Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Data Mining Framework for EDC Terminal Repair Protocol: Combining Apriori and PrefixSpan

Suwandhy Praharto[1)*], Handri Santoso[2)]
[1,2)]Pradita University, Indonesia
[1)]suwandhy.praharto@student.pradita.ac.id, [2)]handri.santoso@pradita.ac.id

Submitted : May 9, 2025 | **Accepted** : May 15, 2025 | **Published** : May 20, 2025

Abstract: Electronic Data Capture (EDC) terminals are vital for financial transactions, but their repair processes often lack standardization, causing inefficiencies. Data mining techniques like Association Rule Mining (ARM) and Sequential Pattern Mining (SPM) can extract hidden patterns from service logs to inform maintenance strategies. This research addresses the limited use of these techniques within Electronic Data Capture (EDC) repair centers. Specifically, it applies Association Rule Mining (ARM) using the Apriori algorithm, and Sequential Pattern Mining (SPM) using the PrefixSpan algorithm, to optimize repair protocols based on historical repair data from PT. XYZ Indonesia. The study aimed to discover frequent fault-action-component associations and repair sequences to formulate standardized procedures. A quantitative case study analyzed 56,629 repair transactions. After data cleaning and transformation, Apriori (evaluated by support, confidence, lift) mined association rules, while PrefixSpan found frequent sequential patterns (evaluated by minimum support). Several high-confidence rules emerged: "Battery Not Charging" almost always led to "Replace Battery Pack" (≈95% confidence, lift ≈6.0), and error "2000000" (tamper indication) strongly correlated with detampering procedures and internal battery replacement (≈96% confidence, lift ≈4.9). PrefixSpan uncovered consistent repair sequences, including length-3 patterns for complex issues, with "Replace CMOS → Reinstall OS" for error "7FFFFF" being a prominent shorter sequence. Integrating these data-driven patterns into protocols and aligning inventory can improve service efficiency, reduce repair time, and enhance EDC reliability.

Keywords: Apriori; Association Rule Mining; Data Mining; EDC Repair; PrefixSpan; Repair Protocols; Sequential Pattern Mining

INTRODUCTION

Electronic Data Capture (EDC) terminals – card payment devices – form the backbone of modern financial transactions and retail operations. When EDC devices fail, prompt and effective repair is crucial for avoiding service disruptions. However, maintenance centers often lack systematic protocols that rely on technician experience. Data mining offers a solution: by analyzing large maintenance logs, hidden patterns of faults and remedies can be revealed (Valtierra-Rodriguez et al., 2020). In industry, data-driven maintenance (predictive and preventive maintenance) is gaining traction as it optimizes uptime and resource use (Esteban et al., 2023; Torres-Prieto et al., 2023).

Currently, EDC repair processes often lack standardized procedures. This situation creates diagnostic difficulties, particularly when technicians encounter multiple faults simultaneously. This uncertainty complicates repair time estimation and can lead to inefficiencies, such as trial-and-error approaches. Data mining, specifically Association Rule Mining (ARM) and Sequential Pattern Mining (SPM), offer powerful tools for extracting hidden patterns from historical repair data (Shaukat et al., 2015; Gunawan et al., n.d.). ARM can identify items (faults, actions, and components) that frequently occur together (Shaukat et al., 2015), whereas SPM analyzes the order in which events occur (Pei et al., 2001). Although these techniques have been applied in various maintenance domains, including power distribution (Zhang et al., 2021), building systems (Al-Refaie & Hamdieh, 2023), and related financial hardware, such as ATMs (Rachburee et al., 2017), their specific application to optimize EDC device repair protocols is limited, representing a novelty in this study.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

This study aims to bridge this gap by applying a framework combining ARM (using the Apriori algorithm) and SPM (using the PrefixSpan algorithm) to a large dataset of historical EDC repair records (56,629 transactions) from PT. XYZ Indonesia. Furthermore, this study demonstrates the practical implementation of these specific data mining techniques, as studied in Information Technology curricula, to address real-world operational challenges within an electronic device repair center. This study sought to address the following research questions: What fault combinations frequently co-occur in EDC repair data? Can specific fault combinations predict components that require replacement? Are there common sequential patterns in the repair process of specific faults? How can these identified patterns inform operational recommendations for repair centers?

By achieving these objectives, this study provides novel insights into EDC failure characteristics and proposes a data-driven approach to enhance the efficiency and quality of repair services in this critical domain.

LITERATURE REVIEW

Association Rule Mining (ARM) is a well-established data mining technique used to discover interesting relationships or associations among items in large datasets, famously applied in market basket analysis (Shaukat et al., 2015). The goal is to find rules of the form X→Y, indicating that the presence of itemset X implies the likely presence of itemset Y, where X and Y are disjoint itemsets (X∩Y=∅) (Shaukat et al., 2015). The key algorithms include Apriori, FP-Growth, and Eclat. The strength of these rules was evaluated using metrics such as support, confidence, and lift (Shaukat et al., 2015).

- **Support:** Measures the frequency of the itemset $(X \cup Y)$ in the dataset (T) . This indicates how often a combination of items appears in all transactions.

$$\text{Support}(X \rightarrow Y) = \frac{|\{t \in T : X \cup Y \subseteq t\}|}{|T|} \quad (1)$$

  Here, t represents a transaction and T is the set of all transactions.

- **Confidence:** measures the conditional probability of finding itemset (Y) in transactions that already contain itemset (X). This indicates the reliability of the rule.

