Structural Health Monitoring (SHM) System Project Report

By Data Science Gamma Team

Problem Statement

Structural Health Monitoring (SHM) systems play a critical role in ensuring the safety of bridges, buildings, and other infrastructure. The goal of this project was to develop a structural health monitoring system capable of detecting anomalies in buildings, bridges, and other critical infrastructure. Traditional anomaly detection methods primarily rely on vibration data alone. However, anomalies in structures are often influenced by multiple factors such as temperature fluctuations, wind pressure, and displacement. This project aims to address this limitation by designing a robust SHM system capable of analyzing multi-modal sensor data for accurate anomaly detection**.**

Proposed Solution

1. Data Acquisition

The dataset used for this project was the **Bridge Sensor Dataset**, sourced from the **Kaggle platform**. The dataset includes not only vibration measurements but also temperature and windspeed data collected from multiple sensors installed on different bridges. The data was resampled at **15-minute intervals** and consists of a total of **1,340 samples.**

A1 | fx | acceleration_x

	A	B	C	D	E	F	G	H	I
1	acceleration_x	acceleration_y	acceleration_z	temperature_c	humidity_percent	wind_speed_mps	fft_peak_freq	fft_magnitude	degra
2	-0.42890208	0.0091630160561347	-0.448429584	24.263204667433644	82.5734763539799	9.129823960618234	3.264360223804593	1.393159023119367	75.15
3	0.0868931336052859	-0.005482015	0.4229731795807928	19.34383842666696	83.54511192850582	6.67618459443649	1.2902510957552502	0.1909759759150162	10.84
4	-0.168531289	-0.703499567	-0.403903128	18.02108244916831	41.88153995517544	1.0841212582924094	4.899751515371277	1.4028450396890637	61.67
5	-0.242926499	-0.046838367	-0.359685418	19.908277910897397	81.78719982929506	0.003721755612151	2.347589644163494	0.6592386479462076	26.57
6	0.1756377321990435	0.0007916719464689	-0.242574132	30.99528642281544	89.394147300635 1	1.0322348117796911	0.9286260965345204	0.7334595466349689	38.94
7	-0.259787757	0.3647989130091563	-0.047853745	27.391412301887826	80.90537452741162	5.584153902116978	1.516210469749629	1.3090070976811965	11.47
8	-0.074894835	-0.059274914	0.1627848622872557	26.337492267471134	34.92240393081248	1.9378151285037348	3.0487082394906606	1.1265959165370412	44.77
9	-0.010518654	-0.238065528	0.4445744804354921	24.08982745007574	64.46455903006995	6.412842218460321	0.6105169614085397	0.4761079972568497	36.62
10	0.0599371751285615	0.3353023559712556	-0.22486475	35.894112566158924	62.35618432388262	4.433310445152855	1.048861889383938	0.5023623555821698	55.22
11	-0.178393454	0.0239273616643467	-0.238199587	15.47217184498405	75.44805493380865	10.603704006891174	4.33648654577363	0.7121099981115404	39.44
12	-0.047667791	-0.325624832	0.066583545234985	29.77891262277787	81.19293129004359	12.721583152342664	1.854870920966473	0.1161408570251796	10.18
13	-0.086493601	0.2495966821911312	-0.169792207	33.95246080253226 5	55.5370108707277	1.4857157312328724	3.662463302491368	1.229999523545933	31.06
14	0.0763003959789012	-0.159946223	-0.869479664	29.264595701653477	72.03681732407296	9.810276196966354	2.932720423297745	0.9017012406454368	80.86
15	-0.056157077	0.1215852680768982	0.5971845831980307	26.48748133561184	48.32035721228847	14.058373323750896	0.5936163638801741	1.4802055791480495	39.43
16	0.0545283496560288	-0.360564854	0.094512189944205	19.609707522042772	58.08458068477162	5.601470963009303	3.201642943349224	0.7099329113824216	82.70
17	-0.274603509	0.2388745184398457	0.0167121116555815	24.761679357287147	89.99741401405504	8.954895084838217	4.032197751491174	0.3249492402560666	64.64
18	-0.510359924	0.0299101520608541	0.0171540386704145	23.3442341976522	69.92681299848499	12.698677310513864	1.1563732767688053	0.7104549824520945	54.54
19	0.1936633058889393	-0.291268226	-0.395257249	20.170232372070966	36.55184702329802	1.12192712188715	3.755940280345596	1.4503997215733366	54.36
20	0.3020185150614979	-0.323347902	0.074894831120987	23.607644620364702	83.70554378025258	7.888558345374262	3.99650655210043	0.533291187322033 5	13.40
21	-0.198506797	-0.29974076	0.5997766417326169	24.4499249466252 5	37.63562315416016	14.816713292067783	3.890440316377936	0.692223200972253 8	41.26

2. Data Preprocessing and Feature Engineering

The data underwent thorough preprocessing and cleaning:

Null values were imputed.

