Explainable Dynamic Graph Neural Networks for Predictive Maintenance in Vehicle Chassis Systems

You Wu, Junwei Su, Sirui Zhang, and Zhijian Li
Data Science Program
BNU-HKBU United International College
Zhuhai, China
{r130026160, r130026129, r130026199}@mail.uic.edu.cn, zhijianli@uic.edu.cn

***Abstract*—Predictive maintenance is vital for commercial vehicle fleets to minimize unexpected downtime and curb emergency repair costs. Although standardized fault codes (SPN/FMI) aid in diagnosing failures, their limited temporal and spatial consistency poses challenges for conventional time-series approaches to identifying high-cost faults. This paper introduces a *Hybrid Node-level Relationship-based Graph Convolutional Network with Random Forest* (NRP-GCN-RF), which encodes fault code interactions as a graph to capture non-temporal dependencies more effectively.**

Unlike prior statistical models and basic SPN-FMI integrations, our method integrates spatial, operational, and co-occurrence patterns within a structured learning framework, improving the representation of complex failure behaviors. By leveraging ensemble techniques such as Random Forest and XGBoost, NRP-GCN-RF achieves a 98.93% overall accuracy, with precision for high-cost failures increasing from 60% to 95% and recall improving by 25%. The proposed approach also addresses class imbalance and refines emergency fee estimation, thereby offering a robust solution for proactive fleet maintenance planning.

These results highlight the effectiveness of graph-based modeling with heterogeneous fleet data, providing an interpretable and scalable architecture for commercial vehicle fault prediction.

***Index Terms*—Fault Prediction, Graph Neural Networks, Association Rule Mining, Vehicle Maintenance, Cross-Chassis Learning**

I. Introduction

Commercial vehicles play a vital role in industries such as logistics, transportation, and construction, where unexpected failures can result in substantial financial losses, operational delays, and safety risks. Breakdowns occurring during transit often require emergency services, including roadside repairs, towing, or urgent part replacements, leading to significant unplanned costs. These expenses can escalate rapidly, particularly in high-utilization fleets, where downtime directly impacts revenue.

Predictive maintenance has emerged as a key strategy to mitigate these risks by identifying potential failures before they lead to major disruptions. Fault codes, specifically *Suspect Parameter Number* (SPN) and *Failure Mode Indicator* (FMI), provide essential diagnostic information for critical vehicle components such as engines, transmissions, and braking systems. However, real-world fault logs pose several challenges. First, faults occur as discrete events rather than at regular intervals, resulting in inhomogeneous sequences that traditional time-series models struggle to handle. Second, fault records often contain noise and missing values, complicating direct modeling. Lastly, high-cost but rare failures suffer from class imbalance, making them difficult to predict with conventional methods.

Existing approaches relying on chronological fault sequences and single-variable patterns often exhibit limitations in handling these challenges. To overcome these issues, we propose **NRP-GCN-RF**, a hybrid framework that integrates graph-based learning with machine learning models for predictive maintenance. Unlike sequential methods, our approach models fault codes as a structured graph, where nodes represent codes and edges capture co-occurrence relationships. This structure enables the model to effectively capture interdependencies between subsystems while being robust to irregular fault occurrences.

Furthermore, we incorporate spatial, operational, and aggregated features using *Random Forest* and *XGBoost* to enhance predictive performance, particularly for high-cost failures. By leveraging relational learning and ensemble-based feature integration, NRP-GCN-RF improves predictive accuracy and provides insights into fault progression patterns, facilitating more reliable emergency cost forecasting.

II. Related Work

Existing vehicle fault detection methods can be broadly categorized into statistical approaches, sequence modeling techniques, and graph-based frameworks, each aiming to capture different aspects of complex fault patterns. However, these methods often face difficulties when confronted with dynamic, multi-fault scenarios and intricate spatiotemporal dependencies. Recent efforts to integrate multi-task learning, cross-platform strategies, and adaptive thresholding highlight a growing interest in more holistic solutions but also reveal persistent gaps in explainability and robust performance.

Traditional approaches generally rely on statistical analyses to detect emerging fault patterns. For instance, Peng et al. [7] employ frequency analysis and association rule mining to isolate high-occurrence fault codes, while Zhang et al. [8] use sliding window analysis to capture short-term fault trends in vehicle chassis systems. Although these methods

(Corresponding author: zhijianli@uic.edu.cn)

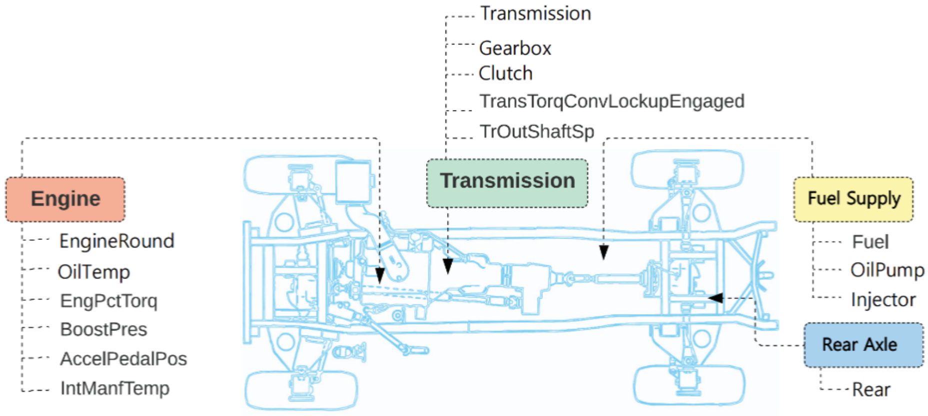


Fig. 1: Reference Diagram of Automotive Systems and Components.

TABLE I: Fault Clustering Rules Based on Keywords

No.	Keywords in Fault Description
1	Rear Axle, Axle
2	Engine, Oil
...	...
8	ECU, Electronics, Motor
9	Other Fault Descriptions

have demonstrated the value of historical data for preliminary fault detection, they typically assume stationary conditions and rely heavily on rigid thresholds, which can lead to suboptimal performance in rapidly changing operational environments.

