THE PRESENT WORK WAS SUBMITTED TO THE INSTITUTE OF GENERAL MECHANICS

RHEINISCH WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN
FACULTY OF MECHANICAL ENGINEERING

PROJECT REPORT

TITLE: INTELLIGENT MONITORING OF THE PRODUCTION SYSTEM

INTELLIGENT MONITORING OF ENGINEERING SYSTEMS

Presented by: Deven, Paul (453329), MME-CAME

Kshitij, Shahare (452785), MME-PS

Rohan, Ravi Mani (452770), MME-PS

Vikhyat, Jammalamadaka (451897), MME-CAME

Group: 3

Declaration

We declare to the best of our knowledge that we have written this report by ourselves and only with the help of all references listed in the bibliography.

Aachen, July 23, 2024

Table of Contents

List of Figures

List of Tables

1 Introduction

Intelligent monitoring plays a critical role in engineering systems for various reasons. It improves the dependability and effectiveness of operations through the provision of real-time data and insights into system performance. This early detection of anomalies and potential failures helps in reducing downtime and maintenance costs. By addressing issues before they escalate, intelligent monitoring aids in preventing expensive repairs and extending the lifespan of equipment.

Additionally, intelligent monitoring facilitates decision-making processes. Engineers and managers can make well-informed decisions based on precise data, thereby enhancing overall system management. This technology also assists in optimizing performance by continuously analyzing and adjusting system parameters to achieve optimal results.

Moreover, intelligent monitoring contributes to safety by ensuring that systems operate within safe limits and alerting operators to any hazardous conditions. This is particularly crucial in industries where system failures can lead to severe consequences, such as aviation, nuclear power, and manufacturing.

Figure 1: BMW car assembly line

Furthermore, intelligent monitoring is vital for complying with industry standards and regulations. It offers the necessary documentation and proof that systems are functioning correctly and safely, which is essential for audits and inspections, helping companies avoid penalties and legal complications.

The significance of intelligent monitoring in engineering systems cannot be emphasized enough. It enhances efficiency, safety, and reliability, while also supporting regulatory compliance and informed decision-making. By harnessing advanced technologies, organizations can guarantee that their engineering systems operate at peak performance and safety standards.

The project performed aims to identify flaws in the Assembly line in real time to avoid any interruptions in the workflow. The objective was to collect acceleration to time data by utilizing the accelerometer of a mobile phone while it traverses the Assembly line, and then utilize the gathered data to determine the location of defects through the training of a machine learning model for this purpose.

The Division of responsibilities is as follows:

- Data Collection: Kshitij, Rohan
- Data Preprocessing: Rohan, Vikhyat
- Feature Selection: Deven, Kshitij
- Computational Model: Deven, Vikhyat
- Presentation and Report: Deven, Kshitij, Rohan, Vikhyat

2 Methods

2.1 Signal (pre-)processing

The experiment campaign consisted of 9 groups where each group ran 5 test cases each. Each group performed one test case where there were no defects in the system. The other four test cases were run with defects in the system in the form of frequencies applied, defects or inclinations.

We first performed a Fast Fourier transform on the data obtained to verify if the frequency data matches with the data given. After noticing an issue with our group's data, we performed the experiment again.

After verification of the collected data, we created a set of timings which links to the position of the phone when it travels through the line. It was decided to label the data with these numbers which helps us locate the areas where the phone detects any defects in the Assembly line. It was named as follows

Using this method, we trimmed the data by removing locations 8 and 2 for analyzing the damper and incliners. We considered the Time data and linear accelerations X, Y and Z for the computational model and removed Absolute acceleration as it didn't show as much variance for the model after performing a PCA Analysis. A metadata is created in Python which is a dictionary designed to identify damping and inclination locations based on group and case numbers. Its purpose is to classify, for specific time stamps or sections, whether a damping or inclination defect is present, indicated in the form of 0s (no defect) or 1s (defect).

