A Feature Model-Based Configurable Framework for Reproducible Intervention Evaluation

Carlos Murillo-Barrera[1], Miguel Vera[2], Nicolás Márquez[3]*, Cristian Vidal-Silva[4]*
Facultad de Ciencias Sociales, Educación Comercial y, Derecho, Universidad Estatal de Milagro (UNEMI), Milagro, Ecuador[1]
Facultad de Ingeniería, Negocios y, Ciencias Agroambientales, Universidad Viña del Mar, Viña del Mar, Chile[2]
Escuela de Auditoría e Ingeniería en Control de Gestión, Universidad Autónoma de Chile, Talca, Chile[3]
Facultad de Ingeniería y Negocios, Universidad de Las Américas, Santiago, Chile[4]

***Abstract*—Intervention evaluation frequently relies on fixed analytical workflows that provide limited support for methodological adaptability, reproducibility, and systematic management of analytical variability. This study proposes a Feature Model-based configurable framework that represents data preparation, feature construction, statistical inference, robustness validation, and decision-support generation as interdependent analytical features. Mandatory, optional, alternative, and dependency-constrained decisions are modeled explicitly, enabling the systematic generation, validation, and diagnosis of reproducible analytical configurations. The framework was instantiated in a managerial capability development intervention involving micro-enterprises. Across the strategic, financial, and operational capability dimensions, the generated pipelines identified statistically significant improvements with Cohen's d values ranging from 0.48 to 0.65. The direction and inferential interpretation of the intervention effects remained stable across the admissible preprocessing, imputation, and testing configurations evaluated in the case study. Because the empirical assessment uses a single dataset from one application domain, the findings establish technical feasibility and within-case robustness rather than cross-domain generalizability. The study contributes a variability-aware analytical architecture and an open-source Python-based engine integrating Feature Model reasoning, configuration validation, and Parallel FastDiag-based conflict diagnosis, thereby supporting transparent, inspectable, and adaptable intervention-evaluation workflows.**

***Keywords*—Feature models; variability management; configurable systems; reproducible analytics; decision support systems; intervention evaluation**

I. Introduction

Intervention evaluation has become an essential activity in domains such as education, healthcare, organizational development, public policy, and sustainability management, where evidence-based decision making increasingly depends on the systematic analysis of complex and heterogeneous datasets [1]–[3]. As organizations generate larger volumes of information and rely more heavily on analytical evidence to support strategic and operational decisions, evaluation processes have evolved from isolated statistical procedures toward integrated computational environments that combine data preparation, analytical reasoning, validation, and decision support [4], [5].

Recent advances in data analytics, machine learning pipelines, and scientific workflow technologies have significantly improved the automation and reproducibility of analytical processes [6]–[9]. Modern analytical environments increasingly rely on structured pipelines capable of organizing data acquisition, preprocessing, feature engineering, statistical inference, visualization, and reporting into repeatable execution stages. Such approaches contribute to transparency, traceability, and reproducibility while reducing manual effort and facilitating the replication of empirical studies [10]–[12]. Nevertheless, many existing pipeline-based solutions remain centered on predefined analytical workflows and provide limited support for representing alternative methodological strategies or systematically managing dependencies among analytical decisions [13].

This limitation is particularly relevant in intervention evaluation scenarios. Analysts frequently face alternative decisions regarding data preprocessing strategies, missing-value treatment, feature construction procedures, statistical testing methods, robustness assessment techniques, and reporting mechanisms. These methodological choices can significantly influence analytical outcomes and subsequent recommendations for decision makers [14]–[16]. In practice, many of these decisions remain embedded within scripts, spreadsheets, or procedural artifacts rather than being represented explicitly as configurable elements. Consequently, analytical reproducibility, methodological traceability, and systematic comparison of alternative evaluation strategies become difficult to achieve [13], [17].