Building on these statistical foundations, sequence modeling approaches leverage recurrent architectures to encode temporal dependencies. Tang et al. [9] proposed a GRU-based framework to track evolving fault events across continuous vehicle usage, and Wang et al. [10] combined Word2Vec embeddings with Transformers to derive semantic relationships among diagnostic SPN/FMI codes. While these methods capture contextual and sequential nuances, they often overlook the broader dependencies between code occurrences, leading to gaps in modeling global relational structures among fault events.

Graph-based techniques address some of these limitations by modeling relationships and interactions between various components. Sun et al. [11] introduced a heterogeneous graph neural network (GNN) to account for spatial dependencies among fault sources, whereas Zhou et al. [12] explored dynamic graph convolutional networks in industrial settings. Despite promising results, most existing graph methods remain either static or narrowly focused on single-task objectives, limiting their adaptability in multi-fault scenarios and restricting their potential for cross-platform generalization. Recent attempts to incorporate multi-task learning in ensemble models [13]–[16] have shown that shared representation learning can improve predictive performance, but knowledge transfer across different vehicle types or operational conditions remains a challenge.

Another line of work seeks to adjust detection thresholds dynamically. Previous studies, such as [4] and [8], have revealed how static thresholding can lead to both false alarms and missed detections when environmental or operational factors shift. Approaches that adapt thresholds based on real-time data streams offer better resilience but still lack a unified framework that combines robust modeling of spatiotemporal relationships with dynamic threshold adjustments.

In summary, while prior research has contributed valuable techniques—ranging from statistical analyses and temporal modeling to graph-based methods and multi-task learning—there is still a lack of an integrated solution that addresses dynamic conditions, multi-fault interactions, cross-platform constraints, and adaptive thresholds simultaneously. The proposed framework builds on these existing works by employing dynamic graph neural networks within a multi-task setting and by integrating an adaptive threshold mechanism, thus offering a more comprehensive and explainable solution for predictive maintenance in vehicle chassis systems.

III. Research Methodology

A. Data Preprocessing

1) Fault Dataset: The fault dataset consists of multiple key fields, including *ChassisNumber, SystemType, NameAndModel, FaultName, FaultCount, SPN, FMI, FaultStartTime*, and *FaultEndTime*. While *FaultName* includes up to 701 unique entries, only 135 distinct *SPN-FMI* pairs are observed. This substantial reduction in dimensionality suggests that using SPN-FMI pairs as primary fault identifiers can enhance specificity and reduce redundancy.

A frequency analysis reveals considerable variation in SPN-FMI occurrence, with some combinations appearing disproportionately more frequently than others. To ensure data reliability, the fault records were sorted by *ChassisNumber* and *FaultStartTime*, forming chronological sequences for each vehicle. Rare SPN-FMI combinations below a predefined occurrence threshold were removed to focus on more representative fault patterns.

2) Maintenance Dataset: The maintenance dataset includes key fields such as *IdentificationNumber, ChassisNumber, ProductionDate, FaultDate, CheckDate, FaultCode, FaultDescription, PartName, and EmergencyFee*. Given the large variety of failed parts (over 200 unique entries), a keyword-based natural language processing (NLP) method was utilized to classify them into nine broader *Fault Categories*, following industry-standard automotive references. This classification ensures that critical diagnostic information is preserved while enhancing the dataset's structure for modeling tasks.

3) Maintenance Dataset: The maintenance dataset includes key fields such as *IdentificationNumber, ChassisNumber, ProductionDate, FaultDate, CheckDate, FaultCode, FaultDescription, PartName, and EmergencyFee*. Due to the large variety of fault types (over 200 unique entries), a keyword-based natural language processing (NLP) method was employed to classify these faults into nine broader *Fault Categories*, guided by industry-standard automotive literature. This approach simplifies the dataset, preserves diagnostic relevance, and enhances modeling efficiency.

The NLP classification grouped faults based on keywords extracted from the fault descriptions. Table I presents a simplified example of the clustering rules applied to categorize the fault data.

To provide a clearer perspective on how the fault categories relate to vehicle components, a reference diagram of automotive systems and their associated features is depicted in Figure 1. This visual aid highlights the major components of a vehicle, linking them to their respective fault categories.

This classification process enables the extraction of meaningful patterns within the fault data, allowing the model to focus on component-specific failure mechanisms. By systematically grouping faults, the model can better manage imbalanced fault distributions and improve overall prediction accuracy.

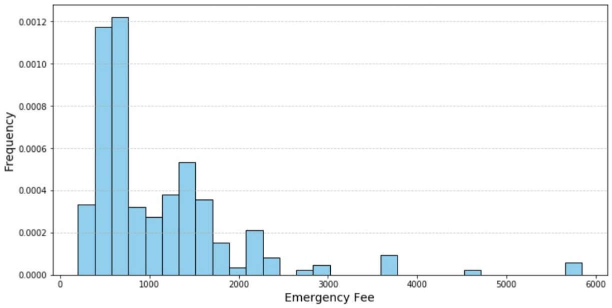


Fig. 2: Frequency distribution of emergency fees.

To better interpret emergency fee data, maintenance costs were grouped into four predefined price brackets: **0–200**, **200–1,000**, **1,000–2,000**, and **above 2,000** units. Figure 2 demonstrates that the majority of emergency fees fall below 2,000 units, with a pronounced peak in the lowest range. This suggests that most maintenance operations incur relatively low costs, while high-cost repairs remain less frequent.

Building on this, Figure 3 further categorizes emergency fees into the four brackets, aligning with the observed distribution in Figure 2. This step highlights the consistency between frequency and categorical distributions, facilitating subsequent fault analysis and cost prediction.

Finally, Figure 4 examines the relationship between SPN-FMI combinations and emergency fees. By identifying common and distinct fault combinations across different fee categories, this analysis reveals patterns linking specific fault codes to cost brackets. These insights establish a direct connection between fault characteristics and maintenance costs, providing a foundation for predictive modeling and decision-making.

