A Preliminary Research on Feature Selection Methods for Rail Break Prediction

1 Research Introduction

In railway safety prediction, we often deal with datasets that bring together many different types of information — from infrastructure characteristics like curvature and track age, to operational records such as train speed and tonnage, and sensor-based signals including vibration, acceleration, and sound.

While this richness is valuable, it also creates challenges: too many features can be redundant or noisy, which slows down computation and may even reduce prediction accuracy.

For our broken rail prediction task, feature selection becomes especially important. By filtering out irrelevant inputs and keeping only the most informative ones, we can make the prediction model, such as ResNet-Transformer model, more efficient, more accurate, and easier to interpret for practical railway maintenance decisions.

2 Research Goals

A general feature selection for classification framework is demonstrated in Figure 1. Feature selection mainly affects the training phase of classification. After generating features,instead of processing data with the whole features to the learning algorithm directly, feature selection for classification will first perform feature selection to select a subset of features and then process the data with the selected features to the learning algorithm

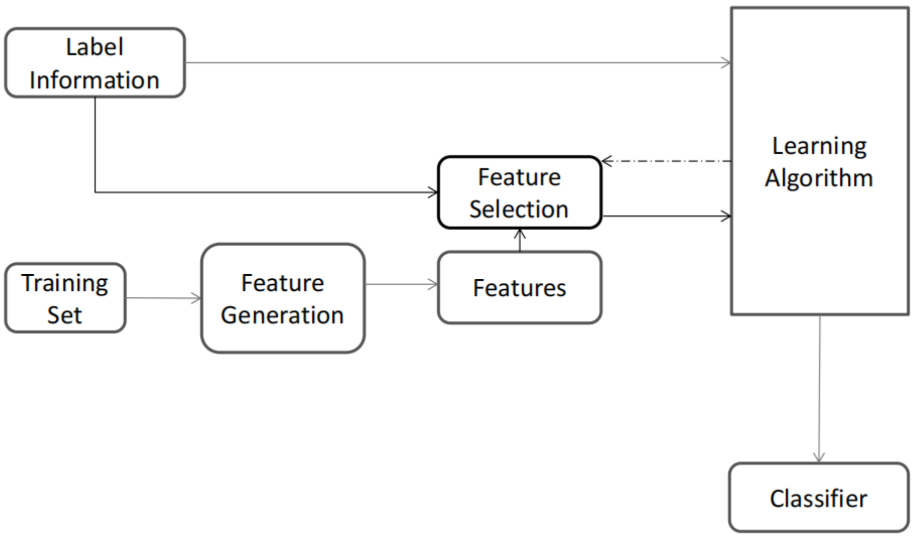

Figure 1. A General Framework of Feature Selection for Classification

The goal of feature selection in this project is to enhance prediction of rail breaks by:

1. Reducing redundancy among correlated features.
2. Increasing computational efficiency and robustness of neural network–based classifiers such as ResNet-Transformer.

3 Current Dataset and Feature Characteristics

The dataset used in this project integrates multiple sources of railway operation and condition data:

3.1 DataSet

Railbreaklocations	Defines each track section at a 20m resolution, identified by BaseCode and kilometer start/end markers
Tonnagedata	Records accumulated train tonnage over each section within defined time windows
Wagondata	Contains rich sensor readings from the Instrumented Coal Wagon (ICW)
Trainingcontext	Maps the above features to target labels, where the binary outcome indicates whether a rail break occurs within the next 30 days

3.2 Feature Characteristics

The features can be grouped into seven categories:

1. **Spatial Information**

BaseCode
SectionBreakStartKM
SectionBreakFinishKM
KMLocation
Curvature
Track_Offset

Meaning: They describe the track section's location, curvature, and geometry, providing the static background for assessing rail break risks.

2. **Operational & Control Data**

Speed

BrakeCylinder
IntrainForce
ICWVehicle
FileLoadStatus
RecordingDateTime
RecordingDate

Meaning: These features show the train’s operating state, such as speed, braking pressure, and traction force, which directly affect track stress and load conditions.

