Computational Intelligence in Engineering

Group 14 - Project B : To determine if the Hohenzollernbrücke structure has some imperfections and locate their exact position

IAM Institute of General Mechanics | RWTH AACHEN UNIVERSITY

Team Introduction

- **Marc Weismüller**
  - Data (Pre-)processing and Correlation Matrix
  - U-NET Model for finding the location of the missing nodes
  - Presentation
- **Praveen Raaj Rajamurugesan**
  - LSTM - Binary Classifier
  - Testing and Validating the Models
- **Ramesh Akilandeswarri Angala Varadarajan**
  - Study of different ML models, Confusion Matrix
  - Presentation and Presentation preparation, Report preparation
- **Gokul Visakakannan**
  - Hyper Parameter Tuning & Cross Validation
  - Presentation preparation, Report preparation
- **Adarsh Adarsh**
  - Study of different network architectures on learning curves
  - Presentation preparation, Report preparation

Group 14 : Overview

CONTENT

- **INTRODUCTION**
  - Motivation
- **METHODS**
  - Data (Pre-)processing
  - Artificial Intelligence
- **LSTM - Binary Classifier**
  - Network Architecture - Sequential Neural Network
- **RESULTS & DISCUSSION**
  - Hyper Parameter Tuning & Cross Validation
  - Generalization Ability (Prediction for Test dataset)
  - U-NET (Approach to find the imperfect nodes)
  - Images
- **CONCLUSION & OUTLOOK**
  - Conclusion
  - Outlook

INTRODUCTION

Motivation

- Key applications and motivations for this include **early detection of structural imperfections**, achieved by analysing simulation data to develop AI/ML models
- Identifying and precisely localizing imperfections is crucial, allowing us to **plan targeted repairs**
- Load impact analysis, including gravitational forces, train loads and crosswinds, helps in **assessing the bridge's performance under diverse conditions**
- Continuous monitoring of the bridge health using sensors and data analysis can provide **real-time insights into its condition**
- Machine learning algorithms can be integrated into a monitoring system to **detect unusual changes in the structural behaviour**
- The project also emphasizes **predictive maintenance**, utilizing data-driven predictions to **optimize the bridge's lifespan and minimize downtime**

METHODS

Data (Pre-)processing

- Redundant features which have a **pearson correlation** that is higher than **0.7** are not considered.
- $D_T = \sqrt{(D_X^2 + D_Y^2 + D_Z^2)}$

Total_Def,
Def x, Def y, Defz,
shXY, shXZ, shYZ

↓

Total_def,
shXY, shXZ, shYZ

Timestamp :
0, 0.1, 0.5, 0.9, 1.1, 1.5,
1.9, 2.1, 2.5, 2.9, 3

↓

Timestamp :
0.1, 1.1, 2.1, 3

Final set of Feature vectors

Artificial Intelligence

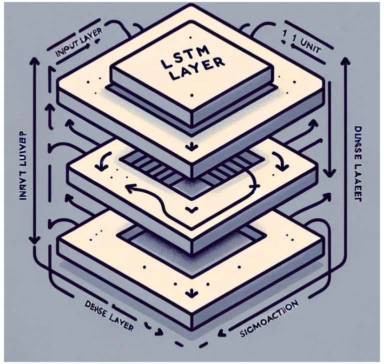
- Tried **LSTM** for both binary and multiclass classification and **U-NET** approach to transform the data into image which can be further used for locating the imperfection
- LSTM for binary classifier had good accuracy with training data.

LSTM - Binary Classifier

Network Architecture - Sequential Neural Network

- **Input Layer :**
  Time steps and features
- **First Layer :**
  - 50 units (neurons)
  - Returns full sequence to next layer
- **Second Layer :**
  - 50 units (neurons)
  - Returns only output of last time step
- **Dense layer :**
  Output of network is single layer, for binary classification sigmoid activation is used
- **Optimizer :** Adam,used for neural networks
- **Loss function :**
  Binary cross entropy which is generally used for binary classification
- **Metrics :**
  Accuracy and F1 score

```
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(X_reshaped.shape[1], X_reshaped.shape[2])))
model.add(LSTM(units=50))
model.add(Dense(units=1, activation='sigmoid'))
custom_optimizer = Adam(learning_rate=0.005)
model.compile(optimizer=custom_optimizer, loss='binary_crossentropy', metrics=['accuracy'])
```