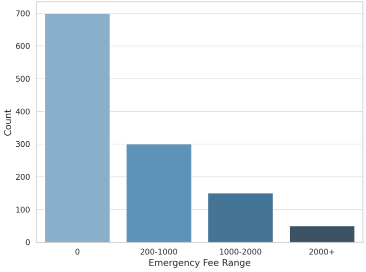


Fig. 3: Categorical distribution of emergency fees.

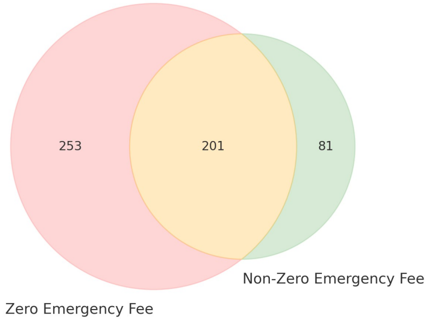


Fig. 4: Relationship between SPN-FMI combinations and emergency fees.

B. Data Integration and Label Construction

To integrate the two datasets, entries were matched based on *ChassisNumber* and relevant date fields. By linking *FaultStartTime* from the fault dataset to *FaultDate* in maintenance records, we established a direct connection between early fault occurrences and confirmed repairs. This integration enables three core prediction tasks:

- **Emergency Fee Requirement**: Binary classification indicating whether an emergency fee was incurred.
- **Emergency Fee Category**: A four-category classification based on cost.
- **Fault Category**: Predicting the fault type among the nine NLP-derived categories.

C. Frequent Fault Combination Analysis

An important observation from the integrated dataset is the significant variation in SPN-FMI occurrence proportions across different vehicles. For instance, the combination *523004-16* appears in over 50% of fault sequences in some chassis, whereas *792-2* is only present in 20% of cases. This suggests that certain fault combinations may have stronger predictive relevance for future failures or costly repairs. To systematically analyze these patterns, we employed an improved

TABLE II: Classification of SPN-FMI Occurrence Ranges

Range	Proportion	Characteristics
Low	0–10%	Minimal variation, stable associations
Medium	10–25%	Moderate variation, context-dependent patterns
High	>25%	Large fluctuations, potentially transient dependencies

association rule mining approach, optimizing the *Apriori* algorithm through matrix compression and trie-based indexing. The key steps are as follows:

1) **Combination Filtering**: SPN-FMI pairs with low occurrence frequency were removed to ensure statistical significance.
2) **Boolean Matrix Representation**: A sparse matrix was constructed, where each row represents a chassis and each column represents the presence of a fault code.
3) **Matrix Compression**: Redundant and infrequent patterns were pruned through row and column compression, reducing computational complexity.
4) **Frequent Itemset Mining**: The optimized *Apriori* algorithm was applied to extract co-occurring fault codes with a minimum support threshold of 1%.
5) **Trie-based Indexing**: Identified frequent patterns were efficiently stored using a trie structure for rapid retrieval and pattern analysis.

The following flowchart in Fig. 5 illustrates the improved Apriori algorithm process.

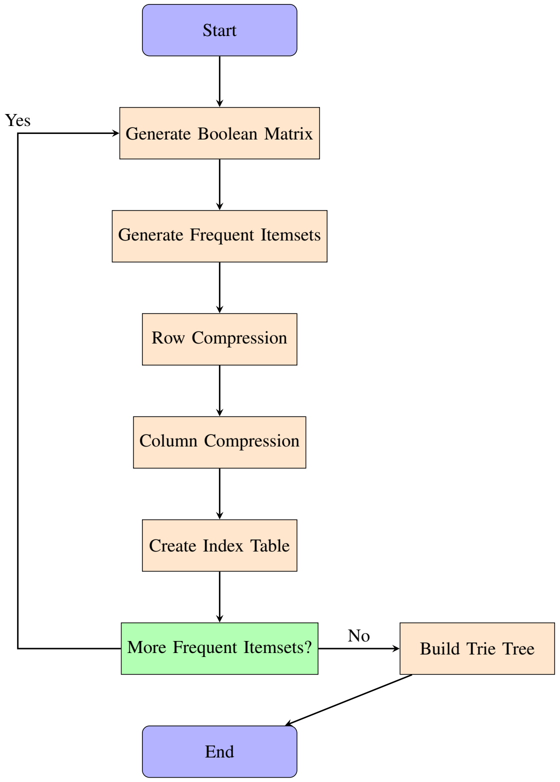


Fig. 5: Flowchart of the Improved Apriori Algorithm

The extracted fault combinations were then categorized into three stability ranges, as summarized in Table II.

By integrating matrix compression and trie-based indexing, the proposed approach improves computational efficiency while preserving association rule accuracy. The identified high-risk fault patterns provide actionable insights for predictive maintenance, enabling early-warning mechanisms, reduced unexpected downtime, and improved resource allocation in vehicle maintenance.

IV. Design and Implementation

We evaluated three approaches for fault prediction: Support Vector Machines (SVM), Random Forests (RF), and a hybrid model combining Graph Neural Networks (GNNs) with RF. Each method utilized different data processing strategies and faced distinct challenges.

A. Support Vector Machines (SVM)

TABLE III: Simulated dataset showing the ”large p, small n” problem with high-dimensional sparse features.

Chassis	FaultLabel	523004-16	520415-0	792-2	523014-8	…
K1036118	1	1	1	0	0	…
K1036118	2	1	1	0	0	…
L8043320	1	0	0	1	1	…
L8043320	2	0	0	1	1	…
L8058404	1	0	0	0	0	…

SVM models were applied using one-hot encoded fault codes, which expanded the feature space into a high-dimensional sparse matrix. This approach introduced the ”large p, small n” problem, where the number of features (p) greatly exceeds the number of samples (n). The resulting sparsity of the dataset led to difficulties in kernel-based approaches. The inner product between vectors became negligible, and as a result, the SVM model suffered from underfitting. In addition, the large feature space failed to capture meaningful patterns among fault codes. To address this, we focused on selecting high-proportion fault codes, which reduced dimensionality and helped the model focus on more diagnostic patterns.