Research in configurable systems engineering provides promising mechanisms for addressing these challenges. Variability management has been extensively adopted in software-intensive systems to represent optional functionalities, alternative configurations, dependency constraints, and adaptation mechanisms [18]–[20]. Among the available variability-modeling approaches, Feature Models have emerged as one of the most widely used mechanisms for representing and reasoning about configuration spaces through mandatory, optional, alternative, and dependency-aware features [18], [20]–[22]. Their successful application has been reported in software product lines, adaptive systems, configurable services, testing environments, and machine learning ecosystems [23]–[26].

Feature Models provide not only a compact representation of configuration alternatives but also a formal basis for validating consistency constraints and automatically generating valid configurations. Recent studies have demonstrated their effectiveness for supporting automated reasoning, configuration validation, performance assessment, and variability-aware system design [24], [27], [28]. Furthermore, advances in automated analysis techniques have considerably expanded the practical applicability of Feature Models in complex decision-support scenarios

*Corresponding author.

[18], [20]. Despite these developments, their application within analytical evaluation environments remains limited.

Existing research primarily focuses on software product lines, configurable platforms, adaptive systems, testing strategies, and machine learning configuration problems [19], [23], [24]. Comparatively little attention has been devoted to representing methodological variability within intervention evaluation processes. As a result, alternative analytical strategies are rarely modeled explicitly, analytical dependencies are seldom validated automatically, and reproducible generation of alternative evaluation configurations remains largely unsupported.

To address this research gap, this study proposes a Feature Model-based configurable framework for reproducible intervention evaluation. The proposed approach conceptualizes evaluation as a configurable analytical system composed of interconnected components responsible for data preparation, feature construction, statistical inference, robustness validation, and decision-support generation. Analytical variability is represented explicitly through Feature Models, enabling systematic generation and validation of alternative evaluation configurations while preserving methodological consistency. The proposed architecture integrates principles from configurable systems engineering, variability management, and reproducible analytical pipelines into a unified evaluation framework [12], [18], [20], [23].

The framework is evaluated through a real intervention scenario focused on managerial capability development among micro-enterprises. Different analytical configurations are instantiated and executed under alternative methodological assumptions to examine reproducibility, robustness, and consistency within the selected case. The obtained results show that the proposed framework can generate valid analytical pipelines, support reproducible execution, and maintain stable conclusions across the analytical configurations considered. Cross-domain generalizability is not claimed from this single-case validation and requires independent replication in additional settings.

In addition, a Python-based analysis engine was developed to operationalize the proposed framework. The engine implements Feature Model reasoning services, analytical configuration validation, and conflict diagnosis through an integration with a Parallel FastDiag-based diagnosis mechanism [29], [30]. This implementation provides a reproducible software infrastructure that can be reused for variability- aware analytical evaluation scenarios.

The main contributions of this work are summarized as follows:

- A configurable analytical architecture that reformulates intervention evaluation as a variability-aware systems-engineering problem.
- A Feature Model-based mechanism for representing analytical variability, methodological dependencies, and valid evaluation configurations.
- An empirical validation demonstrating the applicability of the proposed framework for reproducible and robust intervention assessment.
- An open-source software prototype that provides reproducible infrastructure for Feature Model reasoning, configuration validation, and conflict diagnosis in variability-aware analytical environments.

The remainder of this study is organized as follows. Section II reviews the literature on reproducible analytics, configurable systems, and Feature Model-based variability management. Section III presents the proposed configurable evaluation framework, including its architecture, analytical pipeline, and variability representation mechanisms. Section IV describes the experimental validation and reports the empirical results obtained under different analytical configurations. Section V discusses the implications of the proposed approach for reproducibility, robustness, scalability, and decision-support systems. Finally, Section VI concludes the study and outlines future research directions.

II. Related Work

Research related to the proposed framework can be grouped into three main areas: reproducible analytics, configurable systems, and Feature Model-based variability management.

A. Reproducible Analytics and Analytical Pipelines

The increasing importance of evidence-based decision making has motivated the development of analytical environments capable of supporting transparent and reproducible workflows. Modern data pipelines organize activities such as preprocessing, feature engineering, statistical analysis, and reporting into structured and repeatable execution stages [6], [10], [12]. These approaches improve consistency and facilitate the replication of empirical studies while reducing manual intervention throughout the analytical process.