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} \quad (2)$$

- **Lift:** Indicates the strength of the association between items (X) and (Y), indicating whether they co-occur more frequently than random chance. If the lift value exceeded 1, a meaningful positive association between these items was suggested.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)} \quad (3)$$

The **Apriori algorithm** utilized in this research uses a breadth-first exploration method to generate candidate itemsets iteratively according to a property known as the "Apriori property," which implies that subsets of frequent itemsets must themselves be frequent (Ingle & Suryavanshi, 2015). Although simple, Apriori can be computationally costly because of multiple database scans and candidate generation (Ingle & Suryavanshi, 2015; Zakur & Flaih, 2023). The **FP-Growth algorithm** was proposed as an improvement, which addresses Apriori's inefficiencies by constructing a compressed data structure called an FP-tree, reducing the number of database scans required, and eliminating the explicit generation of candidate item sets (Jang et al., 2021; Rachburee et al., 2017). ARM have found applications in maintenance (Zhang et al., 2021; Bai et al., 2024; Liu et al., 2020). Zhang et al. (2021) applied ARM to power distribution terminal faults and extracted diagnostic rules. Al-Refaie et al. (2023) used the ARM to build maintenance records for forecasting tasks. Rachburee et al. (2017) applied Apriori and FP-Growth to ATM repair logs, noting FP-Growth's superior speed and ARM's ability to predict failed parts.

Sequential Pattern Mining (SPM) enhances traditional association analysis by incorporating the sequence or chronological order of events, thereby identifying subsequences that frequently appear within ordered sequence databases (Pei et al., 2001). This is crucial for analyzing processes, such as repair workflows. A sequential pattern α is considered frequent if its support exceeds a minimum threshold. The support of pattern α in a sequence database (S) is defined as the proportion of sequences in S that contain α as a subsequence:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$\text{Support}(\alpha) = \frac{|\{S_i \in S : \alpha \sqsubseteq S_i\}|}{|S|} \qquad (4)$$

Commonly utilized algorithms for sequential pattern mining include **GSP** (Srikant & Agrawal, 1996), which extends the Apriori algorithm; **SPADE** (Zaki, 2001), which leverages a vertical data format; and **PrefixSpan**(Pei et al., 2001), which uses a pattern-growth method. In this study, the **PrefixSpan algorithm** was selected for sequential pattern analysis because of its efficiency. It avoids explicitly generating candidate sequences by progressively projecting and narrowing down the database according to frequent sequence prefixes (Pei et al., 2001). This method is often more efficient than GSP, particularly for long sequences (Pei et al., 2001; Figueiredo et al., 2022). Furthermore, PrefixSpan's pattern-growth approach circumvents the extensive candidate-sequence generation required by algorithms like GSP and the complex vertical-format manipulations of SPADE. This results in reduced memory overhead and faster execution, especially for long or dense sequences (Pei et al., 2001; Figueiredo et al., 2022). Additionally, its straightforward Python implementation (Fournier-Viger, n.d.) facilitated seamless integration into our analysis pipeline.

SPM has been used in predictive maintenance, such as mining alarm sequences in mining trucks (Kotan, cited in Kahraman et al., 2021) or mobile networks (Figueiredo et al., 2022), and predicting maintenance activities (Al-Refaie & Hamdieh, 2023). Recent surveys highlight the centrality of data mining in predictive maintenance (Esteban et al., 2023; Torres-Prieto et al., 2023).

Although both ARM and SPM are valuable for maintenance data (Esteban et al., 2023), their combined application to optimize EDC repair protocols is scarce. This study leveraged both ARM (Apriori) for static associations and PrefixSpan (SPM) for dynamic workflows.

METHOD

This study employed a quantitative, exploratory data-mining approach using a case study methodology at the repair center of PT. XYZ Indonesia. The dataset comprises 56,629 historical repair records for EDC terminals. Each record contained details like device identifiers, process dates, repair status, and entries for up to five faults (e.g., "Battery Not Charging," "7FFFFF"), five actions (e.g., "REPLACE," "INJECT OS"), and five replaced components (e.g., specific batteries, mainboards).

Data pre-processing was conducted using Python and the Pandas library. This involves handling missing values by filling empty slots with "Unknown." Data cleaning included standardizing textual entries and consolidating related columns. For ARM, relevant faults, actions, and component items from a single repair case are combined into a transactional itemset. For Sequential Pattern Mining (SPM), the data were structured as sequences of repair events. Each repair ticket is transformed into an ordered series of events, with each event representing a combination of faults, actions, and components (e.g., [`'FAULT_7FFFFF|ACTION_REPLACE - CMOS|COMP_BAT10074'`, `'ACTION_INJECT - OS'`]). This structured sequence data were then input into the PrefixSpan library for analysis.

Association Rule Mining was performed using the Apriori algorithm in the mlxtend.frequent_patterns module (Raschka, n.d.). The input was one-hot-encoded data. After testing various parameter ranges (support: [0.05, 0.03, 0.01]; confidence: [0.6, 0.7, 0.8]; lift: [1.2, 1.5, 2.0]), the final analysis used optimal parameters of Support: 0.05 (5%), Confidence: 0.6 (60%), and Lift: 1.2 to ensure meaningful and robust rules. Frequent itemsets were generated, followed by rule generation and filtering.