B. Random Forests (RF)

RF was applied to a structured dataset, as shown in Table IV, which includes fault combinations, counts, and proportions. RF handled tabular data effectively but faced challenges with imbalanced classes. To address this, SMOTE was applied to generate synthetic samples for minority fault classes:

$$\hat{x}_{\text{new}} = x_{\text{minority}} + \lambda \cdot (x_{\text{neighbor}} - x_{\text{minority}}),$$

where $\lambda \sim \text{Uniform}(0, 1)$. While SMOTE improved minority class performance, RF alone struggled to capture the relational dependencies between fault instances.

TABLE IV: Restructured Fault Dataset Summary with Key Attributes

ChassisNumber	FaultLabel	High Proportion Combinations	Counts	AvgLongitude	AvgLatitude	...
K1036118	1	['523004-16', '520415-0']	[5, 5]	117.2451	39.3165	...
K1036118	2	['523004-16', '520415-0']	[7, 6]	117.2066	39.3501	...
		...				
L8043320	1	['792-2', '523014-8', ...]	[14, 14, ...]	122.9887	41.0847	...
L8058404	1	['518110-5', '518118-5', ...]	[14, 13, ...]	103.0202	43.2948	...

Note: Some columns and values are truncated for brevity.

C. Hybrid NRP-GCN-RF-RF Model

The proposed hybrid model integrates NRP-GCN-RF with Random Forests (RF) to leverage both relational and feature-based insights from fault data. The fault relationships are encoded as a graph $G = (V, E)$, where nodes V represent fault instances and edges E capture dependencies based on spatial, operational, and statistical relationships. These edges are weighted as:

$$w_{ij} = \alpha \times \text{similarity}(v_i, v_j) + \beta \times \text{correlation}(v_i, v_j),$$

where α and β control the respective contributions of spatial/temporal similarity and statistical dependence.

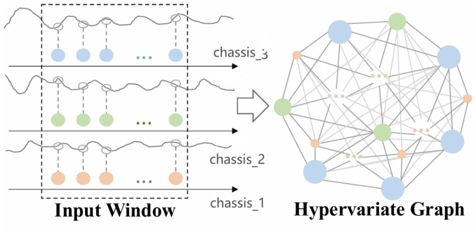


Fig. 6: Graph conversion

Graph embeddings are obtained using a Graph Convolutional Network (GCN), where the node representation at layer k is updated as:

$$h_v^{(k)} = \sigma\left(\sum_{u\in\mathcal{N}(v)} \frac{1}{\sqrt{d_v d_u}} W^{(k)} h_u^{(k-1)}\right),$$

where d_v and d_u are the degrees of nodes v and u, respectively. This aggregation process ensures that both local and global fault relationships are captured.

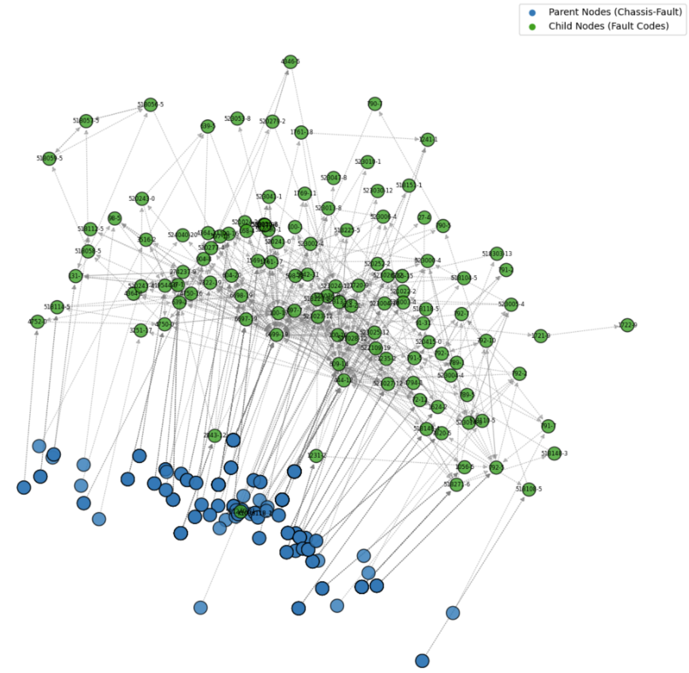


Fig. 7: Clustered Fault Graph. Blue nodes represent fault sequences interacting with green nodes (fault codes) across different chassis, illustrating fault aggregation patterns.

D. Graph Construction and Feature Engineering

To structure the dataset into a graph format, edges are defined based on spatial proximity and operational similarity. Nodes are linked when their geospatial attributes (longitude, latitude) are close or when they share significant operational characteristics, such as recurring fault patterns and frequency-based dependencies. Additionally, node features incorporate one-hot encoded fault combinations, normalized counts, and geographic attributes:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma},$$

where μ and σ represent the mean and standard deviation, ensuring feature standardization. These features are processed using PyTorch Geometric to generate structured embeddings.

E. Graph Visualizations

Understanding the fault dependencies within the dataset requires effective visualization of the graph structure. Figure 8 presents a focused view of a single chassis, where each blue node corresponds to a fault sequence, and green nodes denote specific fault codes. This hierarchical structure highlights how certain fault sequences repeatedly co-occur with particular faults, reinforcing the need for relational modeling.

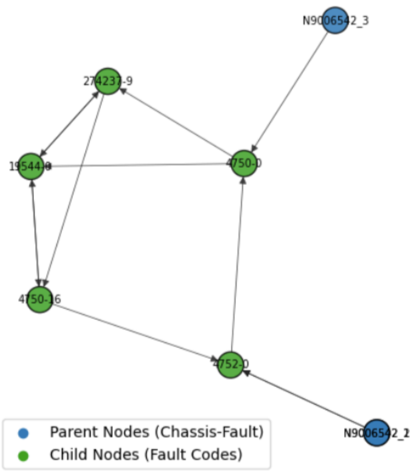


Fig. 8: Directed Graph of Fault Sequences for a Single Chassis.