Recent research has also emphasized the importance of workflow automation and configurable analytical infrastructures. Hasan and Tausif [7] propose automated machine-learning pipelines for enhanced data analytics, while Pulivarthy et al. [11] discuss scalable data-pipeline architectures for real-time analytical environments. Similarly, emerging scientific workflow platforms increasingly support automated execution and reproducible experimentation across heterogeneous computational environments [8].

Despite these advances, most analytical pipelines remain implemented through predefined execution sequences. Alternative preprocessing strategies, statistical procedures, and validation mechanisms are typically embedded within scripts or implementation artifacts rather than being represented explicitly as configurable analytical decisions. Consequently, methodological variability remains difficult to manage systematically [13].

B. Configurable Systems and Variability Management

Variability management constitutes a central research topic in software engineering because it enables the systematic representation and control of alternative system configurations. Configurable systems have been successfully applied in software product lines, adaptive applications, testing environments, and configurable services [19], [31].

Several studies have demonstrated that variability-aware approaches improve maintainability, adaptability, and reuse by

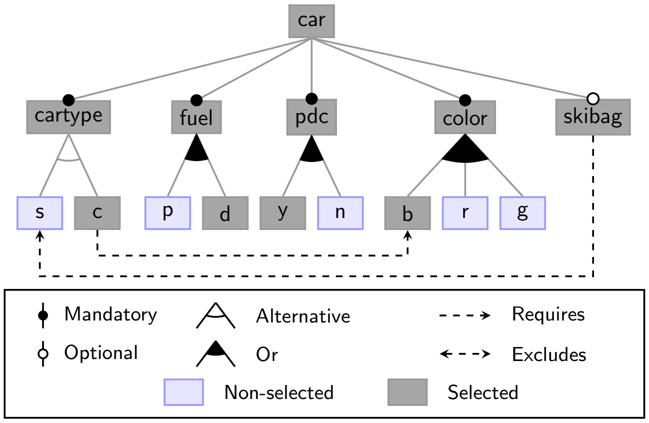


Fig. 1. Feature model illustrating mandatory, optional, alternative (XOR), OR-group, and dependency-based relationships in configurable systems. Reproduced from [32].

explicitly capturing configuration alternatives and dependency constraints [23], [25]. More recently, variability concepts have been extended beyond traditional software products toward machine-learning environments and configurable analytical platforms. For example, Tavares et al. [24] employ Feature Models to support machine-learning model selection, illustrating how variability management can facilitate systematic exploration of alternative analytical solutions.

Although configurable-system research provides mature mechanisms for representing alternatives and validating constraints, relatively few studies have explored their application within intervention evaluation processes. Most existing approaches focus on software configuration rather than methodological variability in analytical workflows.

C. Feature Models and Automated Analysis

Feature Models represent one of the most widely adopted formalisms for variability representation. They provide mechanisms for describing mandatory, optional, alternative, and dependency-related features within a configuration space [18], [20]. In addition to compact variability representation, Feature Models support automated reasoning tasks such as consistency verification, valid configuration generation, dependency analysis, and optimization [20], [27].

Fig. 1 illustrates a simplified Feature Model. The root feature represents the configurable system, while child features describe variability through mandatory, optional, alternative, and OR-group relations. Cross-tree constraints such as *requires* and *excludes* capture dependencies among features and support automated reasoning during configuration generation.

The importance of Feature Model analysis has been demonstrated in multiple application domains. Galindo et al. [18] review automated Feature Model analysis techniques, while Vidal-Silva et al. [21], [22] analyze product configuration mechanisms and automated reasoning approaches for variability-intensive systems. Recent work continues extending Feature Model applications toward adaptive systems, machine-learning configuration, and performance assessment [24], [28].

Despite the maturity of Feature Model research, their application to evaluation frameworks remains largely unexplored. Existing studies primarily address software products, configurable platforms, and engineering systems, leaving a gap regarding the representation and management of methodological variability in intervention evaluation environments.