Data distributions were inspected using visualizations.

Three versions of the dataset were created for experimentation:

Version 1: Basic cleaning and inspection.

Version 2: Additional features engineered from raw data.

Version 3: SMOTE and log transformation applied for balanced and normalized data.

Feature engineering included:

Vibration columns (x_axis, y_axis, z_axis): RMS, Peak, Frequency domain features

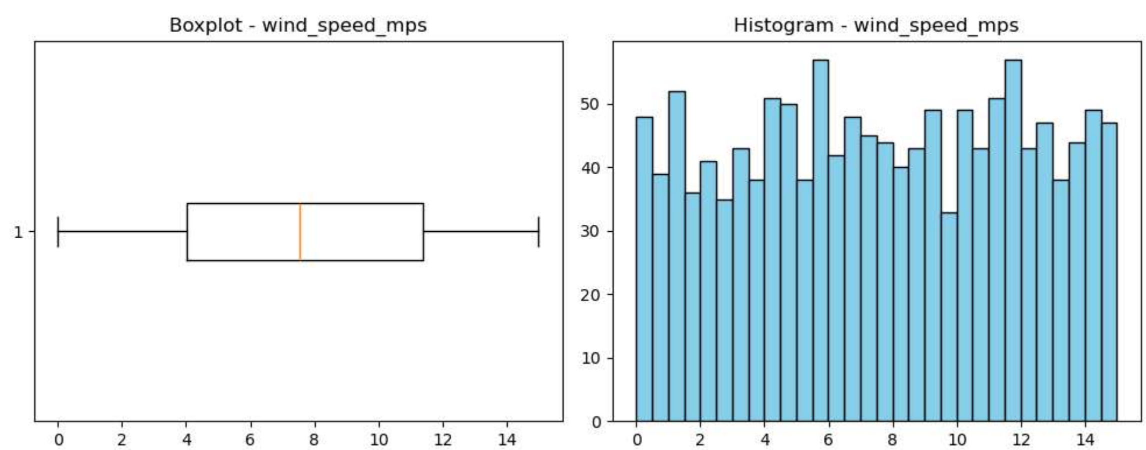
Temperature: Average, deviation

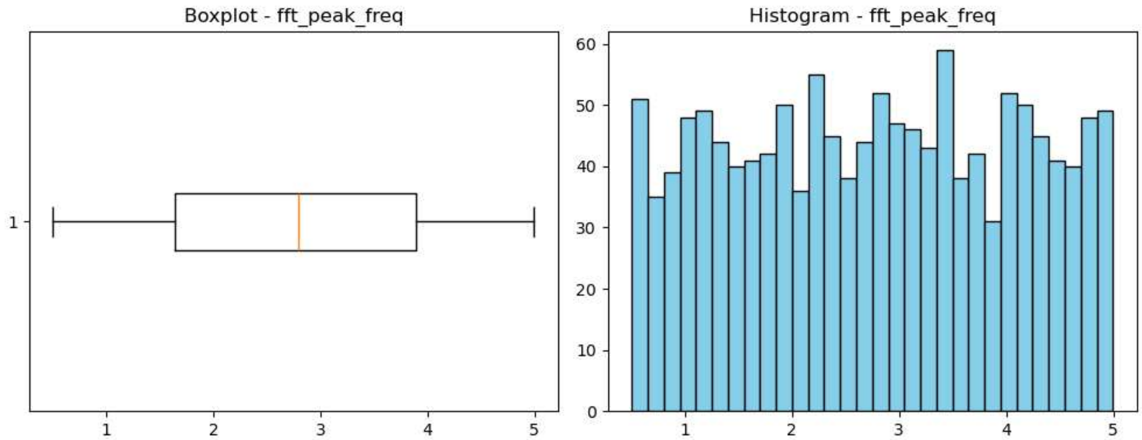
Wind: Max gusts, average

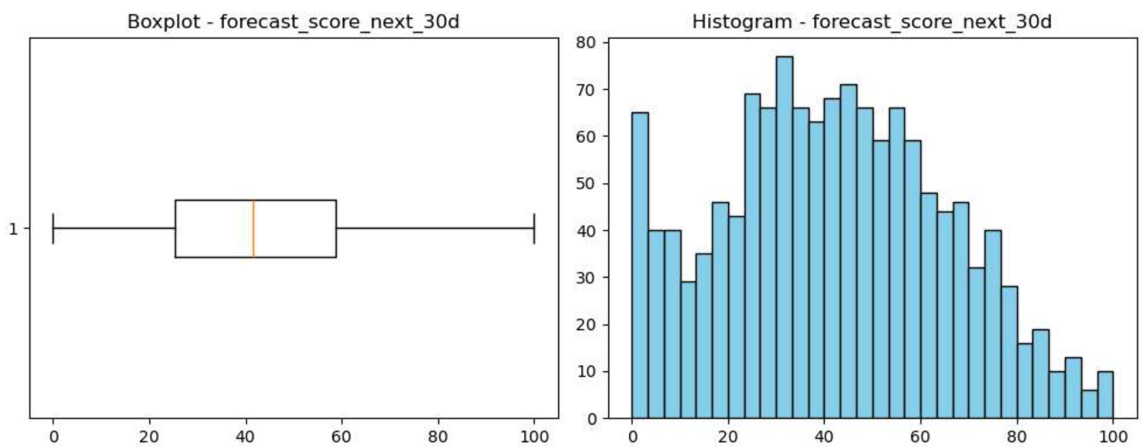
Handling Imbalance:

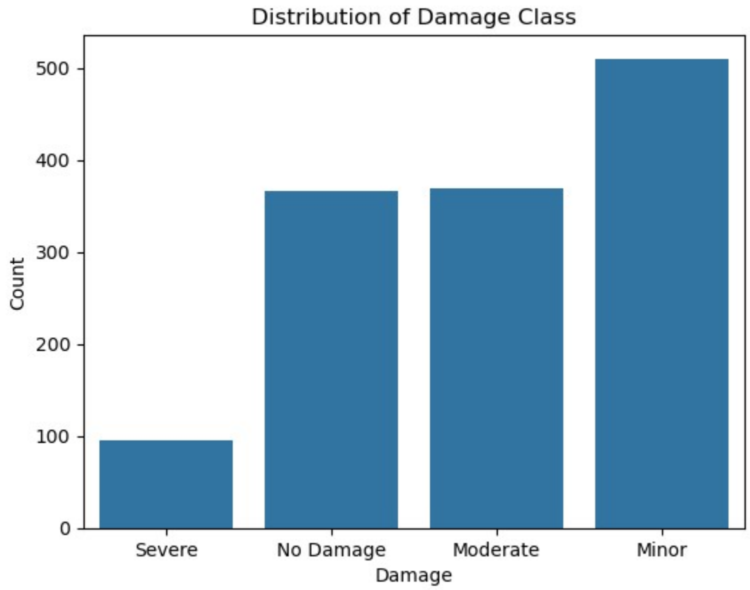
The dataset was imbalanced, so SMOTE was applied to the training data to balance the classes.

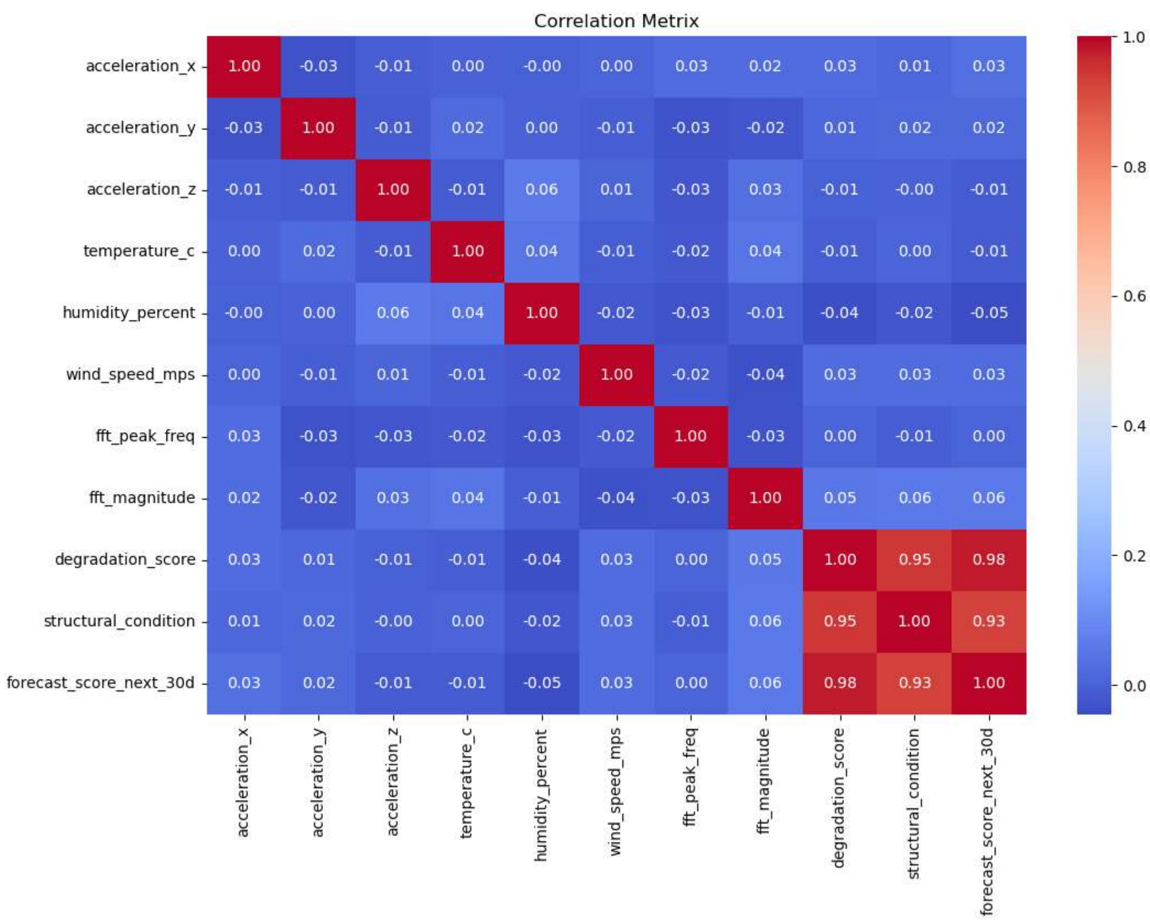
Log transformation was applied to achieve a more symmetric distribution of features.