Expanding beyond a single chassis, Figure 7 visualizes fault clustering across multiple chassis, capturing broader fault propagation patterns. The clustered graph structure reveals dense fault interactions, where blue nodes (fault sequences) frequently share edges with multiple green nodes (fault codes), reinforcing the need for multi-hop message passing within the GNN framework.

Finally, Figure 9 encapsulates the entire dataset's fault relationships in a comprehensive fault network. Here, weighted edges between nodes depict statistically significant relationships, emphasizing both frequent and rare fault interactions. The circular arrangement further aids in identifying structural patterns within the dataset, offering an intuitive view of high-degree fault hubs and less frequent fault occurrences.

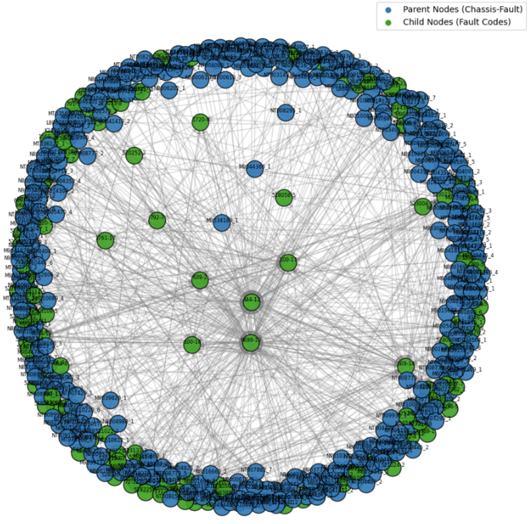


Fig. 9: Complete Fault Relationship Graph. Blue nodes indicate parent nodes (chassis-level faults), and green nodes represent child nodes (specific fault codes). Weighted edges emphasize significant fault correlations.

Through these progressively complex visualizations, we gain a deeper understanding of the fault propagation mechanisms, the significance of multi-hop dependencies, and the necessity of modeling both local and global fault interactions. By incorporating these insights, our hybrid NRP-GCN-RF-RF model achieves more accurate and interpretable fault predictions.

F. Workflow Summary

Figure 10 summarizes the complete fault prediction workflow, integrating preprocessing, graph construction, GNN-RF modeling, transfer learning, and adaptive anomaly detection.

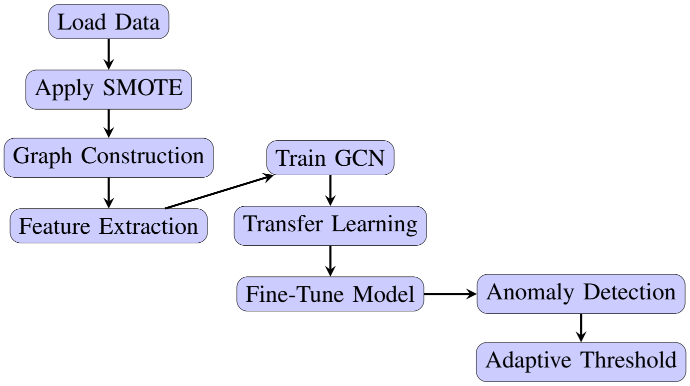


Fig. 10: Refined Data Processing and Model Workflow Including SMOTE

V. Validation

In this section, we evaluate our model on three prediction tasks: **(1)** Y_1**: HasEmergencyFee**, **(2)** Y_2**: EmergencyFeeCategory**, **(3)** Y_3**: FaultCategory**. We present ROC curves, classification metrics, and cross-validation results, highlighting strengths and limitations in real-world scenarios.

A. Fault Code Association Analysis

The Apriori algorithm [23] was employed to discover frequent SPN-FMI combinations and generate association rules, providing insights into co-occurring faults that may indicate more complex or cascading issues.

1) Apriori Method Overview: Apriori iteratively identifies frequent itemsets (fault code pairs or sets) that exceed a user-defined support threshold and then derives association rules that satisfy a minimum confidence level. Two important measures are:

a) Support: For an itemset A, the support quantifies how often A appears in the dataset:

$$\text{supp}(A) = \frac{|T_A|}{|T|}, \tag{1}$$

where $|T|$ is the total number of transactions (fault records), and $|T_A| \subset T$ is the subset containing A. For instance, a support of 0.0196 implies that 1.96% of all records include the specified itemset.

b) Lift: Lift measures the strength of the association between itemsets A and B:

$$\text{lift}(A \Rightarrow B) = \frac{\text{supp}(A \cup B)}{\text{supp}(A) \cdot \text{supp}(B)}. \tag{2}$$

A lift value greater than 1 indicates a positive correlation; higher values (e.g., above 30) imply that A and B frequently co-occur beyond what chance alone would predict.

2) Strongly Associated Rules: Table V lists itemsets with particularly high support, confidence, and lift, indicating nearly inseparable co-occurrences. For example, the pair (1761-18) and (1241-1) demonstrates a lift of 98, revealing a very strong correlation between these two fault codes.

TABLE V: Strongly Associated Rules via Apriori Analysis

Antecedents	Consequents	Support	Confidence	Lift
(1761-18)	(1241-1)	0.0102	1.0	98.0
(520243-0)	(639-5)	0.0102	1.0	98.0
(523053-8)	(523056-8)	0.0204	1.0	49.0
(518112-5)	(131-7)	0.0204	1.0	49.0
(4794-1)	(523006-4)	0.0102	1.0	32.7

Key Observations:

- *High-Lift Pairs:* Itemsets like (1761-18) and (1241-1) exhibit lift close to 98, suggesting these codes almost always appear together.
- *Fault Overlaps:* Multiple items in the table belong to related subsystems, hinting at potential chain-reaction faults that can escalate maintenance costs.

3) Moderately Associated Rules: Table VI lists itemsets with lower but still notable lift values, indicating significant but not absolute co-occurrences. For instance, (524040-20) and (597-13) share a lift of 51, implying they occur together far more often than random chance would suggest.