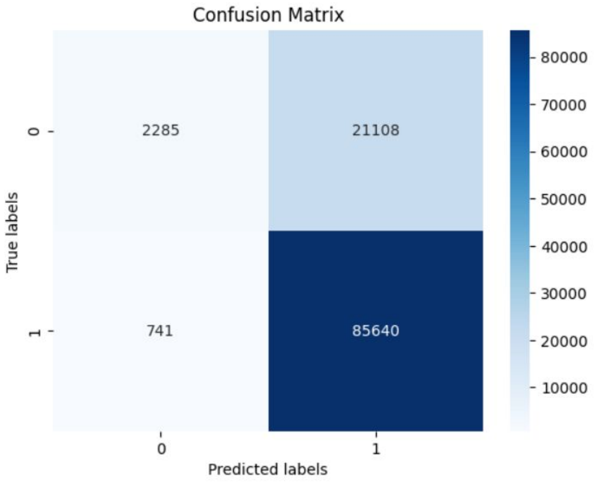
LSTM with two layers and one dense layer [1]

RESULTS & DISCUSSION

Hyper Parameter Tuning & Cross Validation

- Early stopping is used for preventing overfitting
  -'val_loss' is used as monitor
- Learning rate was tuned to optimal value based on accuracy over **10 epochs**
- **Weighted F1 Score :** 0.886447343779262
- **Accuracy =** 0.8099

```
Learning Rate: 0.01 - Accuracy: 0.7925009727478027
Learning Rate: 0.0001 - Accuracy: 0.7868985533714294
Learning Rate: 0.0005 - Accuracy: 0.7892670631408691
Learning Rate: 0.001 - Accuracy: 0.794386625289917
Learning Rate: 0.005 - Accuracy: 0.7949241399765015
Learning Rate: 0.01 - Accuracy: 0.7925009727478027
```

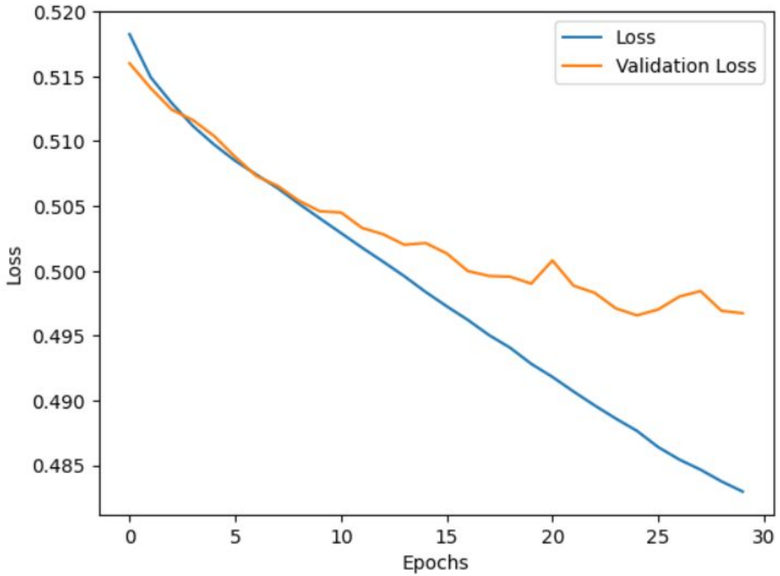


RESULTS & DISCUSSION

Hyper Parameter Tuning & Cross Validation

- With help of early call back on validation loss epoch converges around **30**
- We use **StratifiedKFold** cross validation function to avoid overfitting. We get F1 score average of **0.8828** and accuracy of **0.8012**