D. Research Gap

The literature demonstrates substantial progress in reproducible analytics, configurable systems, and Feature Model-based variability management. However, these research streams remain largely isolated. Analytical evaluation frameworks typically provide reproducible execution but limited support for formal variability management, whereas configurable-system approaches focus on software products rather than analytical workflows.

Based on the reviewed literature, no integrated framework was identified that represents intervention evaluation as a configurable analytical system whose preprocessing strategies, analytical methods, robustness procedures, and decision-support mechanisms can be modeled, validated, and instantiated through Feature Models. This gap motivates the configurable evaluation framework proposed in this work.

III. Proposed Configurable Evaluation Framework

This section presents the proposed Feature Model-based configurable framework for reproducible intervention evaluation. The framework conceptualizes analytical evaluation as a configurable system whose components, methodological decisions, and validation mechanisms can be represented explicitly through variability models. Rather than relying on fixed analytical workflows, the proposed approach enables the systematic generation and validation of alternative analytical configurations while preserving methodological consistency and reproducibility.

The framework integrates concepts from configurable systems engineering, variability management, and reproducible analytics [18]–[20]. Analytical decisions are modeled as configurable features connected through dependency constraints, allowing valid evaluation pipelines to be generated automatically from a shared variability representation.

A. System Architecture

The proposed architecture is organized into five interconnected layers, as illustrated in Fig. 2. The first layer is responsible for data acquisition and integration from heterogeneous sources. The second layer performs preprocessing activities such as data cleaning, normalization, transformation, and missing-value handling. The third layer executes analytical procedures including feature construction, statistical analysis, and intervention assessment. The fourth layer performs robustness and validation activities, while the fifth layer generates decision-support outputs for analysts and stakeholders.

This layered architecture separates analytical concerns while allowing different methodological alternatives to be instantiated according to specific evaluation requirements. As a result, evaluation workflows can be adapted without modifying the underlying system structure.

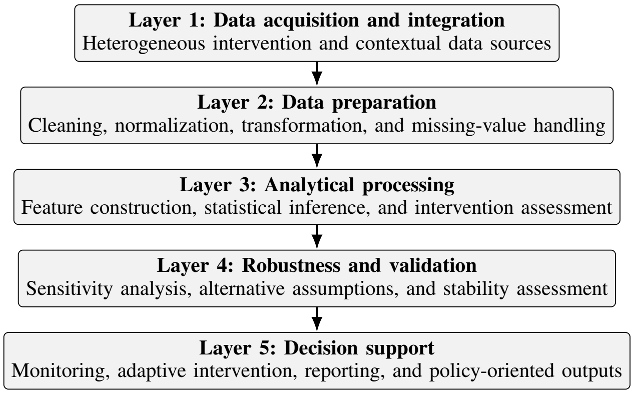


Fig. 2. Layered architecture of the proposed configurable evaluation framework.

B. Configurable Analytical Pipeline

The analytical process is represented as a configurable pipeline composed of sequential but variable stages. Each stage may contain multiple implementation alternatives that can be selected according to evaluation objectives, data characteristics, or methodological preferences.

The pipeline includes five principal activities:

- Data preparation.
- Feature construction.
- Statistical analysis.
- Robustness validation.
- Decision-support generation.

Data preparation includes activities such as cleaning, normalization, and treatment of missing values [14], [16]. Feature construction transforms raw variables into analytical representations suitable for evaluation tasks. Statistical analysis executes the selected analytical procedures, while robustness validation evaluates the stability of obtained results under alternative assumptions. Finally, decision-support generation produces interpretable outputs that facilitate evidence-based decision making.

Representing these activities as configurable elements allows multiple evaluation strategies to be instantiated while preserving analytical traceability and reproducibility.

C. Feature Model Representation

Variability within the evaluation process is represented through a Feature Model. Feature Models provide a compact mechanism for describing mandatory, optional, alternative, and dependency-related features within a configuration space [18], [20].