TABLE VI: Moderately Associated Rules via Apriori Analysis

Antecedents	Consequents	Support	Confidence	Lift
(524040-20)	(597-13)	0.0196	1.0	51.0
(518108-5)	(790-5)	0.0196	1.0	34.0
(521022-2)	(520252-2)	0.0588	0.86	14.6
(1720-9)	(1624-2)	0.0882	0.90	10.2
(518114-5)	(4750-16)	0.0294	0.75	6.38

Key Observations:

- *Sensor-Control Unit Links:* Pairs like (518108-5) and (790-5) may indicate collaboration between different sensor lines in the same or adjacent modules.
- *Mixed Severity:* While some itemsets involve routine warnings, others point to critical components (e.g., oil pressure, EGR cooler), emphasizing targeted checks in maintenance protocols.

4) Summary of Association Insights: The above rules highlight how certain SPN-FMI codes repeatedly co-occur, implying either shared fault origins or sequential failure pathways:

- *Frequent High-Lift Pairs*: Indicate codes that appear nearly in tandem, guiding proactive checks for connected subsystems (e.g., EECU with induction system).
- *Moderately Strong Associations*: Reveal broader patterns across various vehicle components (lighting, braking, engine subsystems), suggesting that chain failures could be anticipated by monitoring these itemsets.

Overall, Apriori-based association analysis provides an additional layer of diagnostic insight, complementing the predictive models by spotlighting co-occurrence patterns that might not be explicitly captured in a purely classification-focused framework.

B. Task-wise Performance with NRP-GCN-RF

TABLE VII: Classification Report: Y_1*: HasEmergencyFee*

Class	Precision	Recall	F1-Score	Support
0	0.91	0.84	0.87	25
1	0.94	0.97	0.96	67
Accuracy		0.9348 (on 92 samples)		
Macro Avg	0.93	0.91	0.92	92
Weighted Avg	0.93	0.93	0.93	92

TABLE VIII: Classification Report: Y_2*: EmergencyFeeCategory*

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.82	0.90	17
1.0	0.92	0.94	0.93	81
2.0	0.96	0.93	0.95	86
3.0	0.95	1.00	0.97	73
Accuracy		0.9455 (on 257 samples)		
Macro Avg	0.96	0.92	0.94	257
Weighted Avg	0.95	0.95	0.95	257

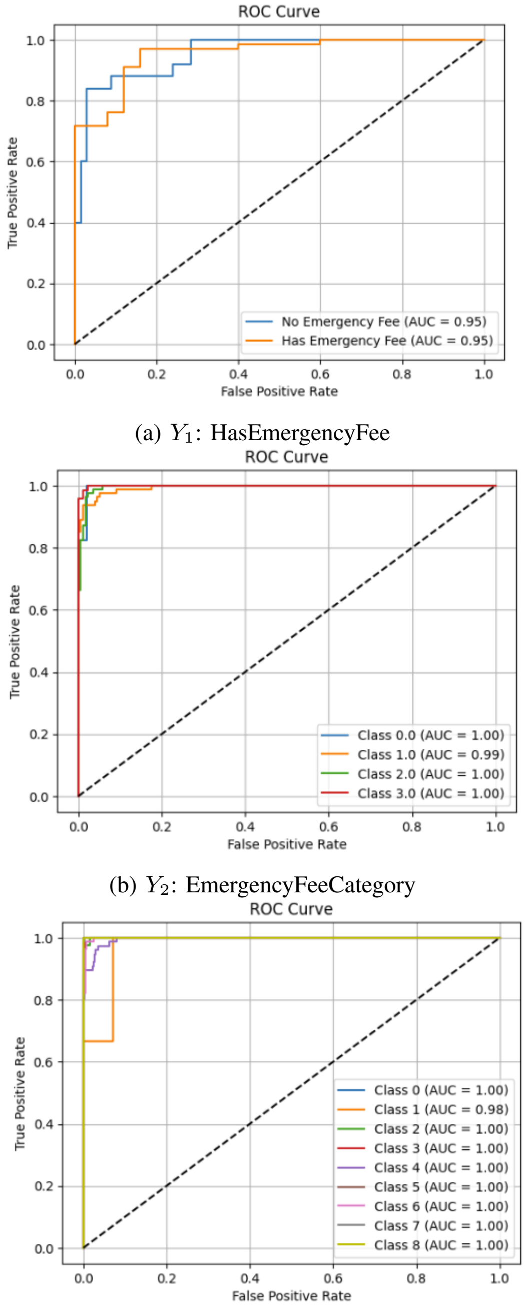
TABLE IX: Classification Report: Y_3*: FaultCategory*

Class	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	50
1	1.00	0.67	0.80	3
2	1.00	0.98	0.99	62
3	0.99	1.00	0.99	66
4	1.00	0.98	0.99	54
5	0.98	1.00	0.99	55
6	0.98	0.96	0.97	53
7	1.00	1.00	1.00	67
8	1.00	1.00	1.00	57
Accuracy		0.989(on 467 samples)		
Macro Avg	0.99	0.95	0.97	467
Weighted Avg	0.99	0.99	0.99	467

TABLE X: Overall Performance Comparison Across Three Tasks

Model	Y_1: HasEmergencyFee			Y_2: EmergencyFeeCategory			Y_3: FaultCategory		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
SVM	0.74	0.74	0.74	0.74	0.72	0.73	0.80	0.80	0.79
RF	0.90	0.90	0.90	0.92	0.92	0.92	0.95	0.95	0.95
NRP-GCN-RF	**0.94**	**0.94**	**0.94**	**0.93**	**0.93**	**0.93**	**0.98**	**0.99**	**0.99**

C. Overall Performance Comparison



(a) Y_1: HasEmergencyFee

(b) Y_2: EmergencyFeeCategory

(c) Y_3: FaultCategory

Fig. 11: ROC Curves for Y_1, Y_2, and Y_3.

Table X gives a high-level overview of the precision, recall, and F1-scores across the three tasks for SVM, Random Forest (RF), and our NRP-GCN-RF model. The table underscores NRP-GCN-RF's ability to outperform baseline models in almost every metric.