Boxplot - fft_peak_freq
Histogram - fft_peak_freq
1
1 2 3 4 5
0 10 20 30 40 50 60


Boxplot - forecast_score_next_30d
Histogram - forecast_score_next_30d
1
0 20 40 60 80 100
0 10 20 30 40 50 60 70 80





3. Modeling Approach

Initial Machine Learning Models

Machine learning models including **Logistic Regression, SVM, XGBoost, and Random** Forest were first applied. However, the performance was poor, with accuracy ranging between **40–68%,** as the **features did not exhibit strong linear relationships** and the dataset included multiple data modalities (vibration, temperature, windspeed).

Hybrid Deep Learning Model

To address this, a hybrid deep learning model was proposed, consisting of:

1D CNN for capturing temporal patterns

LSTM for sequence modeling

Attention mechanism for focusing on relevant features

Classifier for final anomaly detection

```
HybridModel(
  (cnn): Sequential(
    (0): Conv1d(19, 32, kernel_size=(3,), stride=(1,), padding=(1,))
    (1): ReLU()
    (2): Conv1d(32, 32, kernel_size=(3,), stride=(1,), padding=(1,))
    (3): ReLU()
    (4): AdaptiveAvgPool1d(output_size=32)
  )
  (lstm): LSTM(32, 32, num_layers=2, batch_first=True, bidirectional=True)
  (attention): Attention(
    (attn): Linear(in_features=64, out_features=1, bias=True)
  )
  (fc): Linear(in_features=64, out_features=4, bias=True)
)
```

Training Details:

Framework: PyTorch

Data preparation ensured correct shapes and scaling. **2D inputs were converted to 3D** for sequence modeling.

All three dataset versions were trained; Version 2 (after feature engineering) provided the best results with **training and validation accuracy around 95%.**

Hyperparameter Optimization:

Optuna was used with **3-fold cross-validation** to find the best hyperparameters.

After training with optimized parameters and applying early stopping, training and validation accuracy reached **98.25% without overfitting.**

model = HybridModel(

input_channels=19,

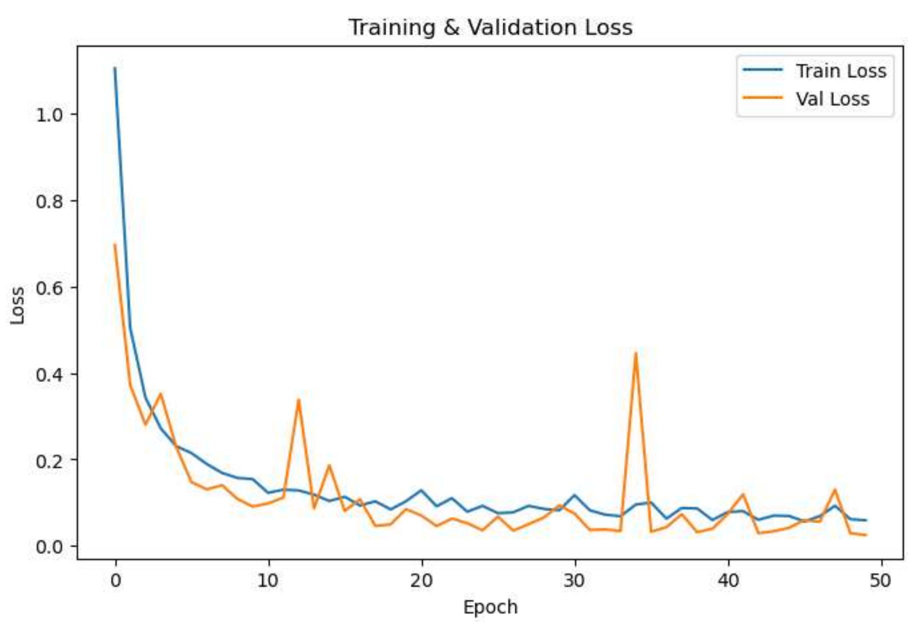
cnn_channels=32,

lstm_hidden=32,

lstm_layers=2,

num_classes=4)