Sequential Pattern Mining utilizes the PrefixSpan algorithm implemented via the Python prefixspan package (Fournier-Viger, n.d.), based on Pei et al. (2001). The input consisted of ordered sequences of event triplets. A balanced minimum support threshold of 1698 sequences (approximately 3%) was initially determined, guided by analyses such as the Elbow method shown in Figure 3. However, further analysis explored lower thresholds to identify longer patterns. A maximum pattern length of five was applied.

Results from both methods were analyzed and interpreted, focusing on frequent itemsets, significant association rules (support, confidence, lift), and common sequential patterns (support).

RESULT

The application of ARM and SPM yielded significant patterns related to EDC failures and repairs, following the research flow.

Association Rule Mining (ARM) Findings. Before detailing the specific findings, it is useful to conceptualize the data transformation and output for ARM. The input data consist of transactions, where each transaction represents a single repair case and contains a set of items (faults, actions, and components) associated with that case. For example, one transaction might be {`'FAULT_7FFFFF'`, `'ACTION_REPLACE - CMOS'`, `'COMP_BAT10074'`, `'ACTION_INJECT - OS'`}, while another might be {`'FAULT_Battery Not`

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Charging', 'ACTION_REPLACE - Battery Pack'}. The ARM algorithm processes this transaction collection. The output, after applying ARM, consists of association rules, such as IF {Antecedent Items} THEN {Consequent Items}, along with metrics quantifying the strength of the rule. For example, a rule might state IF {'FAULT_Battery Not Charging'} THEN {'ACTION_REPLACE - Battery Pack'} with high confidence (e.g., 95%) and lift (e.g., 6.0), indicating a strong predictive relationship discovered from the input transactions.

The ARM analysis using Apriori with parameters of 5% support, 60% confidence, and 1.2 lift, successfully identified numerous associations. Specifically, 16 event rules (combinations of fault, action, and component), 24 fault-only rules, and 12 component-only rules are generated. The impact of varying the support and confidence thresholds on the number of component association rules (with lift fixed at 1.2) is illustrated in Figure 1. As expected, lower support and confidence thresholds yield more rules, whereas higher thresholds result in fewer, potentially stronger rules. The chosen parameters (Support=0.05, Confidence=0.6) represent a balance, yielding a manageable number of rules for analysis.

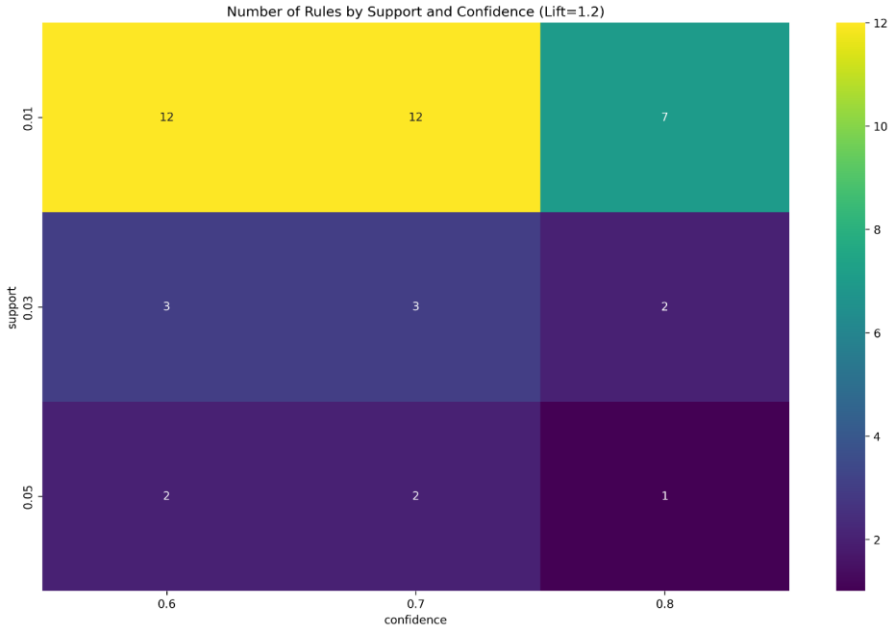

Figure 1. Number of component association rules generated by varying support and confidence thresholds (Lift ≥ 1.2).

The event rule analysis yielded particularly strong associations, with an average lift of 16.45 and a maximum lift reaching 39.80, indicating highly correlated event occurrences. Among the most insightful rules, those linking specific faults to subsequent actions and component replacements were prominent. For instance, the error code "7FFFFF" was identified as the most frequent single fault, present in approximately 50.1% of repairs. Investigating co-occurring faults, the combination of error code "2000000" and the need for "L4 Repair" was notable, occurring in about 12.8% of cases. This suggests a strong link between this error code and more severe repair interventions. Extending this, the triplet involving "Battery Not Charging", "2000000", and "L4 Repair" was the most dominant three-fault combination, found in approximately 7.8% of transactions, highlighting a common complex failure scenario. One notable high-lift event rule (Lift: 39.80, Support: 1.0%, Confidence: 86.7%) involves the fault code '2000000' (tamper indication). Initially, this fault often requires replacing the battery pack (component code 1070020063). Subsequently, this frequently leads to a 'Battery Not Charging' fault, requiring CMOS battery replacement (component code 1070020014). Additionally, it usually coincides with a higher-level repair action known as 'L4 Repair,' which involves detampering procedures and ROM/K21 upgrade. This illustrates a complex interplay where a tamper-related battery replacement might precede or co-occur with other power system failures and advanced repair needs.