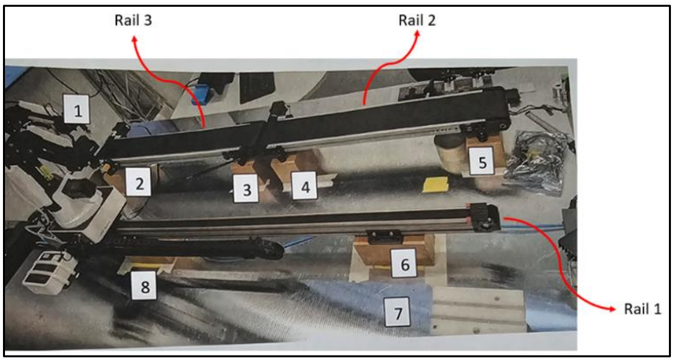

Figure 2: Mock Assembly Line

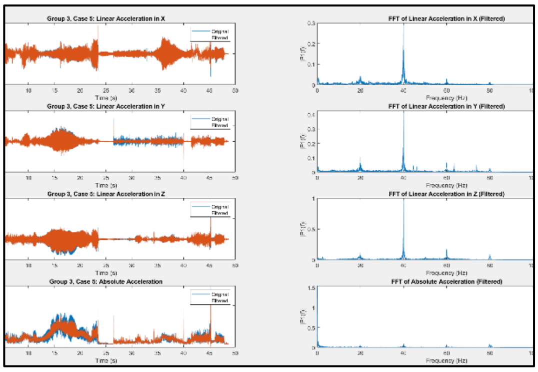

Figure 3: FFT performed on MATLAB

Time region (s)	Location
0-10	8
10-18	7
18-22	6
23-27	5
27-31	4
31-36	3
36-40	2
40-55	1

Table 1: Time region

2.2 Computational Model

For our project, we initiated signal testing using the Fast Fourier Transform (FFT) algorithm, which was necessary given the nature of our time-series data. After evaluating various models suitable for time-series data analysis, including

- Fully connected feed-forward
- Convolutional Neural Network (CNN) [10]
- Kolmogorov-Arnold Network (KAN) [1]

Upon testing, our primary CNN model demonstrated the highest accuracy compared to the other models [Refer Figure 5: Flowchart of CNN model].

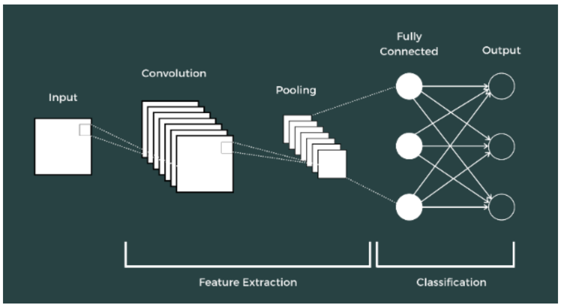

Figure 4: Convolutional Neural Network

2.3 Best Performing Model

The input data shape for our best-performing model was trimmed post every iteration for sound predictions. Following changes were made to the input data

- Removal of Section 8 and 1 as well as Case 2 and 3 from each group (to overcome class imbalances)
- Implementation of frequency in input data for better pattern identification and later removal to avoid Curse of Dimensionality. [12]

Upon testing, our primary CNN model demonstrated issues with determining the frequency and its location. Therefore, after a few iterations we decided to remove frequency and its location from output and use the CNN to detect anomalies for damper and incliner and their respective locations only.

