# White Paper

AI-Powered Dynamic Business Rule Discovery Using Transaction Analytics

## Abstract

In high-volume transactional systems, dynamically identifying business transaction rules is crucial to ensuring system stability and performance. This white paper introduces an AI-driven approach to derive transaction rules by analyzing relationships between independent factors (e.g., transaction count, batch job volume) and dependent factors (e.g., error count, failure rate, response time degradation). By leveraging in-house Gen AI, the solution generates natural language recommendations based on detected patterns, improving clarity and actionability.

## 1. Introduction

Traditional transaction rules rely on predefined static thresholds, which fail to adapt to dynamic system behaviors. Our AI-powered approach extracts rules by analyzing historical data and identifying transaction limits where performance degradation occurs. The solution ensures real-time adaptability across multiple domains, such as web applications, batch jobs, and API processing.

## 2. Approach: Dynamic Rule Extraction Using AI & Analytics

### 2.1 Key Concept: Rule Factor vs. Validation Factor

- \*\*Rule Factor (Independent Variable)\*\*: Defines expected behavior (e.g., transaction count, request rate, batch size).  
- \*\*Validation Factor (Dependent Variable)\*\*: Measures system impact (e.g., error count, failure rate, response time).  
- \*\*AI-Powered Rule Discovery\*\*: Automatically detects correlations and generates human-readable recommendations.

## 3. Example Scenarios

### 3.1 Web Application Transaction Limits

- \*\*Independent Variable\*\*: Transaction count (requests per second).  
- \*\*Dependent Variable\*\*: Error count.  
- \*\*Dynamic Rule Extraction\*\*:  
 - \*\*100 requests/sec\*\* → \*\*0% errors\*\* (Stable).  
 - \*\*150 requests/sec\*\* → \*\*3% errors\*\* (Warning).  
 - \*\*200 requests/sec\*\* → \*\*10% errors\*\* (Critical).  
- \*\*AI-Generated Recommendation:\*\*  
 - \*'We've observed that at 200 requests/sec, error rates increase significantly. To maintain system stability, we recommend a limit of 150 requests/sec.'\*

### 3.2 Batch Job Processing

- \*\*Independent Variable\*\*: Number of jobs per batch.  
- \*\*Dependent Variable\*\*: Job failures per batch.  
- \*\*Dynamic Rule Extraction\*\*:  
 - \*\*50 jobs\*\* → \*\*0 failures\*\*.  
 - \*\*75 jobs\*\* → \*\*2 failures\*\* (Marginal Impact).  
 - \*\*100 jobs\*\* → \*\*10 failures\*\* (High Risk).  
- \*\*AI-Generated Recommendation:\*\*  
 - \*'Batch job execution beyond 75 jobs has led to increased failure rates. To maintain reliability, we recommend keeping the batch size at 75 or lower.'\*

## 4. Implementation Plan

### 4.1 Data Collection from Observability APIs

- Fetch transaction \*\*volume & error count\*\* from \*\*AppDynamics, Dynatrace, or other observability tools\*\*.  
- Store \*\*time-series structured data\*\*.

### 4.2 Pattern Analysis & Threshold Detection

- \*\*Statistical Models\*\*: Identify safe transaction limits before failures increase.  
- \*\*Anomaly Detection\*\*: Use \*\*percentile analysis\*\* to flag risky behavior.  
- \*\*Regression Analysis\*\*: Model how transaction count impacts failures.

### 4.3 AI-Powered Recommendation Engine

- AI converts findings into human-readable recommendations.  
- Uses in-house Gen AI to explain trends and suggest proactive measures.

## 5. Why This Will Work

- \*\*Automated, Data-Driven Rule Discovery\*\* – Eliminates reliance on static thresholds.  
- \*\*AI-Powered Natural Language Recommendations\*\* – Enhances clarity for engineers.  
- \*\*Scalability\*\* – Works across web apps, batch jobs, and high-volume systems.  
- \*\*No External API Dependency\*\* – Uses in-house Gen AI for recommendations.  
- \*\*Adaptive & Continuous Learning\*\* – Rules adjust dynamically as transaction behavior evolves.

## 6. Example Banking Use Cases

- \*\*Online Payment Transactions\*\*  
 - \*\*Independent Variable\*\*: Payment requests per second.  
 - \*\*Dependent Variable\*\*: Payment failures or processing time.  
 - \*\*Business Rule Example\*\*: \*'If payments exceed 500 per second, failure rates increase by 5%, suggesting a transaction cap of 450/sec.'\*  
  
- \*\*ATM Cash Withdrawals\*\*  
 - \*\*Independent Variable\*\*: Withdrawals per ATM per hour.  
 - \*\*Dependent Variable\*\*: Machine errors due to load.  
 - \*\*Business Rule Example\*\*: \*'If ATM transactions exceed 100 per hour, machine timeout errors rise, suggesting a cap of 90/hr.'\*  
  
- \*\*Fraud Detection for Transactions\*\*  
 - \*\*Independent Variable\*\*: Unusual transaction volume per account.  
 - \*\*Dependent Variable\*\*: Fraud risk score.  
 - \*\*Business Rule Example\*\*: \*'If a user performs 10+ high-value transactions within an hour, flag the account for review.'\*

## 7. Conclusion

This AI-powered approach dynamically derives transaction rules using real-time data, eliminating static configurations. By integrating Gen AI for natural language rule generation, businesses can ensure system stability while adapting to dynamic workloads.