Regarding components, the component identified as COMP_BAT10074 was the most frequently replaced item (≈48.3% support). Analyzing pairs, the components COMP_1070020014 and COMP_1070020063 were most often replaced together (≈10.0% support). The fault rule analysis, with an average lift of 3.37, showed strong relationships between faults, as visualized in Figure 2. For example, the co-occurrence of "FAULT_L4 Repair" and "FAULT_Battery Not Charging" strongly predicted "FAULT_2000000" (Support: 8.7%,

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Confidence: 76.3%, Lift: 3.92). Similarly, the component rule analysis (average lift 3.71) revealed that if "COMP_BAT10074" (CMOS battery) and "COMP_PCA268-300-02-A" (Mainboard) were replaced, it strongly implied the replacement of "COMP_SUB268-401-01-A" (Sub Assy/Dome sheet) (Support: 1.5%, Confidence: 73.5%, Lift: 6.25).

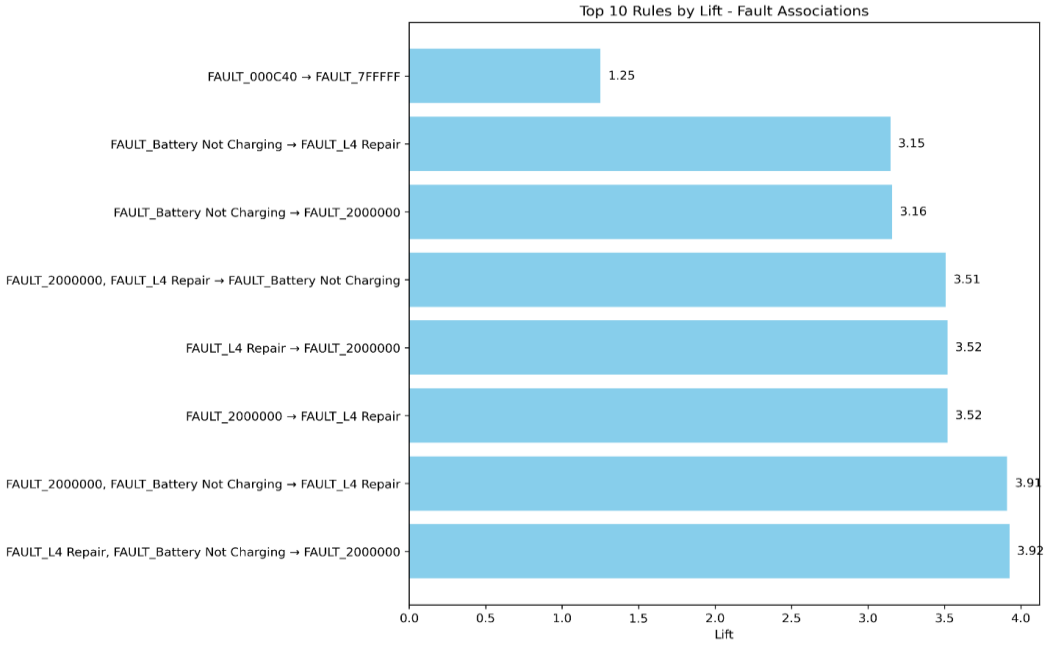

Figure 2. Top 10 fault association rules ranked by lift.

Further strong fault-component rules quantified common repair logic. The rule indicating that a "FAULT_Battery Not Charging" fault leads to a "ACTION_REPLACE - Battery Pack" action showed high confidence (≈95%) and lift (≈6.0). The presence of error code "FAULT_2000000" (indicating tamper) strongly implied the necessity of both "ACTION_Detampering" procedures and "ACTION_REPLACE - CMOS Battery" actions (Confidence ≈96%, Lift ≈4.9). Other highly confident rules included "FAULT_000C40" leading to "ACTION_REPLACE - DOMESHEET" (Confidence 100%) and "FAULT_7FFFFF" leading to both "ACTION_REPLACE - CMOS" and "ACTION_INJECT - OS" (Confidence ≈99%). These rules provide quantitative support for common repair practices and component dependencies.

Sequential Pattern Mining (SPM) Findings. Similar to ARM, understanding the data format is key to SPM. The input for PrefixSpan was a database of sequences. Each sequence represents one repair case and consists of an ordered list of "events," where each event is typically a combination of faults, actions, and components occurring at a specific step. For example, consider a repair case recorded in the original dataset, with

- Fault1: FAULT_7FFFFF, Action1: REPLACE - CMOS, Component1: COMP_BAT10074
- Fault2: (empty), Action2: INJECT - OS, Component2: (empty)

  This is transformed into a single input sequence for SPM: ['(FAULT_7FFFFF | ACTION_REPLACE - CMOS | COMP_BAT10074)', '(ACTION_INJECT - OS)'].

  Another example involves three steps:
- Fault1: FAULT_Battery Not Charging, Action1: REPLACE - Battery Pack, Component1: COMP_1070020014
- Fault2: FAULT_2000000, Action2: REPLACE - CMOS, Component2: COMP_1070020063
- Fault3: FAULT_L4 Repair, Action3: detampering - Upgrade ROM/K21, Component3: (empty)

  This transforms into the sequence: ['(FAULT_Battery Not Charging | ACTION_REPLACE - Battery Pack | COMP_1070020014)', '(FAULT_2000000 | ACTION_REPLACE - CMOS | COMP_1070020063)', '(FAULT_L4 Repair | ACTION_detampering - Upgrade ROM/K21)'].