Fig. 3 illustrates the conceptual Feature Model adopted in this work. The root feature represents the configurable evaluation framework. From this root, variability is organized into four major groups corresponding to preprocessing, analytical methods, validation procedures, and reporting mechanisms.

TABLE I. FEATURE RELATIONSHIPS USED IN THE CONFIGURABLE FRAMEWORK

Relationship	Description
Mandatory	Required in all configurations
Optional	Included when necessary
Alternative	Exactly one feature selected
Or-group	One or more features selected
Requires	Dependency constraint
Excludes	Mutual exclusion constraint

Mandatory features define components that must appear in every valid configuration, whereas optional features allow adaptation to specific evaluation requirements. Alternative groups represent mutually exclusive methodological choices, while cross-tree constraints ensure that incompatible analytical decisions cannot coexist within the same configuration.

Table I summarizes the principal variability relationships used within the framework.

The resulting variability representation enables systematic exploration of valid analytical configurations while ensuring consistency among methodological decisions.

D. Configuration Reasoning

Once variability is represented through the Feature Model, automated reasoning mechanisms can be applied to validate configurations and generate admissible analytical pipelines [18], [20].

Configuration reasoning begins by selecting the desired analytical features. Dependency constraints are subsequently evaluated to verify consistency. Invalid configurations are automatically rejected, while valid configurations are transformed into executable analytical pipelines.

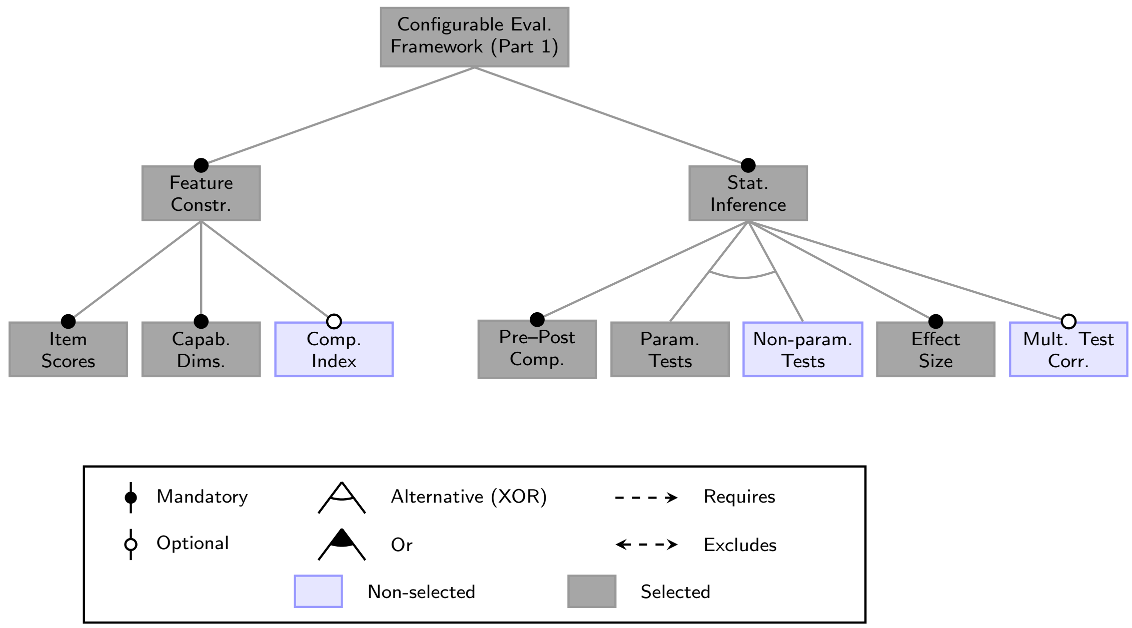
Algorithm 1 summarizes the configuration generation process.