Training Vs Validation loss



4. Testing and Evaluation

The best saved model was tested on unseen data, achieving an accuracy of **96.62%.**

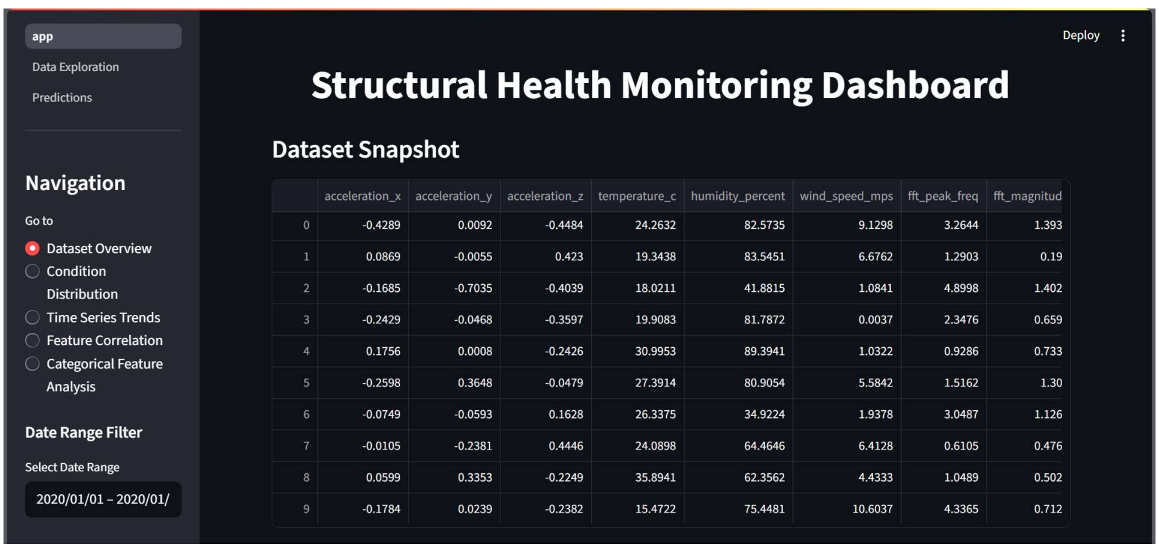
Performance metrics such as Precision, Recall, and F1-Score were calculated and visualized using a classification report.

5. User Interface (UI)

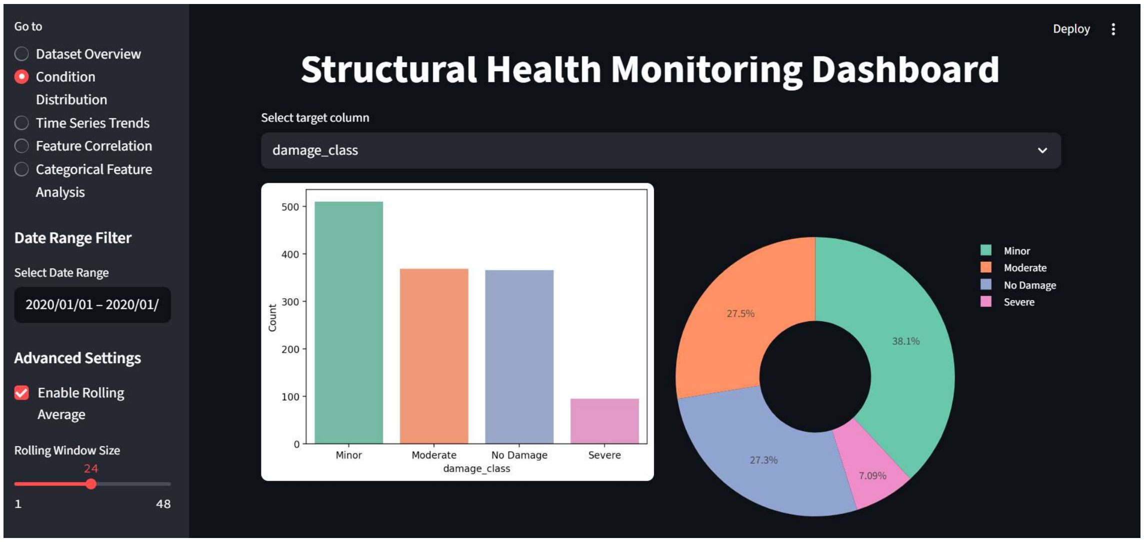
A multi-page dashboard was created using Streamlit and deployed **on Streamlit** Community Cloud.

