inspector's Training Manual 90", Federal Highway Administration, U. S. Department of Transportation, July 1991.

[28] REMR Technical Note OM-CI-1.2, "The REMR Condition Index, Condition Assessment for Maintenance Management of Civil Works Facilities", Suppl. 7, 1996.

[29] Stecker, J., Greimann, L. F., and Rens, K. L., "Chicago Harbor Lock Inspection", Technical Report submitted to U.S. Army Corps of Engineers, Chicago District, 1993.

[30] Greimann, L. F., Stecker, J., and Rens, K. L., "Inspection and Rating of Miter Lock Gates", Technical Report REMR-OM-7, 1990.

[31] Greimann, L. F., Stecker, J., and Rens, K. L., "REMR Management Systems Navigation Structures, Condition Rating Procedures for Sector Gates", REMR-OM-13, 1993.

[32] Greimann, L. F., Stecker, J., and Rens, K. L., "Training Manual for Inspection and Rating of Miter Gates, Sector Gates, Steel Sheet Pile, and Operating Equipment", U.S. Army Corps of Engineers, Construction Engineering Research Laboratories (CERL), Champagne, Illinois, 1992.

[33] Greimann, L. F., Stecker, J., Rens, K. L., and Nop, M., "REMR Management Systems Navigation Structures, User's Manual for Inspection and Rating Software, Version 2.0", REMR-OM-15, 1994.

[34] Hearn, G. and Shim, H.-S., "Integration of Nondestructive Evaluation Methods and Bridge Management Systems", Infrastr. Condition Assessment: Art, Science & Practice, ASCE Conference, Boston, 1997, pp. 464-473.

[35] Hearn, G. and Shim, H.-S., “Integration of Bridge Management Systems and Nondestructive Evaluations”, Journal of Infrastructure Systems, Vol. 4, No. 2, June 1998, pp. 49-55.

[36] Bulusu, S. and Sinha, K. C., “Comparison of Methodologies to Predict Bridge Deterioration”, Transportation Research Record 1597, 1997, pp. 34-42.

[37] White, III, C. C. and White, D. J., “Markov Decision Processes”, European Journal of Operational Research, 39, 1989, pp. 1-16.

[38] Scherer, W. T. and Glagola, D. M., “Markovian Model for Bridge Maintenance Management”, Journal of Transportation Research, Vol. 20, No. 1, 1994, pp. 37-51.

[39] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, Oliver Stegle. Deep learning for computational biology. Molecular Systems Biology (2016) 12, 878.

[40] Benjamin Shickel, Patrick James Tighe, Parisa Rashidi. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. DOI:10.1109/JBHI.2017.2767063. IEEE Journal of Biomedical and Health Informatics, 2018.

[41] W. Byeon, T. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In CVPR, 2015.

[42] M. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009.

[43] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In CVPR, 2015.

[44] R. Girshick. Fast r-cnn. In ICCV, 2015.

[45] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In ICML, 2015.

[46] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. arXiv:1412.4729, 2014

[47] Riccardo Miotto, Li Li, Joel T. Dudley. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. DOI:10.1038/srep26094. Scientific reports, 2016.

[48] Nguyen, R. Alqurashi, Z. Raghebi, F. Banaei-kashani, A. C. Halbower, and T. Vu, “A Lightweight and Inexpensive In-ear Sensing System For Automatic Whole-night Sleep Stage Monitoring”, ACM Conference on Embedded Network Sensor Systems (SenSys 2016). New York, NY, 2016. Best Paper Award

[49] Nguyen, R. Alqurashi, Z. Raghebi, F. Banaei-kashani, A.C. Halbower, T. Dinh, and T. Vu, “In-ear Biosignal Recording System: A Wearable For Automatic Whole-night Sleep Staging”, Workshop on Wearable Systems and Applications (WearSys 2016).

[50] Farnoush B. Kashani, Gerard Medioni, Khanh Nguyen, Luciano Nocera, Cyrus Shahabi, Ruizhe Wang, Cesar E. Blanco, Yi-An Chen, Yu-Chen Chung, Beth Fisher, Sara Mulroy, Philip Requejo, Carolee Winstein. Monitoring mobility disorders at home using 3D visual sensors and mobile sensors. Proceedings of the 4th Conference on Wireless Health, Baltimore, Maryland, 2013.

[51] Max Lee, Farnoush Banaei-Kashani, “Addressing Overfitting in Convolutional Neural Networks for Classification of Multivariate Spatiotemporal Data”, Under review.

[52] Sridharan Raghavan, Farnoush Banaei-kashani, Seth Creasy, Ed Melanson, Leslie Lange, Michael A. Rosenberg, “Activity Classification for Patients with Ventricular Arrhythmia”, accepted for publication at PlosOne.