Algorithm 1 Generation of Valid Analytical Configurations

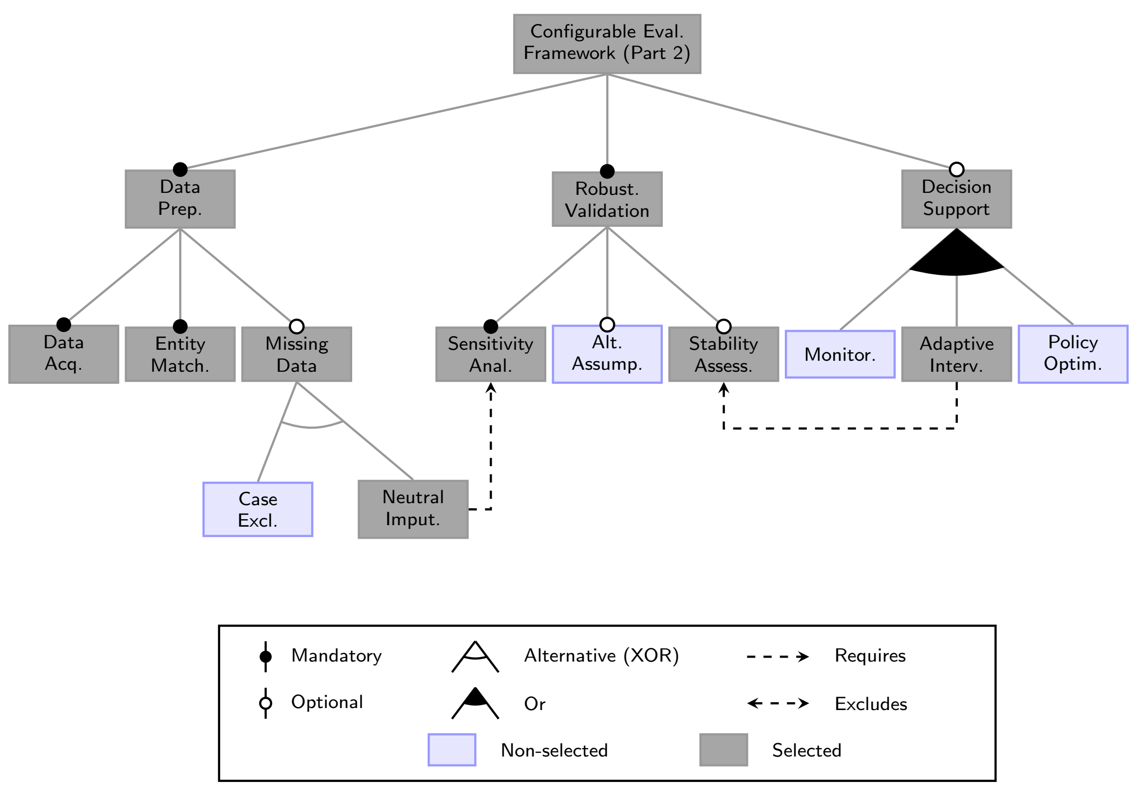
Select candidate features
Validate mandatory features
Evaluate dependency constraints
Remove inconsistent selections
Generate valid configuration
Instantiate analytical pipeline
Execute evaluation workflow

The automated generation of valid configurations provides three important advantages. First, it guarantees methodological consistency through constraint validation. Second, it facilitates reproducibility by explicitly documenting analytical decisions. Third, it enables the systematic exploration of alternative evaluation strategies without requiring manual reimplementation of analytical workflows.

The resulting framework therefore combines variability-aware system design with reproducible analytical execution, providing a foundation for configurable intervention evaluation environments.



(a) Feature construction and statistical inference.



(b) Data preparation, robustness validation, and decision support.

Fig. 3. Feature Model representation of analytical variability in the proposed configurable evaluation framework. Vector PDF sources are used to preserve legibility at publication scale.