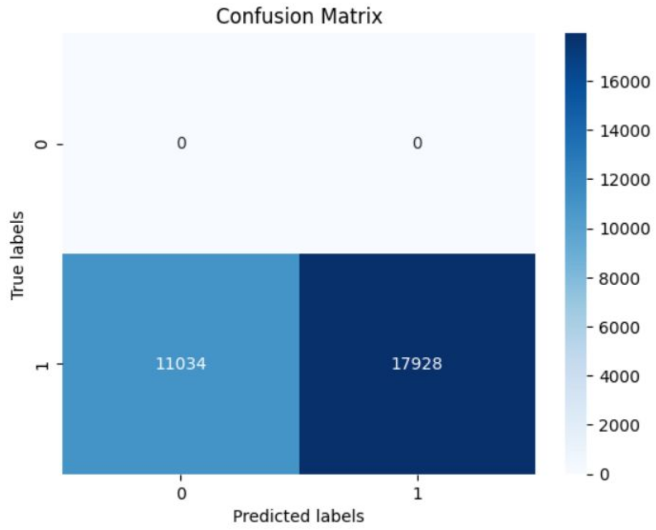
```
# F1 Score Callback
f1_callback = F1ScoreCallback(X_test, y_test)
# Perform cross-validation with F1 score callback
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```



RESULTS & DISCUSSION

Generalization Ability (Prediction for Test dataset)

- Test Accuracy is poor compared to Train Accuracy
  - **Accuracy =** 0.619
  - **F1 Score :** 0.764



[2, 3,]

U-NET (Approach to find the imperfect nodes)

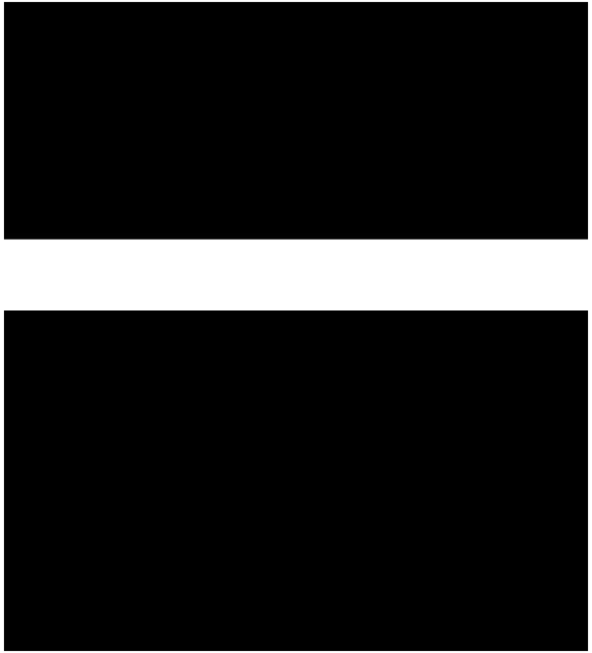
- Masks (labels) to train the model for the segmentation task can be created by black and white images where missing nodes of the imperfect structure are white pixels.
- More data is required or a custom U-Net with larger convolution matrix and custom architecture.

RESULTS & DISCUSSION

Images

Image of the Transformed Sensor Data

Image of the Mask (Label)



Prediction of the model on the image
White pixels are imperfect nodes

[4, 5, 6]

CONCLUSION & OUTLOOK

CONCLUSION:

- Binary classifier was done using LSTM Model and UNET for predicting the state of node.
  - LSTM Binary classifier :
    - **Training accuracy**: 0.8099 **Test accuracy**:0.619
    - **F-Score accuracy**: 0.8864 **F-Score accuracy:**0.764
    - The model converges around epoch of 30
    - The generalization doesn't seem to have implemented maybe due to imbalance of labeled data.And UNET requires more data to improve its performance

OUTLOOK:

- To implement and improve these algorithms to minimise imperfections, monitor and enhance structural integrity for similar structures such as Hohenzollernbrücke structure
- Once imperfections are detected, AI algorithms can suggest repair and predictive maintenance strategies as well as their associated cost and timeline

REFERENCES

[1] https://journals.sagepub.com/doi/full/10.1177/14759217211036880

[2] Original Unet paper with architecture of model : https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[3] The used Unet implementation in Pytorch: https://github.com/milesial/Pytorch-UNet

[4] Unet for time series sensor data : https://ieeexplore.ieee.org/document/10210541

[5] https://www.dfki.de/fileadmin/user_upload/import/9072_sensor2image_deepnet.pdf

[6] https://www.sciencedirect.com/science/article/pii/S0888327020307846

[7] https://shop.mh-p.de/autothumb/720x0/Koeln.jpg

[7]

Thank you for your attention!