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

This transformation into sequences is necessary because SPM algorithms, such as PrefixSpan, are designed to operate on ordered lists of events to discover temporal patterns (i.e., which event follows another). The original dataset structure, with separate columns for each potential fault/action/component (Fault1, Action1, Fault2, Action2, etc.), does not inherently represent this temporal order within a single repair case, in a format suitable for standard SPM algorithms. Flattening the data into a sequence of events allows the algorithm to analyze the progression of steps within each repair.

The PrefixSpan algorithm processed the entire dataset consisting of 56,629 repair sequences. Each sequence represented a series of repair steps. Sequences ranged from 1 to 5 steps in length, averaging 2.49 steps, with most sequences having two steps. A total of 4,686 unique event types (fault-action-component combinations) were identified across all sequences. The support for individual events varied greatly, with a median support count of only two, indicating that many events were rare, while the maximum support reached 24,934 for the most common event. The output consists of frequent sequential patterns, which are subsequences found in many input sequences above the minimum support threshold. For example, a frequent pattern discovered might be <(`FAULT_7FFFFF` | `ACTION_REPLACE - CMOS` | `COMP_BAT10074`), (`ACTION_INJECT - OS`)>, indicating that the event of replacing the CMOS with error 7FFFFF is frequently followed by the event of injecting the OS.

The selection of an appropriate minimum support threshold was guided by analyzing the relationship between the threshold and the proportion of unique events covered, as shown in Figure 3. The "elbow point" suggests a statistically optimal threshold around a support count of 10, but this was deemed too low for practical significance.

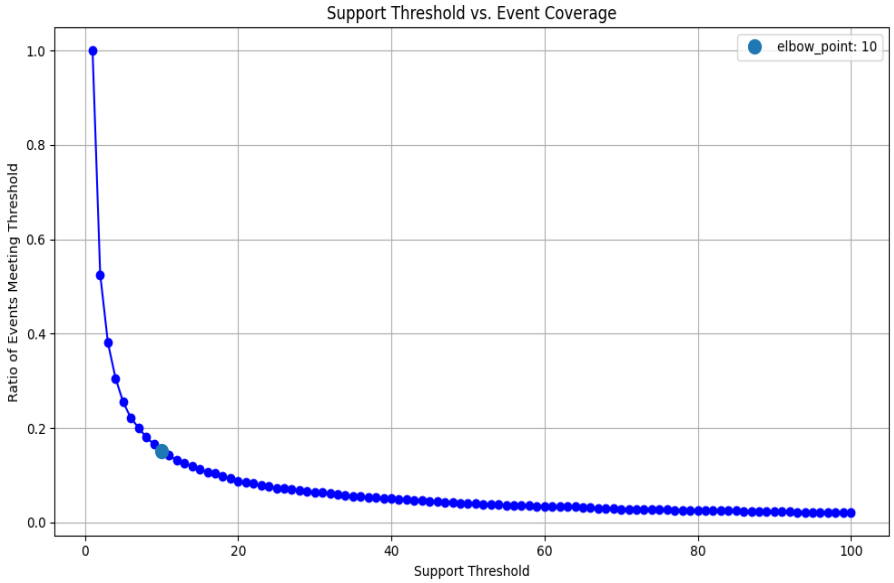

Figure 3. Support Threshold vs. Event Coverage Ratio, indicating an Elbow Point.

SPM results using PrefixSpan initially identified 17 frequent patterns at a 3% support threshold (1698 sequences), comprising 12 length-1 and 5 length-2 patterns. The top 10 patterns by support, encompassing both single events and two-step sequences, are summarized in Table 1. The most frequent single-event pattern was the action "(`ACTION_INJECT - OS`)", occurring in approximately 44.0% of sequences. The most frequent two-step sequence, found in about 24.2% of cases, was "(`FAULT_7FFFFF` | `ACTION_REPLACE - CMOS` | `COMP_BAT10074 - Battery`) → (`ACTION_INJECT - OS`)", confirming a standard workflow for the common "7FFFFF" error.

To explore more complex workflows, the support threshold was lowered iteratively. At a support threshold of 89 sequences (0.16%), 26 patterns of length 3 were discovered. The most frequent length-3 pattern (Support: 1498 sequences, 2.65%) represented a complex battery and tamper issue: (`FAULT_Battery Not Charging` | `ACTION_REPLACE - Battery Pack` | `COMP_1070020014`) → (`FAULT_2000000` | `ACTION_REPLACE - CMOS` | `COMP_1070020063`) → (`FAULT_L4 Repair` | `ACTION_detampering - Upgrade ROM/K21`). This sequence suggests an initial battery pack replacement, followed by a CMOS replacement related to a tamper code, and finally requiring an L4 detampering procedure. Other length-3 patterns involved different combinations of battery issues, tamper codes, CMOS replacements, and detampering actions, often culminating in an L4 repair step.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Further lowering the support threshold to 12 sequences (0.02%) revealed 35 patterns of length 4. The most frequent length-4 pattern (Support: 118 sequences, 0.21%) extended the top length-3 pattern by including an LCD replacement step: (FAULT_Battery Not Charging | ...) → (FAULT_2000000 | ...) → (FAULT_3000020 | ACTION_REPLACE - LCD | ...) → (FAULT_L4 Repair | ...). Other length-4 patterns involved sequences like handling a "NO Power" fault followed by dome sheet replacement, CMOS replacement, and finally OS injection. Patterns of length 5 were extremely rare, not meeting even very low support thresholds (e.g., none found above 0.02% support). This indicates that while complex multi-step repairs involving 3 or 4 distinct stages occur, they represent a small fraction of the total repairs, with the vast majority following shorter, more common sequences.