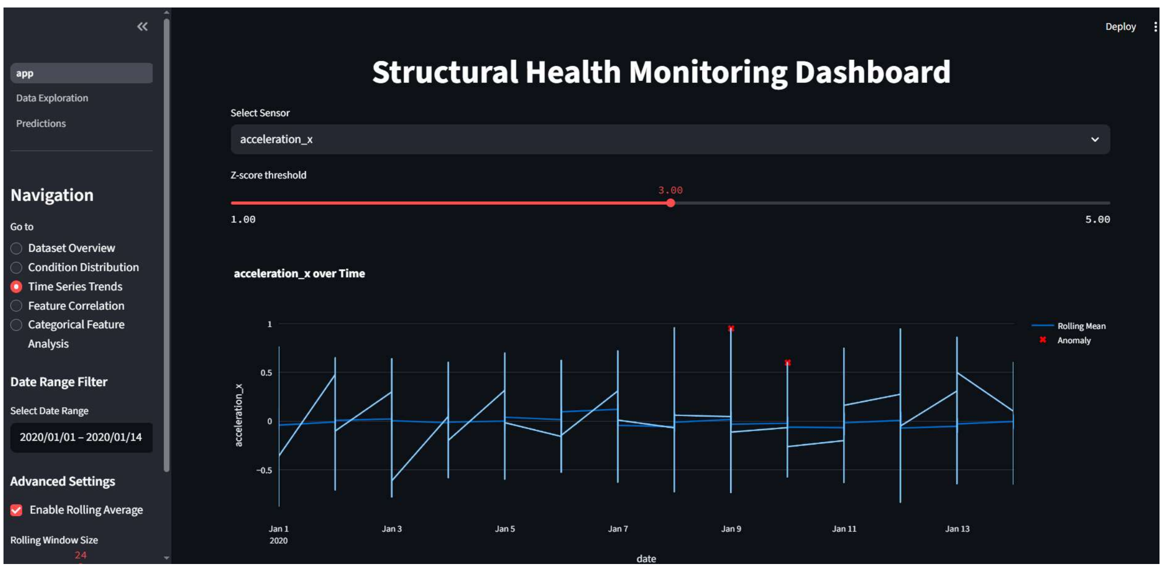
Page 1: Data exploration and visualization

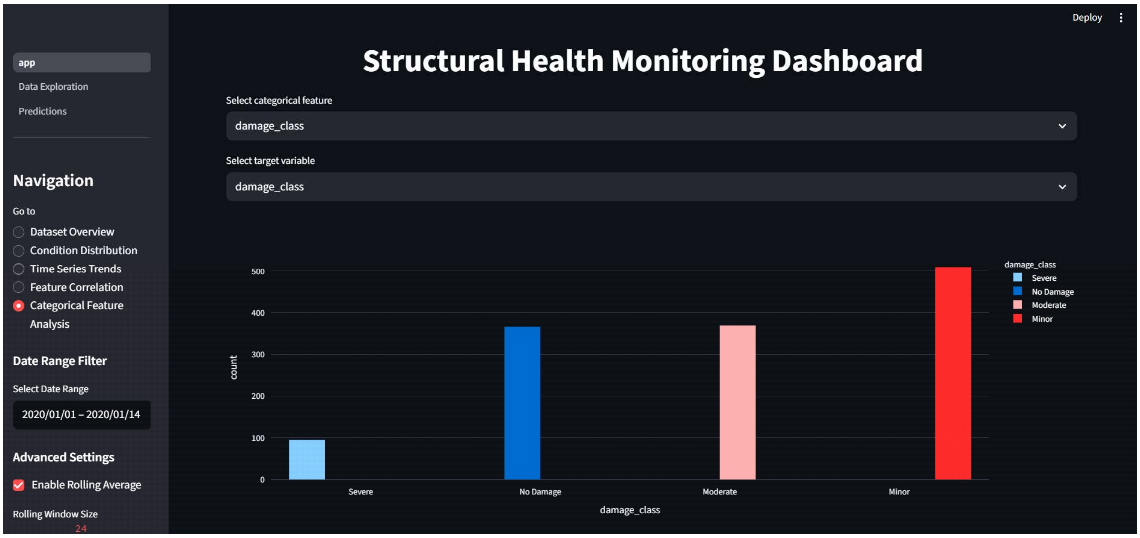
Page 2: Real-time predictions from the hybrid deep learning model. Predictions were saved in JSON formate.