Key Insight: NRP-GCN-RF leverages the relational structure of fault codes, leading to notable gains in recall and F1-score for minority classes and high-cost fault scenarios.

D. Analysis of Results

The results in Tables VII, VIII, and IX, along with overall performance metrics in Table X, highlight the predictive effectiveness of the proposed NRP-GCN-RF model. The ROC curves in Figure 11 further confirm its strong classification ability across tasks.

SMOTE preprocessing played a key role in balancing class distributions. For Y_1: *HasEmergencyFee*, minimal application of SMOTE resulted in moderate class balance (Table VII). For Y_2: *EmergencyFeeCategory*, SMOTE improved balance but led to lower recall for Class 0 (0.82), indicating challenges for underrepresented classes. For Y_3: *FaultCategory*, SMOTE effectively addressed class imbalance, ensuring high precision and recall across all fault categories (Table IX).

NRP-GCN-RF consistently outperformed baselines (Table X). For Y_1, the F1-score of 0.94 and AUC of 0.9716 (Figure 11a) indicate strong binary classification performance. For Y_2, the model achieved an F1-score of 0.93 with near-perfect AUC values (Figure 11b), demonstrating robust multi-class classification. For Y_3, an F1-score of 0.99 highlights the model's ability to leverage graph relationships for fault categorization (Figure 11c). Cross-validation results (Table XI) confirm model robustness, with consistently high accuracies and low variance across tasks.

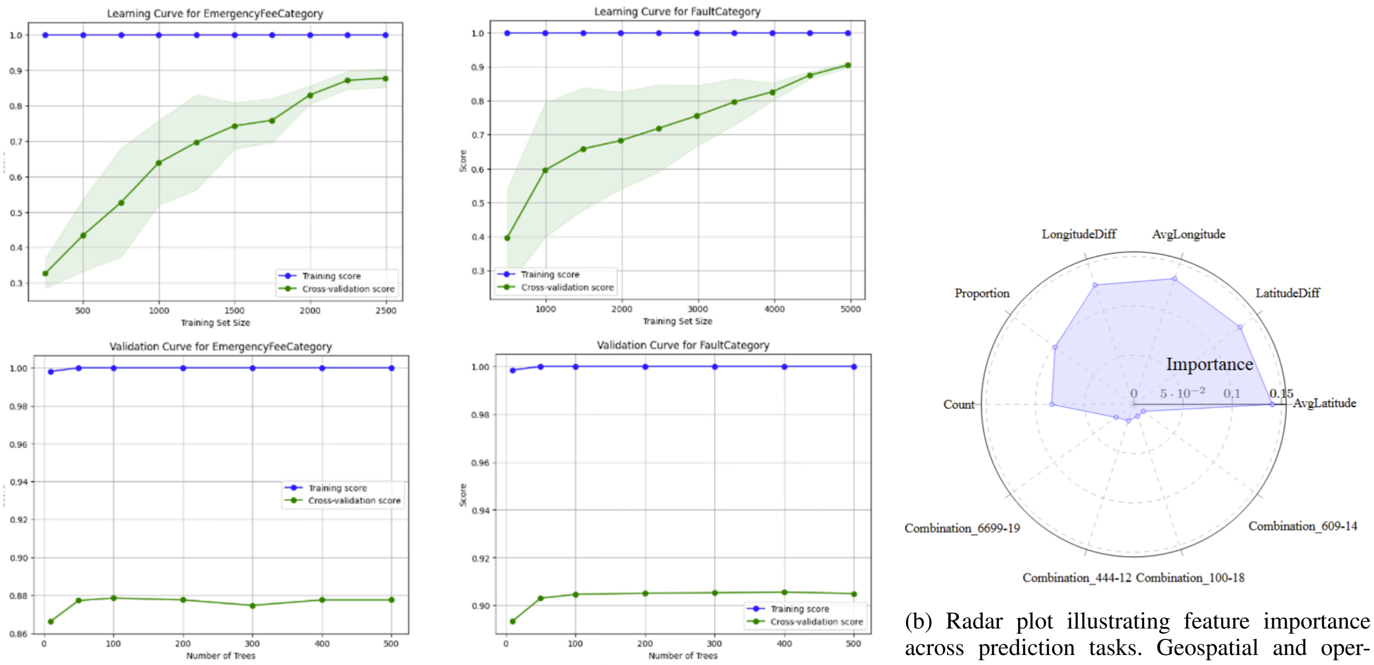
E. Fault Code Combination Analysis

Figure 12b presents a radar plot of feature importance across tasks, emphasizing the role of geographical attributes (‘AvgLatitude‘, ‘AvgLongitude‘), operational metrics (‘Proportion‘, ‘Count‘), and specific fault code combinations. Notably, ‘AvgLatitude‘ was the most influential factor in predicting emergency fee categories (Y_2), underscoring spatial patterns in maintenance cost estimation.

For Y_1: *HasEmergencyFee*, braking and gearbox-related faults (‘789-5‘, ‘1056-5‘) were strong indicators of emergency fees. The radar plot highlights ‘Proportion‘ as a key feature in identifying frequent emergency-related faults.

TABLE XI: Cross-Validation Metrics Across All Tasks with NRP-GCN-RF

Task	CV Scores	Mean Accuracy	Std. Dev.
Y_1: HasEmergencyFee	0.9130, 0.9565, 0.8913, 0.9333, 0.8889	0.9166	0.0257
Y_2: EmergencyFeeCategory	0.9690, 0.9297, 0.9609, 0.9844, 0.9375	0.9563	0.0202
Y_3: FaultCategory	0.9968, 0.9838, 0.9903, 0.9870, 0.9773	0.9870	0.0065



(a) Validation Curve

(b) Radar plot illustrating feature importance across prediction tasks. Geospatial and operational factors, along with specific fault code combinations, contribute to predictive accuracy.

Fig. 12: Side-by-side comparison of validation and metric curves.