3. **Load & Twist Measurements**

Twist14m
Twist2m
LP1,LP2,LP3,LP4

Meaning: They include the rail’s average twist forces and loads at different points, measuring how evenly the track carries stress and where local stress builds up — both key for predicting breaks.

4. **Dynamic Response**

BounceFrt, BounceRr
BodyRockFrt, BodyRockRr

Meaning: They describe the track’s vibration and the train’s rocking during operation, reflecting track stability and the interaction between train and track.

5. **Acceleration Signals Fields**

Acc1, Acc2, Acc3, Acc4
Acc1_RMS, Acc2_RMS, Acc3_RMS, Acc4_RMS
VACC, VACC_L, VACC_R

Meaning: acceleration sensors capture how the track and vehicle respond dynamically. The RMS values show overall energy levels over time, while vertical acceleration highlights vertical impacts on the track.

6. **Rail Condition Fields**

Rail_Pro_L, Rail_Pro_R

Meaning: They capture the rail's left and right profiles, reflecting wear and irregularities that are closely linked to rail fatigue and crack growth.

7. **Acoustic Features Fields**

SND,SND_L,SND_R

Meaning: Acoustic data can reveal abnormal events on the track, including friction, impacts, or early indicators of cracks.

4. Survey of Feature Selection Techniques

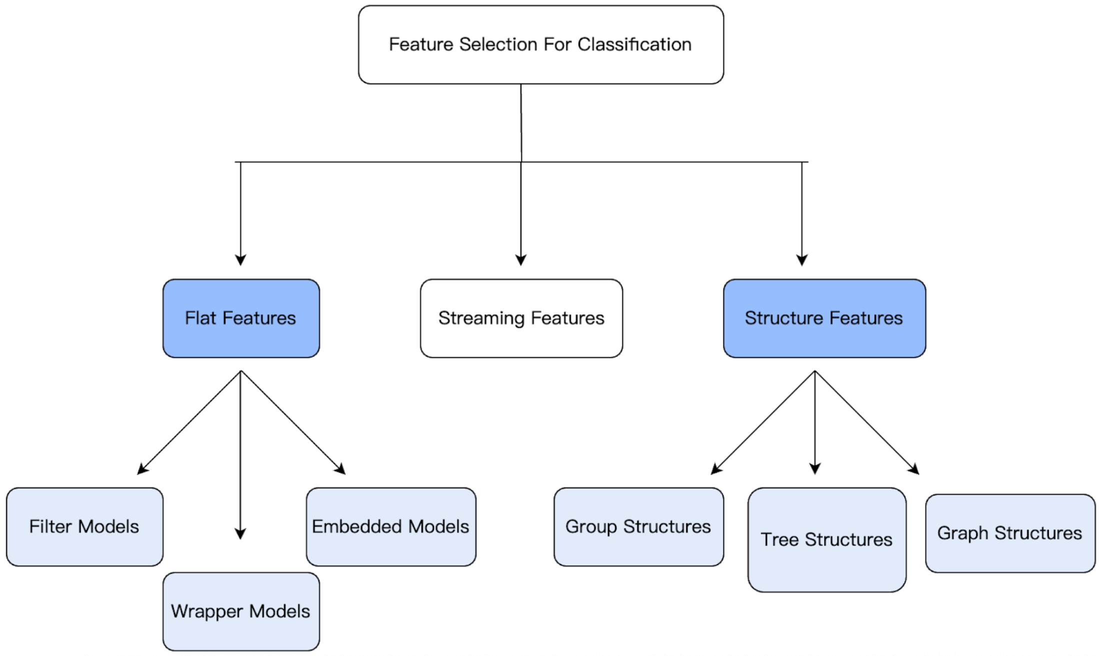

Figure 2. A General Framework of Feature Selection for Classification

In this Section, we divide feature selection for classification into three families according to feature structure: flat features, structured features, and streaming features, as shown in Figure 2.

Accordingly, our analysis first focuses on flat feature methods (filter, wrapper, and embedded models), and then moves to structured group methods. Based on this review, one theoretically suitable approach will be selected for subsequent development and implementation.