Table 1. Top 10 Sequential Patterns by Support (min_support ≈ 3%)

Rank	Pattern	Support Count	Support (%)	Length
1	(ACTION_INJECT - OS)	24934	44.03%	1
2	(FAULT_7FFFFF | ACTION_REPLACE - CMOS | COMP_BAT10074 - ...)	17976	31.74%	1
3	(FAULT_7FFFFF | ACTION_REPLACE - CMOS | COMP_BAT10074 - ...) → (ACTION_INJECT - OS)	13728	24.24%	2
4	(FAULT_L4 Repair | ACTION_detampering - Upgrade ROM/K21)	9678	17.09%	1
5	(ACTION_detampering - Upgrade ROM/K21)	6184	10.92%	1
6	(FAULT_App Error | ACTION_REASSEMBLY - MainBoard)	5526	9.76%	1
7	(FAULT_7FFFFF | ACTION_INJECT - OS)	4575	8.08%	1
8	(ACTION_INJECT - SOFTWARE)	4427	7.82%	1
9	(FAULT_2000000 | ACTION_REPLACE - CMOS | COMP_1070020014 - ...)	3223	5.69%	1
10	(FAULT_App Error | ACTION_REASSEMBLY - MainBoard) → (ACTION_INJECT - OS)	3117	5.50%	2

(Note: Component names abbreviated for brevity in table)

DISCUSSIONS

This section provides an explanation and an in-depth analysis of the results, compares them with those of other studies, and discusses the limitations.

Explanation and Analysis of Results. The combined ARM and SPM approaches provide complementary insights. The ARM identifies static relationships—items that frequently occur together. In contrast, SPM revealed dynamic patterns by identifying the specific order of the repair events. The consistency between the methods, such as the link between error "2000000" (tamper) and subsequent detampering actions found by both ARM and SPM, validates the findings. The ARM results, particularly the event rules combining faults, actions, and components, yielded the strongest associations (average lift of 16.45), underscoring the significant interdependencies between these elements, and suggesting a focus on these combined patterns for prediction and optimization.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The high frequency of specific faults ("7FFFFF") and actions ("INJECT OS") points towards common failure modes and potentially standardized (even if undocumented) recovery procedures. Battery-related issues clearly emerged as a primary concern, reflected in both fault frequency and component replacement rates. The strong associations discovered by the ARM, particularly those with high confidence and lift, provide empirical evidence for potential diagnostic shortcuts and predictive component needs. For example, the near certainty of needing a battery pack replacement when "Battery Not Charging" occurs (95% confidence) justifies immediate action. The identification of component clusters (e.g., the group involving mainboard modules like "`COMP_1070020014`", "`COMP_1070020063`", "`COMP_1160030013`") suggests systemic failures where multiple related parts fail concurrently, guiding more holistic repair approaches.

The SPM results highlight the standard workflows. The frequent sequence of replacing the CMOS battery followed by OS injection for "7FFFFF" errors likely represents an effective and commonly adopted procedure. The dominance of short patterns (1-2 steps) indicates that most repairs follow relatively simple, common pathways. The discovery of length-3 and length-4 patterns, albeit at lower support levels, reveals the structure of more complex repairs, often involving cascading issues related to power (battery pack and CMOS) and security (tamper codes, detampering, and L4 repair). The rarity of the length-5 patterns suggests a practical upper limit for common complex repair sequences in this context. These identified sequences can be translated into the proposed Standard Operating Procedures (SOPs), as shown in Figure 4, providing technicians with clear, data-backed workflows for common and complex repair scenarios.

Comparison with Other Studies. this study align with the broader literature on using data mining for maintenance. The effectiveness of ARM in identifying fault-component relationships is consistent with studies in other domains, such as power distribution (Zhang et al., 2021) and ATM repair (Rachburee et al., 2017). Similarly, the use of SPM to uncover repair workflows echoes the findings in mobile network maintenance (Figueiredo et al., 2022) and building systems (Al-Refaie & Hamdieh, 2023).

The novelty of this study lies in applying this combined framework specifically to the EDC repair domain. While Rachburee et al. (2017) analyzed ATM repairs, the specific fault codes (like tamper-related "2000000") and components identified here are unique to EDC devices. The PrefixSpan algorithm aligns with modern SPM practices noted by Figueiredo et al. (2022). Compared to Apriori, alternative ARM algorithms, such as FP-Growth, might offer efficiency gains on larger datasets, as observed by Rachburee et al. (2017) and Jang et al. (2021).

Limitations of the Study. several limitations. First, the analysis relies on the quality and consistency of the historical repair logs. Inconsistent terminology or the use of generic error codes (like "7FFFFF") could potentially introduce noise or mask specific underlying issues. Second, the study was retrospective; patterns identified reflect past behavior and might change with new device models, software updates, or evolving repair practices. Third, the choice of parameters (e.g., minimum support thresholds) influences the results. While chosen scientifically (e.g., using the Elbow method shown in Figure 3), the thresholds might exclude some less frequent but potentially interesting patterns, particularly for longer sequences that require very low support thresholds (e.g., 0.16% for length 3 and 0.02% for length 4). Finally, interpreting extremely long or complex patterns is challenging.

CONCLUSION

This study successfully applied a data mining framework combining Association Rule Mining (Apriori) and Sequential Pattern Mining (PrefixSpan) to analyze a substantial dataset of EDC terminal repair records. This study identified significant recurring patterns related to device failure, component replacement, and repair procedures.

