LAYER-WISE RELEVANCE PROPAGATION (LRP)

1. INTRODUCTION

Our project aims to improve railway safety by using machine learning to predict possible rail failures. Since rail breaks can lead to serious safety risks and financial losses, it's important that our models are not only accurate but also easy to understand. Explainable AI (XAI) helps us see why the model makes certain predictions, making it more transparent and trustworthy.

In this sprint, our team explored various XAI methods. My individual task focuses on researching Layer-wise Relevance Propagation (LRP) — a deep learning explainability method. The purpose of this report is to explain what LRP is, describe how it works, and evaluate whether it can be effectively applied to the models used in our project, including LightGBM, SVM, DNN, and Transformer architectures.

2.THEORETICAL BACKGROUND OF LRP

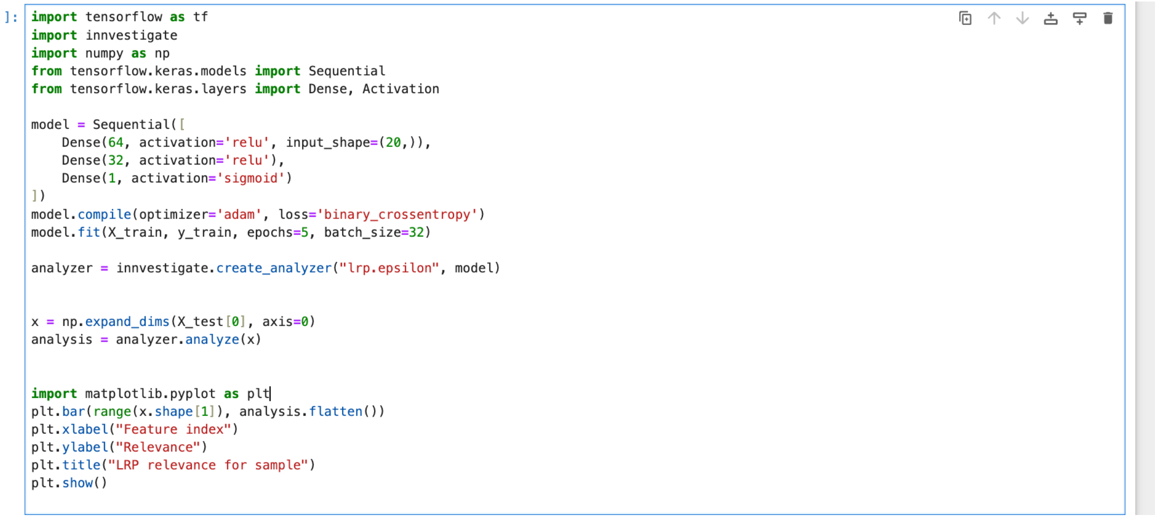
Layer-wise Relevance Propagation (LRP) is an explainability technique first introduced by Bach et al. (2015). It was developed to interpret predictions made by deep neural networks. The method decomposes a model's output backward through the layers to determine how much each input feature contributes to the final decision. This approach is based on the principle of relevance conservation, meaning that the total prediction score is redistributed layer by layer down to the input features.

In simple terms, LRP answers the question: 'Which input features were most responsible for the model's output?' For example, in a rail break prediction model, LRP can help identify whether high axle load, temperature variation, or rail age contributed most to a high predicted risk score. Unlike gradient-based methods that only provide directional sensitivity, LRP provides both the sign and magnitude of each feature's relevance, giving a more detailed understanding of model behavior.

3.IMPLEMENTATION AND TOOLS

Several libraries now support LRP for practical use. The most common include **iNNvestigate** (for TensorFlow/Keras) and **Captum** (for PyTorch).

Sample code:



```
import tensorflow as tf
import innvestigate
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation

model = Sequential([
    Dense(64, activation='relu', input_shape=(20,)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam', loss='binary_crossentropy')
model.fit(X_train, y_train, epochs=5, batch_size=32)

analyzer = innvestigate.create_analyzer("lrp.epsilon", model)


x = np.expand_dims(X_test[0], axis=0)
analysis = analyzer.analyze(x)


import matplotlib.pyplot as plt
plt.bar(range(x.shape[1]), analysis.flatten())
plt.xlabel("Feature index")
plt.ylabel("Relevance")
plt.title("LRP relevance for sample")
plt.show()
```

4. APPLICABILITY ANALYSIS TO OUR PROJECT

The following analysis evaluates LRP’s applicability to each model used in our project based on five factors: compatibility, interpretability, reliability, implementation cost, and domain usefulness.

Model	**Explanation Detail**	**Implementation Cost**	**Suitable**	**Comment**
LightGBM	Not directly supported	Low	No	Tree-based model requires SHAP instead of LRP
SVM	Can be approximated via linear kernels	High	No	LRP only works for special SVM variants
DNN	Feature-level and neuron-level relevance	Medium	Yes	Highly compatible and interpretable
Transformer	Token/temporal relevance visualization	High	Yes	Supported through Transformer-LRP

Overall, LRP is best suited for our deep learning models (DNN and Transformer), where it can reveal feature-level and temporal contributions to predicted rail break probabilities. For non-neural models like LightGBM and SVM, SHAP and other model-agnostic methods are more appropriate.

5. DISCUSSION

The most obvious advantage of LRP is that it offers clear, quantitative explanations of how different features influence a model's output. This is valuable in a safety-critical context like rail maintenance, where engineers need to understand why the system predicts a high-risk segment. LRP can, for instance, highlight that increasing axle load or frequent temperature fluctuations significantly raise the risk score.

However, there are also limitations. LRP requires additional computation during inference, especially for deep models. It is not natively compatible with non-neural algorithms, and its implementation complexity grows with network depth. Moreover, results can sometimes be sensitive to parameter choices or activation functions. Therefore, results should be interpreted together with domain knowledge.

Future work may involve applying LRP to our trained DNN and Transformer models to visualize which environmental and operational factors contribute most to predicted failures. Combining LRP with SHAP could also yield a hybrid explainability framework that covers both tree-based and deep models in our pipeline.

6. CONCLUSION

Layer-wise Relevance Propagation (LRP) is a powerful and theoretically grounded XAI technique suitable for explaining complex neural models. It is particularly effective for deep networks and Transformers, which are widely used in our Rail Break Prediction project. Through relevance maps, LRP can enhance model transparency by showing which physical and operational factors most influence the prediction. However, due to its limited compatibility with tree-based and kernel-based models, LRP should be applied mainly to the deep learning components of the system.

In summary, LRP is a feasible and valuable explainability method for DNN model.

7. REFERENCES

Arras, L. et al. (2019). Explaining Recurrent Neural Network Predictions in Sentiment Analysis.
PyTorch Captum Documentation. (2023). https://captum.ai/

Bach, S. et al. (2015). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation.

Chefer, H. et al. (2021). Transformer Interpretability Beyond Attention Visualization. CVPR.

Montavon, G. et al. (2017). Explaining nonlinear classification decisions with deep Taylor decomposition.