4.1 Filter Methods

The essence of filter methods is to select features based on their own statistical properties or their correlation with the target variable, without relying on subsequent learning models, and with low computational cost.

1. Continuous features and continuous target variables

Method	Advantages	Limitations
Pearson Correlation	Simple, efficient, captures linear correlation strength	Only captures linear relationships, ignores non-linear ones
Spearman Correlation	Works for non-linear & non-normal data, rank-based	Less precise for purely linear data, ignores exact values

2. Discrete features and discrete target variables

Method	Advantages	Limitations
Chi-square Test	Tests independence, interpretable, useful for categorical data	Requires large sample size, unstable with small expected frequencies

3. Mixed types: continuous features vs. discrete target variables

Method	Advantages	Limitations
ANOVA	Detects mean differences across groups, useful for discrimination tasks	Assumes normality & equal variance, misses complex non-linear effects
Mutual Information	Captures both linear & non-linear dependencies, flexible, general use	Higher computational cost, less efficient on very large datasets

In the broken rail prediction project, most features are continuous while the target is categorical, making **ANOVA** and **Mutual Information** the most suitable feature selection methods. Mutual Information is particularly valuable as it captures both linear and non-linear dependencies, which reflect the complex dynamics of track–train interactions.

4.2 Wrapper Methods

Wrapper methods use the training model as the evaluator and take model performance as the measure of feature quality. The drawback is that if there are N features, an exhaustive search would require 2^N evaluations, which is impossible to implement. Feasible wrapper methods are mainly divided into sequential selection algorithms and heuristic search algorithms(Chandrashekar & Sahin 2014, p.19). The detailed introduction is given below.

4.2.1 Wrapper Methods: Sequential selection algorithms

Method	Description	Advantages	Limitations
Sequential Forward Selection (SFS)	Start with empty set, add features one by one based on best score.	Simple, fast for small feature sets.	Greedy, may keep redundant features.
Sequential Backward Selection (SBS)	Start with all features, remove least useful ones step by step.	Consider interactions between features.	Very slow with large feature sets.
Sequential Floating Forward Selection (SFFS)	SFS + backtracking (remove redundant after adding).	More flexible, reduces overfitting to redundant features.	Higher complexity, still limited for very high dimensions.
Adaptive Sequential Floating Forward Selection (ASFFS)	Adds/removes groups of features at once, based on performance metrics (AUC, F1, etc.).	Balances efficiency and accuracy, adaptive search.	Complexity increases with large datasets.

4.2.2 Wrapper Methods: Heuristic search algorithms

Method	Description	Advantages	Limitations
Genetic Algorithm (GA)	Randomly generate feature sets, evolve via selection, crossover, mutation.	Can escape local optima, good for large search space.	Computationally expensive, slower convergence.
CHCGA (Improved GA)	Enhanced GA with elitist selection, HUX crossover, mutation strategies.	Faster convergence, maintains diversity, avoids stagnation.	More complex design, tuning required.

In our project, the feature space is mostly within a few dozen dimensions. Therefore, sequential selection methods (SFS, SBS, SFFS, etc.) are more practical, offering effective feature filtering with manageable computational cost while retaining important interactions.

In contrast, heuristic methods (GA/CHCGA) are better suited for high-dimensional or complex non-linear scenarios, as they can avoid local optima but require higher computational effort.

For our project, If we only consider within the scope of Wrapper Methods, sequential approaches appear more applicable, while heuristic methods remain a potential option for future exploration.

4.3 Embedded Methods

Embedded Models embedding feature selection with classifier construction, have the advantages of wrapper models - they include the interaction with the classification model and filter models - they are far less computationally intensive than wrapper methods.

Method	Description	Advantages	Limitations
Pruning Methods （Recursive Feature Elimination)	Iteratively removes least important features using model weights (e.g., SVM).	Simple and effective; works well with small/medium datasets.	Computationally expensive for large feature sets.
Built-in Mechanism Models (ID3, C4.5)	Decision trees select features by information gain or entropy.	Automatically embedded in model training; interpretable.	Sensitive to noise and overfitting; less effective for high-dimensional data.
Regularization Models (Lasso, Adaptive Lasso, Elastic Net, Bridge Regularization)	Add penalties to coefficients to enforce sparsity and select features.	Handle high-dimensional data; robust to multicollinearity; widely used in linear models.	Limited in capturing non-linear relationships; performance depends on penalty choice.