Key findings include the identification of dominant faults (e.g., "7FFFFF", "Battery Not Charging", "2000000"), frequently replaced components (e.g., CMOS Battery, mainboard modules), strong associations between specific faults and required actions/components (e.g., tamper errors necessitating detampering and battery replacement), and common sequential repair workflows, including both short (1-2 step) and more complex (3-4 step) patterns for specific scenarios.

The results provide valuable data-driven insights that can be translated into practical improvements for repair centers. Specific recommendations include using identified strong rules (especially event rules with high lift) to predict likely failure combinations, focusing on fault-component associations to optimize inventory management, and monitoring high-confidence event sequences to improve predictive maintenance capabilities. Developing standardized protocols based on the discovered patterns (as exemplified in Figure 4) and enhancing technician training are also key outputs. Implementing these data-driven insights can significantly enhance the operational efficiency. Specifically, these improvements can shorten the repair times and increase the reliability of EDC terminal services. This study underscores the value of applying data mining techniques to service operation data to uncover actionable knowledge and driving process improvements. Future work could explore alternative algorithms, such as FP-Growth or GSP, test lower support thresholds to find niche patterns, incorporate domain knowledge further, investigate classification models, and validate the proposed protocols in a field study.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 2, April 2025
DOI : https://doi.org/10.33395/sinkron.v9i2.14759
e-ISSN : 2541-2019
p-ISSN : 2541-044X

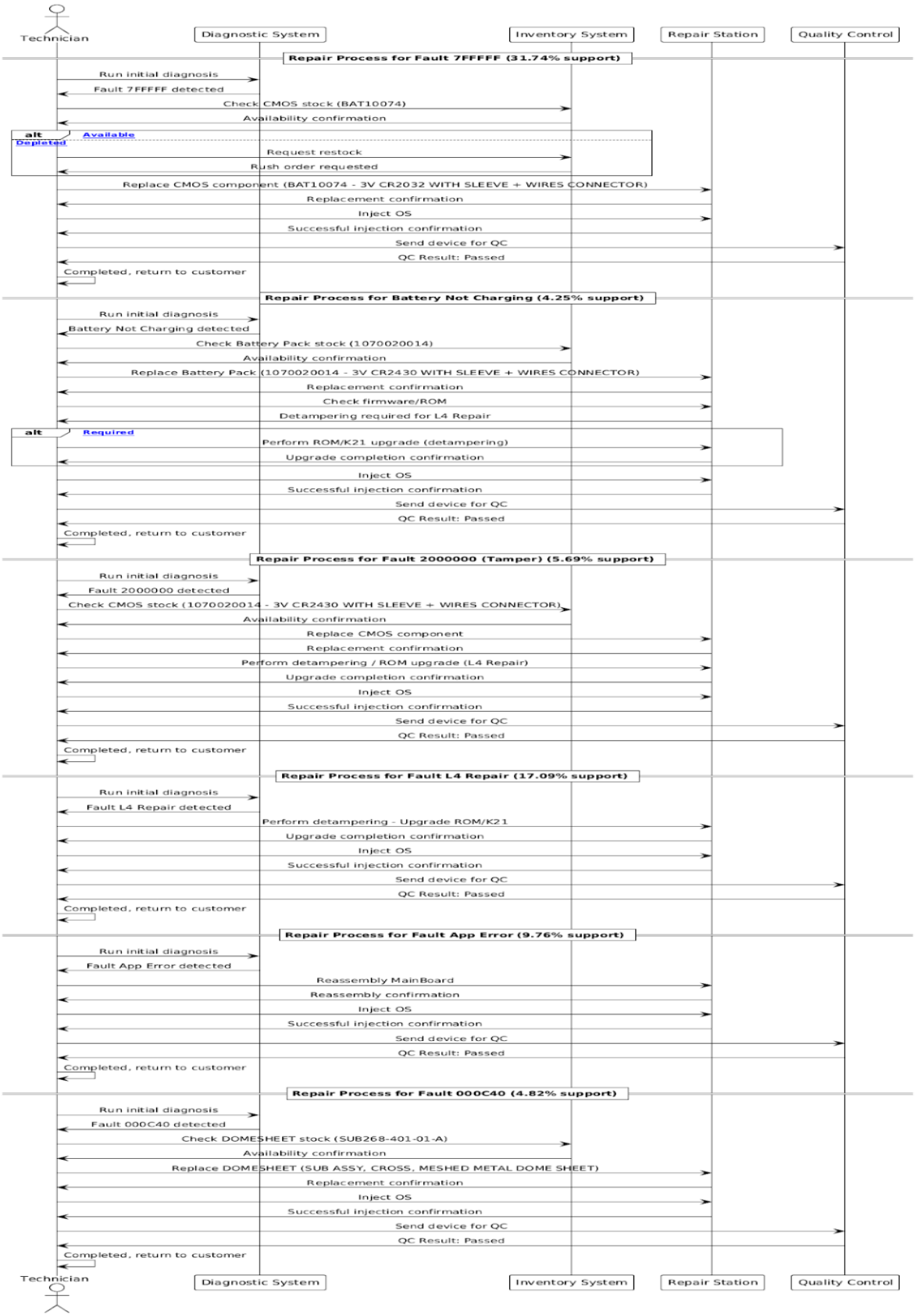

Figure 4. Proposed Standard Operating Procedures (SOPs) based on Frequent Repair Sequences.