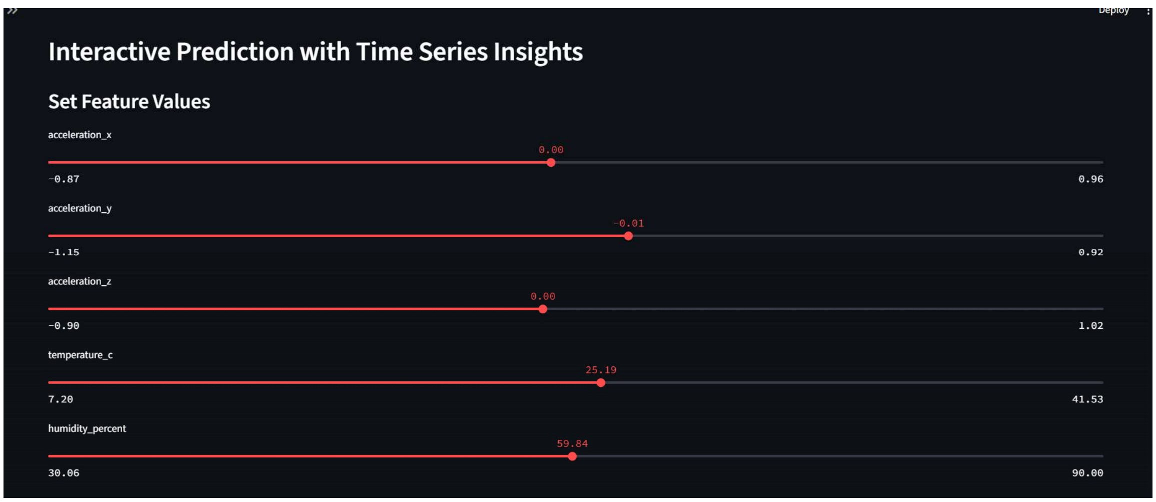


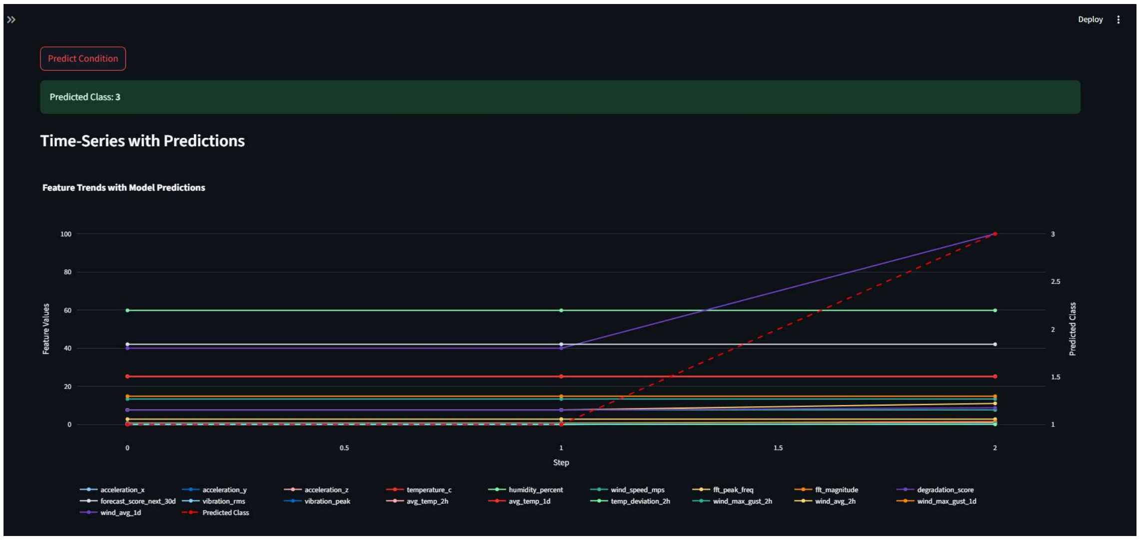
	acceleration_x	acceleration_y	acceleration_z	temperature_c	humidity_percent	wind_speed_mps	fft_peak_freq	fft_magnitud
0	-0.4289	0.0092	-0.4484	24.2632	82.5735	9.1298	3.2644	1.393
1	0.0869	-0.0055	0.423	19.3438	83.5451	6.6762	1.2903	0.19
2	-0.1685	-0.7035	-0.4039	18.0211	41.8815	1.0841	4.8998	1.402
3	-0.2429	-0.0468	-0.3597	19.9083	81.7872	0.0037	2.3476	0.659
4	0.1756	0.0008	-0.2426	30.9953	89.3941	1.0322	0.9286	0.733
5	-0.2598	0.3648	-0.0479	27.3914	80.9054	5.5842	1.5162	1.30
6	-0.0749	-0.0593	0.1628	26.3375	34.9224	1.9378	3.0487	1.126
7	-0.0105	-0.2381	0.4446	24.0898	64.4646	6.4128	0.6105	0.476
8	0.0599	0.3353	-0.2249	35.8941	62.3562	4.4333	1.0489	0.502
9	-0.1784	0.0239	-0.2382	15.4722	75.4481	10.6037	4.3365	0.712