[53] Banaei-Kashani, F., & Arens, K. (2022). Data-Driven Bridge Management Using Descriptive and Predictive Machine Learning Models (No. CDOT-2022-09). Colorado. Dept. of Transportation. Research Branch.

[54] Nickless, K., “Investigation of Mechanistic Deterioration Modeling for Bridge Design and Management,” Colorado State University, 2017.

[55] Firouzi, A., and Rahai, A., “Reliability Assessment of Concrete Bridges Subject to Corrosion-Induced Cracks During Life Cycle Using Artificial Neural Networks,” Computers and Concrete, Vol. 12, pp. 91–107, 2013.

[56] Huang, Z., Yin, X., and Liu, Y., “Physics-Guided Deep Neural Network for Structural Damage Identification,” Ocean Engineering, Vol. 260, 112073, 2022.

[57] Daw, A., Karpatne, A., Watkins, W. D., Read, J. S., and Kumar, V., “Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling,” in Knowledge-Guided Machine Learning, pp. 353–372, Chapman and Hall/CRC, 2022.

[58] Yousefpour, N., and Wang, B., “Physics-Inspired Deep Learning and Transferable Models for Bridge Scour Prediction,” arXiv preprint arXiv:2407.01258, 2024.

[59] Haotian Guo, Yuxuan Wang, Jiamin Zhang, Zelin Lin, Yuxuan Tong, Luyang Yang, and Changjie Huang, “Label-Efficient Interactive Time-Series Anomaly Detection,” arXiv preprint arXiv:2212.14621, 2022.

[60] Kyösti Tuutti, “Corrosion of Steel in Concrete,” Swedish Cement and Concrete Research Institute, Stockholm, Sweden, 1982.

[61] E. C. Bentz and M. D. A. Thomas, “Life-365 Service Life Prediction Model: Computer Program for Predicting the Service Life and Life-Cycle Costs of Reinforced Concrete Exposed to Chlorides,” 2001.

[62] B. Violetta, “Life-365 Service Life Prediction Model,” Concrete International, Vol. 24, No. 12, pp. 54–57, 2002.

[63] Y. Hu, J. W. Van De Lindt, and J. Li, “Mechanistic Deterioration Modeling for Bridge Design and Management,” Journal of Bridge Engineering, American Society of Civil Engineers (ASCE), Vol. 18, No. 9, pp. 901–910, 2013.

[64] Z. P. Bažant, M. J. Verdure, and M. Kaplan, “Mathematical Model for Freeze–Thaw Damage in Concrete,” Journal of Engineering Mechanics, ASCE, Vol. 114, No. 9, pp. 1670–1688, 1988.

[65] O. B. Isgor and A. G. Razaqpur, “Finite Element Modeling of Coupled Heat Transfer, Moisture Transport and Carbonation Processes in Concrete Structures,” Cement and Concrete Composites, Vol. 26, No. 1, pp. 57–73, 2004.

[66] Z. P. Bažant and S. Baweja, “Creep and Shrinkage Prediction Model for Analysis and Design of Concrete Structures: Model B3,” Materials and Structures, Vol. 28, pp. 357–365, 1995.

[67] Y. Liu and R. E. Weyers, “Modeling the Time-to-Corrosion Cracking in Chloride Contaminated Reinforced Concrete Structures,” ACI Materials Journal, Vol. 95, No. 6, pp. 675–681, 1998.

[68] A. Balafas and C. Burgoyne, “Modeling the Structural Effects of Rust Expansion in Reinforced Concrete,” Cement and Concrete Composites, Vol. 33, No. 9, pp. 917–924, 2011.

[69] Hatami, A., and Morcous, G., “Integrating National Bridge Inventory and Bridge Management System Databases for Improved Bridge Management Decisions,” Journal of Bridge Engineering, ASCE, Vol. 16, No. 3, pp. 313–323, 2011.

[70] Chang, C., Tsai, C., and Chou, C., “Predicting Bridge Condition Ratings Using Data-Driven Regression Models,” Structure and Infrastructure Engineering, Vol. 14, No. 2, pp. 204–215, 2018.

[71] Goyal, A., Morcous, G., and Abdelkader, E., “Predicting Bridge Element Deterioration Using Deterministic and Stochastic Models,” Journal of Bridge Engineering, ASCE, Vol. 22, No. 3, 2017.

[72] Morcous, G., Lounis, Z., and Mirza, M. S., “Comparing Stochastic and Deterministic Approaches to Bridge Deterioration Modeling,” Journal of Bridge Engineering, ASCE, Vol. 7, No. 6, pp. 353–361, 2002.

[73] Morcous, G., “Performance Prediction of Bridge Deck Systems Using Markov Chains,” Journal of Performance of Constructed Facilities, ASCE, Vol. 20, No. 2, pp. 146–155, 2006.

[74] Wellalage, N. M., Thambiratnam, D. P., and Chan, T. H. T., “Bridge Deterioration Forecasting Using Stochastic Modeling,” Structure and Infrastructure Engineering, Vol. 10, No. 3, pp. 355–364, 2014.