*name of corresponding author

REFERENCES

Aggarwal, H., Kumar, V., & Arora, H. D. (2023). Data mining algorithm based on Renyi fuzzy association rule: An application for the selection of suitable course. *Research in Statistics, 1(1), 2271902*. https://doi.org/10.1080/27684520.2023.2271902

Al-Refaie, A., & Hamdieh, M. (2023). A data mining framework to predict the sequential pattern of maintenance activities and spare parts. *Journal of Quality in Maintenance Engineering, 29*(3), 697-718. https://doi.org/10.1108/JQME-06-2022-0056

Bai, Y., Li, H., Wang, W., Liu, S., Zhang, N., & Zhang, C. (2024). Optimization algorithm of Association Rule Mining for heavy-haul railway freight train Fault data based on distributed parallel computing. *Science Progress, 107*(4), 00368504241301181. https://doi.org/10.1177/00368504241301181

Esteban, A., Zafra, A., & Ventura, S. (2023). Data mining in predictive maintenance systems: A taxonomy and systematic review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13*(1), e1471. https://doi.org/10.1002/widm.1471

Figueiredo, L. E. N., Caetano, I., & Almeida, V. (2022). A fault management preventive maintenance approach in mobile networks using sequential pattern mining. In *Proceedings of the 19th International Conference on Wireless Networks and Mobile Systems (WINSYS)* (pp. 77–85). SCITEPRESS – Science and Technology Publications. https://doi.org/10.5220/0011371900003281

Fournier-Viger, P. (n.d.). *The PrefixSpan Algorithm Documentation*. Retrieved May 6, 2025, from https://www.philippe-fournier-viger.com/spmf/PrefixSpan.php

Gunawan, D., ., S., & ., S. (n.d.). Perbandingan Algoritma GSP, PrefixSpan Dan SPADE Dalam Menemukan Pola Runtun Pada Keranjang Belanja Konsumen. *Jurnal Ilmiah MATRIK, 23*(1), 89-96. https://doi.org/10.30812/matrik.v23i1.1651 (Year needed for APA 7th)

Ingle, M. G., & Suryavanshi, N. Y. (2015). Association Rule Mining using Improved Apriori Algorithm. *International Journal of Computer Applications, 112*(4), 37–42. https://doi.org/10.5120/19658-1297

Jang, H., Yang, Y., Park, J. S., & Kim, B. (2021). FP-Growth algorithm for discovering region-based association rule in the IoT environment. *Electronics, 10*(24), 3091. https://doi.org/10.3390/electronics10243091

Kahraman, C., Onar, S. C., Oztaysi, B., Sari, I. U., Al-Refaie, A., & Kob K., B. (Eds.). (2021). *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation: Proceedings of the INFUS 2021 Conference*, held August 24-26, 2021. Volume 1 (Vol. 307). Springer Nature. https://doi.org/10.1007/978-3-030-85577-2 (Specific chapter by Kotan needed)

Khasanah, A. U., Erlangga, D. A., & Jamil, A. M. (2018). An application of Data mining techniques in designing catalogue for a laundry service. *MATEC Web of Conferences, 154, 01099*. https://doi.org/10.1051/matecconf/201815401099

Liu, J., Shi, D., Li, G., Xie, Y., Li, K., Liu, B., & Ru, Z. (2020). Data-driven and Association Rule Mining-based Fault diagnosis and Action mechanism analysis for building chillers. *Energy and Buildings, 216*, 109957. https://doi.org/10.1016/j.enbuild.2020.109957

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering (ICDE 2001)* (pp. 215–224). IEEE Computer Society. https://doi.org/10.1109/ICDE.2001.914830

Pereira, E., Silva, L. C., Silva, D., & Souza, J. (2022). Analyzing alarm logs for failure prediction in cellular networks via sequential pattern mining: A case study in a Tier-1 mobile operator. *Computer Networks, 217*, 109315. https://doi.org/10.1016/j.comnet.2022.109315

Rachburee, N., Arunrerk, J., & Punlumjeak, W. (2017). Failure part mining using association rules mining by FP-Growth and Apriori algorithms: Case of ATM maintenance in Thailand. In *Information Technology and Communications Security* (LNCS 449, pp. 19–26). Springer. https://doi.org/10.1007/978-3-319-67131-1_3

Raschka, S. (n.d.). *MLxtend: Apriori Module Documentation.* Retrieved May 6, 2025, from http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/

*name of corresponding author

Shaukat, K., Zaheer, S., & Nawaz, I. (2015). Association Rule Mining: An Application Perspective. *International Journal of Computer Science and Innovation*, *2015*(1), 29–38.

Torres-Prieto, R., Martínez-Puerto, P. A., Ortiz, J., & López-Puente, J. (2023). A survey on data mining for data-driven industrial assets maintenance. *Technologies*, *11*(2), 67. https://doi.org/10.3390/technologies11020067

Valtierra-Rodriguez, M., Contreras-Valdez, F., Granados-Lieberman, D., & García-Hernández, G. (2020). Predictive data mining techniques for fault diagnosis of electric equipment: A review. *Applied Sciences*, *10*(3), 950. https://doi.org/10.3390/app10030950

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, *42*(1-2), 31–60. https://doi.org/10.1023/A:1007652502315

Zhang, X., Tang, Y., Liu, Q., Liu, G., Ning, X., & Chen, J. (2021). A Fault Analysis Method Based on Association Rule Mining for Distribution Terminal Unit. *Applied Sciences*, *11*(11), 5221. https://doi.org/10.3390/app11115221

*name of corresponding author