Go to
Dataset Overview
Condition Distribution
Time Series Trends
Feature Correlation
Categorical Feature Analysis
Date Range Filter
Select Date Range
2020/01/01 – 2020/01/
Advanced Settings
Enable Rolling Average
Rolling Window Size
24
1
48
Deploy
Structural Health Monitoring Dashboard
Select target column
damage_class
Count
500
400
300
200
100
0
Minor
Moderate
No Damage
Severe
damage_class
27.5%
38.1%
27.3%
7.09%
Minor
Moderate
No Damage
Severe


app
Data Exploration
Predictions
Navigation
Go to
Dataset Overview
Condition Distribution
Time Series Trends
Feature Correlation
Categorical Feature Analysis
Date Range Filter
Select Date Range
2020/01/01 – 2020/01/14
Advanced Settings
Enable Rolling Average
Rolling Window Size
24
Deploy
Structural Health Monitoring Dashboard
Select Sensor
acceleration_x
Z-score threshold
3.00
1.00
5.00
acceleration_x over Time
Rolling Mean
Anomaly
acceleration_x
1
0.5
0
−0.5
Jan 1
2020
Jan 3
Jan 5
Jan 7
Jan 9
Jan 11
Jan 13
date


Deploy
app
Data Exploration
Predictions
Navigation
Go to
Dataset Overview
Condition Distribution
Time Series Trends
Feature Correlation
Categorical Feature Analysis
Date Range Filter
Select Date Range
2020/01/01 – 2020/01/14
Advanced Settings
Enable Rolling Average
Rolling Window Size
24
Structural Health Monitoring Dashboard
Select categorical feature
damage_class
Select target variable
damage_class
damage_class
Severe
No Damage
Moderate
Minor
count
500
400
300
200
100
0
Severe
No Damage
Moderate
Minor
damage_class

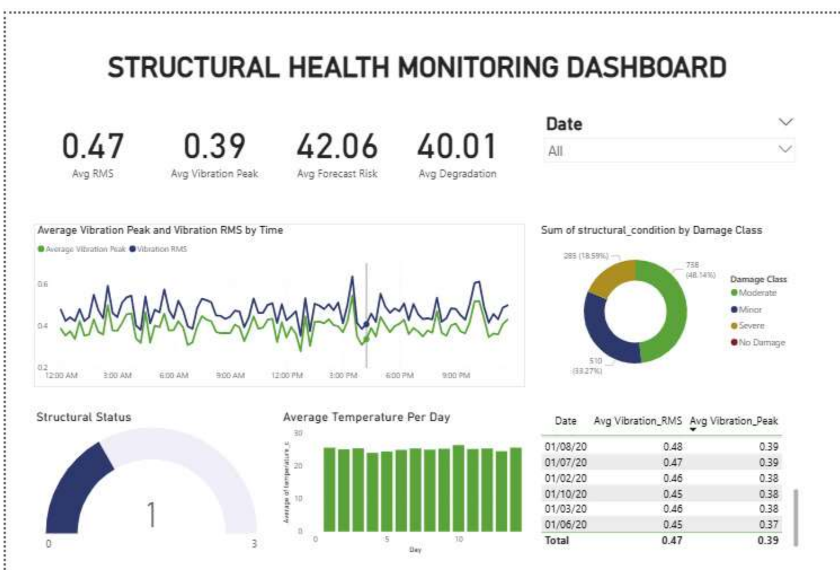

Interactive Prediction with Time Series Insights
Set Feature Values
acceleration_x
0.00
-0.87
0.96
acceleration_y
-0.01
-1.15
0.92
acceleration_z
0.00
-0.90
1.02
temperature_c
25.19
7.20
41.53
humidity_percent
59.84
30.06
90.00



6. Feature Dashboard

A **Power BI dashboard** was developed to visualize all sensor data effectively.

Multiple graphs and charts were used to represent vibration, temperature, and wind data trends, making it easier for stakeholders to analyze structural behavior.



7. Technologies Used

Programming & Frameworks: Python, PyTorch, Streamlit

Data Manipulation & Visualization: Pandas, NumPy, Matplotlib, Seaborn, Plotly

Data Imbalance Handling**: SMOTE**

Hyperparameter Optimization: **Optuna**

Dashboard & Reporting: **Power BI**

Deployment: **Streamlit Community Cloud**

Conclusion

This project successfully developed a multi-modal Structural Health Monitoring system capable of detecting anomalies in bridges and other infrastructure. By combining multiple sensor data types and using a hybrid deep learning model, the system achieved high accuracy (96–98%) in anomaly detection. The interactive Streamlit UI and Power BI dashboard provide an accessible way for stakeholders to monitor structural health in real-time, demonstrating a practical and scalable SHM solution.