[75] Fang, Y., and Sun, L., “Bridge Element-Level Deterioration Prediction Using Hidden Markov Models,” Journal of Bridge Engineering, ASCE, Vol. 23, No. 6, 2018.

[76] Abdelkader, E., Morcous, G., and Lounis, Z., “Network-Level Bridge Deterioration Modeling Using Markov Chains,” Journal of Bridge Engineering, ASCE, Vol. 24, No. 9, 2019.

[77] Sobanjo, J. O., “Artificial Neural Network Approach to Modeling Bridge Deterioration,” Computer-Aided Civil and Infrastructure Engineering, Vol. 12, No. 6, pp. 401–409, 1997.

[78] Tokdemir, O. B., Birgonul, M. T., and Dikmen, I., “Prediction of Bridge Sufficiency Ratings Using Artificial Neural Networks and Genetic Algorithms,” Computer-Aided Civil and Infrastructure Engineering, Vol. 15, No. 6, pp. 373–380, 2000.

[79] Morcous, G., “Comparing Artificial Neural Networks and Case-Based Reasoning for Bridge Condition Prediction,” Journal of Computing in Civil Engineering, ASCE, Vol. 16, No. 4, pp. 273–280, 2002.

[80] Srikanth, V., and Muralidharan, K., “A Review of Artificial Intelligence Applications in Bridge Management Systems,” Automation in Construction, Vol. 120, 152, 2020.

[81] Hearn, G., “Deterioration and Maintenance of Concrete Bridges,” Transportation Research Record: Journal of the Transportation Research Board, No. 1549, pp. 56–64, 1996.

[82] Contreras-Nieto, C. A., Torres-Machi, C., Videla, C., and Maturana, S., “Developing Predictive Models for Bridge Deterioration Using Machine Learning Techniques,” Journal of Infrastructure Systems, ASCE, Vol. 22, No. 4, 2016.

[83] Jonnalagadda, S., Madanat, S., and Peeta, S., “Modeling Bridge Deterioration Using Data-Driven Approaches,” Transportation Research Record: Journal of the Transportation Research Board, No. 2552, pp. 1–8, 2016.

[84] Mia, M. S., and Kameshwar, S., “Machine Learning Framework for Bridge Deterioration Prediction and Uncertainty Quantification,” Engineering Structures, Vol. 278, 115394, 2023.

[85] Rashidi, M., and Elzarka, H., “Optimizing Bridge Deterioration Prediction Models Using Feature Selection and Machine Learning Techniques,” Journal of Infrastructure Systems, ASCE, Vol. 29, No. 2, 2023.

[86] Liu, K., and El-Gohary, N., “Multisource Deep Learning Framework for Bridge Deterioration Prediction,” Computer-Aided Civil and Infrastructure Engineering, Vol. 37, No. 2, pp. 139–157, 2022.

[87] Liu, K., and El-Gohary, N., “Deep Learning-Based Analytics of Multisource Heterogeneous Bridge Inspection Data,” Automation in Construction, Vol. 120, 103374, 2020.

[88] Zhu, M., and Wang, H., “Feature-Aware Deep Neural Networks for Bridge Deterioration Forecasting,” Automation in Construction, Vol. 128, 103759, 2021.

[89] Rajkumar, R., Kar, S., and Dey, N., “Hybrid Deep Learning Approach for Bridge Condition Prediction Using Autoencoder and Random Forest Models,” Expert Systems with Applications, Vol. 228, 120376, 2023.

[90] Jing, X., Li, Z., and Zhang, C., “Long Short-Term Memory Regression Model for Bridge Condition Forecasting,” Engineering Structures, Vol. 300, 117076, 2024.

[91] Miao, T., Liu, K., and Kameshwar, S., “Bridge Deterioration Forecasting Using LSTM-Based Temporal Learning Models,” Automation in Construction, Vol. 154, 105256, 2023.

[92] Abu Dabous, S., Al Jassmi, H., and Al-Tamimi, A., “Modeling Bridge Component Deterioration Using Long Short-Term Memory Networks,” Structure and Infrastructure Engineering, Vol. 20, No. 5, pp. 677–693, 2024.

[93] R. Wang, "Physics-Guided Deep Learning for Dynamical Systems: A Survey," ArXiv, 2021.

[94] R. Wang, K. Kashinath, M. Mustafa, A. Albert and R. Yu, "Towards Physics-Informed Deep Learning for Turbulent Flow Prediction," Proceedings of the 26thy ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020.

[95] D. Wu, L. Gao, X. Xiong, M. Chinazzi, A. Vespignani, Y. Ma and R. Yu, "DeepGLEAM: A Hybrid Mechanistic and Deep Learning Model for COVID-19 Forecasting.," ArXiv, 2021.