In the context of our project, three categories of embedded methods were considered.

1. **Pruning methods** (e.g., RFE with SVM) are intuitive and effective for initial feature screening, but their computational cost grows rapidly with large-scale railway data, making them less suitable as the main approach.

2. **Tree-based models** (ID3, C4.5) provide strong interpretability and can highlight key features such as speed, load, and curvature, but they are sensitive to noise and limited in capturing complex temporal or nonlinear dependencies.

3. **Regularization models** (Lasso, Elastic Net, etc.) are the most applicable for this project, as they handle high-dimensional data with correlated features effectively. While Lasso itself is linear and cannot fully capture nonlinear relations, this limitation can be addressed by **incorporating L1 or Group Lasso regularization within neural networks**. This allows the model to learn nonlinear temporal patterns while simultaneously performing feature sparsification. In this way, the framework retains the predictive strength of neural networks while improving feature selection and interpretability, making regularization-based methods the most practical choice for broken rail prediction.

4.4 Overall Analysis

In this project, three families of feature selection methods—Filter, Wrapper, and Embedded—were evaluated.

1. Filter methods (e.g., correlation, mutual information) are simple and efficient, especially in the preprocessing stage. They help quickly remove irrelevant or redundant variables, but since they evaluate features independently, they cannot fully capture interactions or temporal dependencies in railway data.

2. Wrapper methods provide higher accuracy by evaluating feature subsets through model performance. However, for railway time-series data, they are not practical as the primary approach due to extremely high computational cost and the risk of temporal leakage during repeated training and validation. Thus, they are only suitable as auxiliary tools in small-scale validation.

3. Embedded methods strike the best balance for this project. Regularization-based approaches, such as Lasso, Elastic Net, integrate feature selection into model training, which is computationally more efficient than wrappers and more robust than filters. Moreover, when combined with neural networks, L1/Group Lasso can enforce sparsity while still capturing nonlinear temporal dependencies.

Conclusion:

For our prediction, Embedded methods are the most suitable, supported by Filter methods for preprocessing and potentially assisted by Wrapper methods in small-scale evaluations.

However, since standard Lasso has limitations in capturing grouped and correlated features, it may not fully exploit the structure of railway sensor data, where features naturally fall into groups (e.g., acceleration channels, load points, vibration sensors). Therefore, we propose adopting **Group Lasso** within the Embedded framework, as it can retain or discard entire groups of related features, leading to more interpretable results and improved robustness for broken rail prediction.

5. Next Plan

1. Implement the Group Lasso algorithm within the task module of the InsightFactory platform.

2. Collaborate with peers researching deep learning networks to explore how Group Lasso can be integrated into neural frameworks, e.g., within the Transformer attention mechanism or as an input-layer filter.

3. If time permits, implement and compare additional algorithms (e.g., Wrapper methods) to evaluate performance differences.

6. reference

Chandrashekar, G & Sahin, F 2014, ‘A survey on feature selection methods’, Computers & Electrical Engineering, vol. 40, no. 1, pp. 16–28.

Tang, J., Alelyani, S. and Liu, H. (2014) ‘Feature Selection for Classification: A Review’, in Data Classification: Algorithms and Applications. Chapman and Hall/CRC, pp. 37–64.

Gui, J., Sun, Z., Ji, S., Tao, D. and Tan, T. (2017) ‘Feature Selection Based on Structured Sparsity: A Comprehensive Study’, IEEE Transactions on Neural Networks and Learning Systems, 28(7), pp. 1490–1507. https://doi.org/10.1109/TNNLS.2016.2551724

Wang, X., Dai, J. & Liu, X., 2025. A spatial–temporal neural network based on ResNet-Transformer for predicting railroad broken rails. Advanced Engineering Informatics, 65, 103126. https://doi.org/10.1016/j.aei.2025.103126