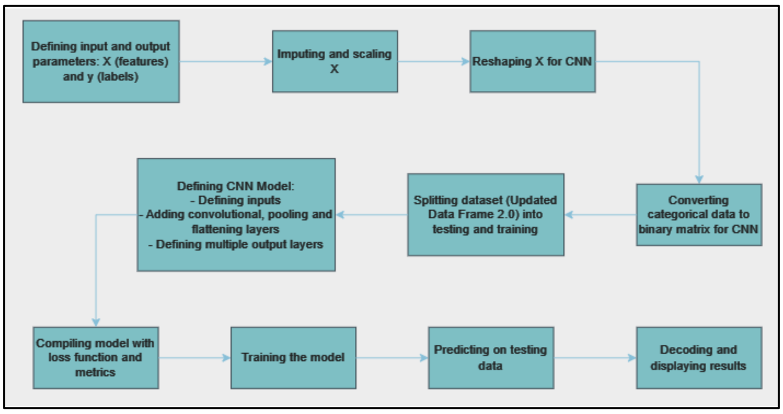

Figure 5: Flowchart of CNN Model used

2.3.1 Model architecture

For our machine learning model, the training data consists of several key inputs: time stamps, accelerations, damping, inclination locations, and section-wise categorization. Similarly, the testing data includes the same inputs as the training data, with the addition of frequency values identified through Fast Fourier Transform (FFT) in Python [Refer Figure 7: FFT on Test data]. This comprehensive data collection ensures that our model can learn from diverse features and accurately predict the target outputs.

The structure of our model is centered around a Fully Connected Conv1D layer, which serves as the foundational layer. This one-dimensional convolutional layer is designed to be fully connected to subsequent layers, allowing it to effectively capture temporal dependencies within the input data. This is crucial for understanding the sequential nature of the data and extracting meaningful patterns over time. Following the Conv1D layer, we incorporate two hidden layers. These layers are instrumental in transforming the input data into higher-level representations, which help the model learn and recognize complex patterns. [Refer Figure 4] To introduce non-linearity into the model, we use the Rectified Linear Unit (ReLU) activation function. This activation function enhances the model's ability to learn intricate patterns by enabling it to handle complex data distributions more effectively.

The outputs of the model are designed to provide key information about the system being analysed. Specifically, the model predicts two primary outputs: the damping location and the inclination location. These outputs are crucial for understanding the behaviour and dynamics of the assembly line, providing valuable insights for optimization and control.

To ensure the model performs optimally, various hyper-parameters were carefully adjusted during the training process. This includes fine-tuning the learning rate and selecting an appropriate loss function. [Refer Figure 6 for: Loss over epoch, for learning rate optimization.]

2.3.2 Training parameters

For our model training, we utilized the Adam optimizer [11] with a learning rate set to 0.0005. The loss functions employed were 'binary_crossentropy' for each of the two output categories: 'damping_loc_output', and 'inclination_loc_output'. Correspondingly, the primary metric used to evaluate the model's performance for each output was 'accuracy'. The model was trained over 15 epochs with a batch size of 32.

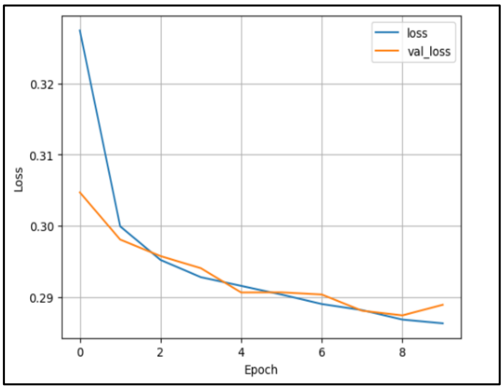

Figure 6: Loss v Epoch (lr = 0.0005)

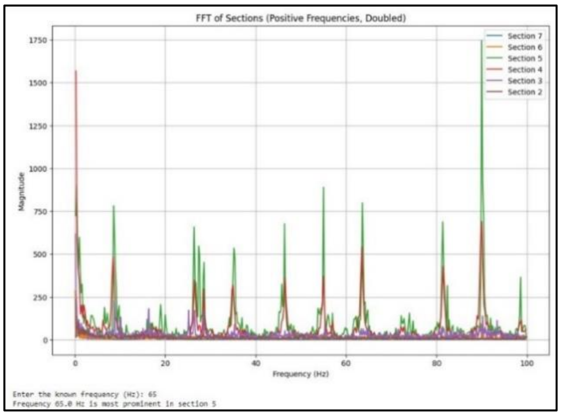

Figure 7: FFT on Test-Data

3 Results

3.1 Cross validation

- The model was trained on randomized train-test split of [70-30] out of 50 total cases and predictions for damping and inclination locations were cross-validated against true-labels for the validation dataset. Accuracy, F1 score, and Confusion matrix were derived from the validation predictions and used as metrics for cross-validation [Refer Figure 8 for Confusion Matrix of validation dataset and Figure 10 for Test data set].
- Results of validation dataset (30%):
  - Damping Location Accuracy = 0.91
  - Damping F1 Score = 0.88
  - Inclination Location Accuracy = 0.95
  - Inclination F1 Score = 0.93