For Y_3: *FaultCategory*, faults such as ‘1761-18‘ (engine temperature sensor failure) and ‘2843-12‘ (rear axle imbalance) were critical for accurate classification. The prominence of ‘Combination_609-14‘ and ‘Combination_6699-19‘ reinforces that certain faults strongly influence categorization.

Shared and unique fault code contributions were observed across tasks. ‘789-5‘ and ‘1761-18‘ were prevalent in both Y_1 and Y_3, indicating common underlying mechanisms, while ‘100-1‘ was exclusive to Y_2, aligning with high-cost maintenance patterns. Correlation analysis further revealed that ‘100-16‘ and ‘3251-17‘ were strongly associated with high emergency fees, while ‘444-12‘ correlated with lower-cost cases.

The radar plot underscores the significance of geospatial and frequency-based factors in fault prediction, supporting more targeted maintenance strategies to reduce costs and enhance reliability.

VI. Conclusion and Future Work

This paper presents **NRP-GCN-RF**, a hybrid model that integrates Graph Neural Networks (GNNs) with traditional machine learning to improve fault prediction accuracy. By leveraging both graph-structured relationships and tabular features, the model enhances classification performance across tasks such as *HasEmergencyFee*, *EmergencyFeeCategory*, and *FaultCategory*. The results demonstrate its advantage over baseline methods, particularly in handling imbalanced data and uncovering fault code correlations that support predictive maintenance and cost management.

Future work will focus on:

- **Dynamic Graphs:** Adapting the model to incorporate new fault data over time, capturing evolving fault patterns.
- **Explainability:** Developing interpretable methods to highlight key graph structures and features driving predictions.
- **Scalability:** Extending the approach to larger datasets and integrating additional data sources, such as sensor readings and real-time telemetry.

These improvements will further enhance the model’s practical applicability, supporting more efficient fault diagnosis and resource allocation in industrial settings.

References

[1] M. Zhang, J. Guo, X. Li, and R. Jin, "Data-Driven Anomaly Detection Approach for Time-Series Streaming Data," Key Laboratory of Remote Sensing of Gansu Province, 2020.

[2] C. Zhang et al., "A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data," University of Notre Dame, 2020.

[3] S. Neupane et al., "A Temporal Anomaly Detection System for Vehicles Utilizing Functional Working Groups and Sensor Channels," IEEE, 2022.

[4] M. Zhang, J. Guo, X. Li, and R. Jin, "Data-Driven Anomaly Detection Approach for Time-Series Streaming Data," Key Laboratory of Remote Sensing of Gansu Province, Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou, China, 2020.

[5] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data," University of Notre Dame, IN, USA; NEC Laboratories America, NJ, USA; Columbia University, NY, USA, 2020.

[6] S. Neupane, I. A. Fernandez, W. Patterson, S. Mittal, and S. Rahimi, "A Temporal Anomaly Detection System for Vehicles Utilizing Functional Working Groups and Sensor Channels," in *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)*, IEEE, pp. 24, 2022.

[7] M. C. Moura, E. Zio, I. D. Lins, and E. Droguett, "Failure and Reliability Prediction by Support Vector Machines Regression of Time Series Data," Department of Production Engineering, Federal University of Pernambuco, Brazil; Department of Energy, Polytechnic of Milan, Italy; Ecole Centrale Paris et Supelec, France.

[8] L. Biddle and S. Fallah, "A Novel Fault Detection, Identification and Prediction Approach for Autonomous Vehicle Controllers Using SVM," Connected Autonomous Vehicles Lab (CAV-Lab), Department of Mechanical Engineering Sciences, University of Surrey, 2021.

[9] A. Theissler, "Detecting Known and Unknown Faults in Automotive Systems Using Ensemble-Based Anomaly Detection," Faculty of Information Technology, Esslingen University of Applied Sciences, 2017.

[10] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[11] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access*, vol. 9, pp. 120020–120035, 2021.

[12] S. M. Namburu, M. Wilcutts, S. Chigusa, L. Qiao, K. Choi, and K. Pattipati, "Systematic Data-Driven Approach to Real-Time Fault Detection and Diagnosis in Automotive Engines," Toyota Motor Engineering Manufacturing North America, USA; Department of ECE, University of Connecticut, USA.

[13] J. Wang, C. Zhang, X. Ma, Z. Wang, Y. Xu, and R. Cattley, "A Multivariate Statistics-Based Approach for Detecting Diesel Engine Faults with Weak Signatures," College of Power and Energy Engineering, Harbin Engineering University, China; Centre for Efficiency and Performance Engineering, University of Huddersfield, UK, 2020.

[14] L.-J. Cao and F. E. H. Tay, "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 14, no. 6, pp. 1506–1518, 2003.

[15] Y. Xia and J. Chen, "Traffic Flow Forecasting Method Based on Gradient Boosting Decision Tree," in *FMSMT*, Atlantis Press, pp. 413–416, 2017.

[16] B. Biller and B. L. Nelson, "Modeling and Generating Multivariate Time-Series Input Processes Using a Vector Autoregressive Technique," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 13, no. 3, pp. 211–237, 2003.

[17] G. E. Box and D. A. Pierce, "Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models," *Journal of the American Statistical Association (JASA)*, vol. 65, no. 332, pp. 1509–1526, 1970.

[18] M. Jin, Y. Zheng, Y.-F. Li, S. Chen, B. Yang, and S. Pan, "Multivariate Time Series Forecasting with Dynamic Graph Neural ODEs," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2022.

[19] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional Neural Networks for Time Series Classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[20] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 5, no. 2, pp. 240–254, 1994.

[21] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in Time Series: A Survey," in *Proceedings of IJCAI*, 2023.

[22] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, and Y. Zheng, "Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey," *arXiv preprint*, vol. abs/2303.14483, 2023.

[23] Min Wang, Xijun Zhu, "Association Rule Analysis Based on Improved Apriori Algorithm," *School of Information Science and Technology, Qingdao University of Science and Technology*, Qingdao, Shandong, China, Received: May 18, 2021; Accepted: June 15, 2021; Published: June 22, 2021.