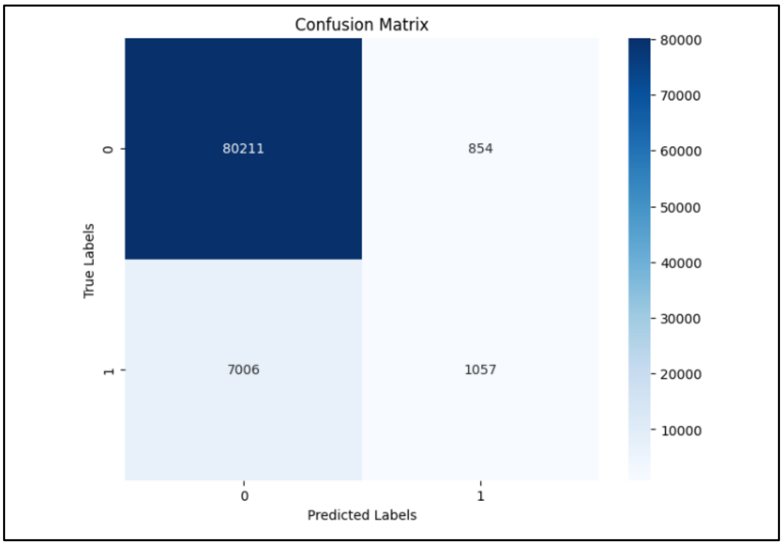

Figure 8: Confusion matrix for validation dataset

3.2 Discussion

[Refer Table 2 for predicted Results, the inconsistencies by model are marked in red while true value are marked green].

- The model has achieved higher accuracy for both damping and inclination locations, however there are significant class imbalances.
- The recall values for damping and inclination (0.08 and 0.13) are low, but the precision is round about 0.6 to 0.67, which is good.
- The model for predicting the frequency and its location was priorly done using FFT.
- The Generalization ability was passed overall. However, there were some issues regarding generalization, discussed in more detail in the subsequent section.

3.3 Generalization ability (test dataset)

Yes, the generalization ability worked but human intervention was required to interpret the results from the model. Further, the model predicted additional defects in the test data, which were not present in the showcased results, probably because of overfitting. Moreover, as there was a huge class imbalance for 0s and 1s, there was a requirement for diverse training data. The frequency obtained in test case 4 was merely 20Hz, which we earlier thought of as noise and removed that.

[Refer Figure 9 for metrics for damping and inclination locations for model performance].

Test Case	Frequency (Hz)	Frequency Location	Damping Location	Inclination Location
1	65	5	-	-
2	0	-	-	4
3	65	5	5	6*	-
4	20	5	7*	-	-
5	65	5	4	3

**The indicated red ones are the predicted results of our group. The true results are marked in green.*

Table 2: Predicted and True Results

```
Damping Location Metrics:
Accuracy: 0.5
Precision: 0.5
Recall: 0.5
F1-Score: 0.5

Inclination Location Metrics:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-Score: 1.0

Frequency Location Metrics:
Accuracy: 0.75
Precision: 0.5625
Recall: 0.75
F1-Score: 0.6428571428571428
```

Figure 9: Metrics for the model performance

```
Damping Location Confusion Matrix:
[[80723   342]
 [ 7384   679]]
Inclination Location Confusion Matrix:
[[84127   369]
 [ 4023   609]]
```

Figure 10: Confusion matrix for test dataset

4 Conclusion and Outlook

4.1 Conclusion

A novel idea for predictive maintenance of Production systems using AI is proposed. The project flew from data collection using Phyphox to Signal Filtering (using 4th order Butterworth filter), Feature extraction (PCA analysis) and finally Computational Modelling & Validation (CNN and KAN). While still working on the nuances of KAN, a Fully Connected Conv1D was trained to compare the accuracies of conventional MLPs (Multi-Layer Perceptron) and KANs [1]. However, few parameters in KAN remain to be tested for comparison. And hence, CNN was used to find the defects in the system because of its performance for classification in time-series data.

The Fully Connected Conv1D took Time, Acc (in X, Y, Z) and Section as input, and achieved an accuracy of 91.19% and 94.95% for damping and inclination location respectively. However, the model had issues in generalizing because catastrophic forgetting is a serious problem in current Machine Learning models [2]. The model's performance is relatively good, with high accuracy and recall but moderate precision. This indicates that the model is good at identifying true cases but also produces some false positives. Analyzing the false positives could help refine the model. The F1-score of 0.643 shows a balance but indicates room for improvement.

Further improvements in CNN modelling could include:

- The feature engineering for precise section transition.
- System dimension, incliner/damper dimension and boundary conditions could be implemented for imposing PDE (Partial Differential Equation) loss when solving PDEs for PINNs [5,6].

Implementation of PINNs for PDE solving could be major area of improvement for the current model. We can replace the paradigm of using MLPs for imposing PDE loss when solving PDEs. We can refer to Deep Ritz Method [4], PINNs [5,6,7] for PDE solving and Fourier Neural operator [8] for frequency determination and location detection. This implementation would require extensive testing data set for testing; however, the question would be if PINNs derived from MLPs would outperform PINNs derived from KANs.

The CNN model generalized approx. 80% of the test results correct which it would owes to the user intervention and re-training for curbing overfitting (an earlier overfit model generalized with 55% accuracy). The importance of MLPs can never be overstated, since they are the default models in machine learning for approximating nonlinear functions, due to their expressive power guaranteed by the universal approximation theorem [3]. Which makes us question: Are MLPs the best non-linear regressors?

4.2 Outlook

The achieved results can be used to define and set standards for accuracies of current Machine Learning models like MLPs in Intelligent Monitoring of Engineering Systems. The novel idea is to use the available results and literature to test and challenge the MLPs status quo. While MLPs have fixed activation functions on nodes ("neurons"), KANs have learnable activation functions on edges ("weights"). KANs have no linear weights at all – every weight parameter is replaced by a univariate function parametrized as a spline. We believe that this seemingly simple change makes KANs outperform MLPs in terms of accuracy and interpretability, on small-scale AI + Science tasks. [1]

Currently, the biggest bottleneck of KANs lies in its slow training. KANs are usually 10x slower than MLPs, given the same number of parameters. We should be honest that we did not try hard to optimize KANs' efficiency though, so we deem KANs' slow training more as an engineering problem to be improved in the future rather than a fundamental limitation. If one wants to train a model fast, one should use MLPs. In other cases, however, KANs should be comparable or better than MLPs, which makes them worth trying. [9]

Furthermore, a few adjustments to the current assembly setup could help extract diverse and quality data for intelligent monitoring. The proximity of Location 3 and 4 in the current assembly might be one of the reasons why CNN produced False Positives for certain timestamps. While the current model does not deal with multiple defects at single location, it does predict the prominent defect with high accuracy. Future models and training could include multiple defects on a single location as might be the case in real-life problems.

5 References

[1] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljaˇci´c, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov Arnold Network

[2] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[3] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

[4] Bing Yu et al. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 6(1):1–12, 2018.

[5] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics, 378:686–707, 2019.

[6] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. Nature Reviews Physics, 3(6):422–440, 2021.

[7] Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. Advances in Neural Information Processing Systems, 36, 2024.

[8] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895, 2020.

[9] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljaˇci´c, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov Arnold Network: Final Takeaway: Should I use KAN or MLP?

[10] Z. Cui, W. Chen, Y. Chen, Multi-Scale Convolutional Neural Networks for Time Series Classification (2016), https://arxiv.org/abs/1603.06995.

[11] Diederik P. Kingma, Jimmy Ba: Adam: A method for Stochastic Optimization; https://arxiv.org/abs/1412.6980

[12] Ming-Jun Lai and Zhaiming Shen. The Kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions. arXiv preprint arXiv:2112.09963, 2021.