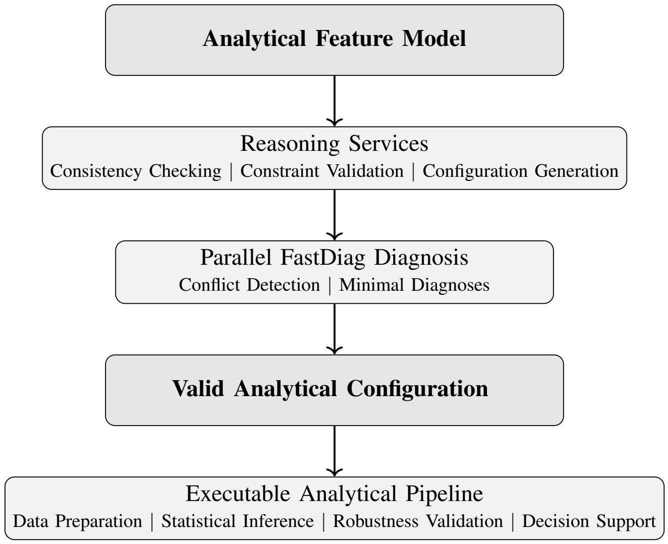


Fig. 4. Architecture of the automated variability-analysis engine integrating Feature Model reasoning, configuration validation, and Parallel FastDiag-based conflict diagnosis.

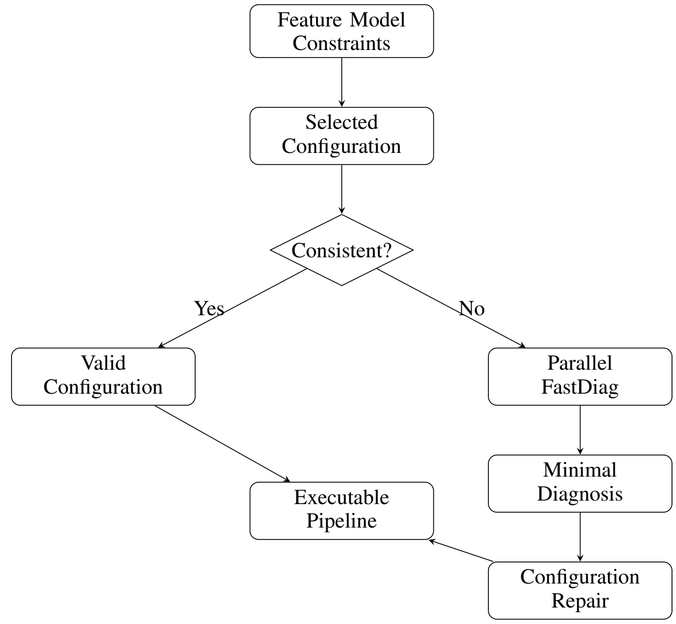


Fig. 5. Automated conflict-diagnosis workflow integrating feature model validation and parallel fastdiag-based configuration repair.

E. Automated Variability Analysis Engine

To operationalize the proposed configurable evaluation framework, an automated variability-analysis engine was developed. The engine combines Feature Model reasoning, constraint validation, configuration generation, and conflict diagnosis services within a unified software architecture. Inspired by the reasoning capabilities commonly available in Feature Model analysis frameworks [18], [20], the proposed solution implements equivalent services in Python and extends them with diagnosis mechanisms based on Parallel FastDiag.

The objective of the engine is to transform variability models into executable analytical pipelines while ensuring that all generated configurations satisfy the dependency constraints defined within the Feature Model. In addition, the engine supports automated diagnosis of inconsistent configurations, providing analysts with explanations and minimal corrective actions whenever constraint violations are detected.

1) Engine architecture: Fig. 4 presents the architecture of the proposed automated variability-analysis engine.

The architecture is organized into four principal components. The first component corresponds to the Feature Model repository, which stores the variability structure representing analytical decisions and their relationships. The second component implements reasoning services responsible for consistency verification, constraint validation, and generation of valid configurations. The third component incorporates a Parallel FastDiag-based diagnosis module capable of identifying minimal sets of conflicting analytical decisions whenever inconsistencies are detected. Finally, the fourth component transforms valid configurations into executable analytical pipelines that can be used during intervention evaluation.

This modular organization separates variability representation from execution logic, facilitating future extensions and incorporation of additional analytical services without modifying the underlying framework.

2) Reasoning services: The reasoning layer provides the core functionality required for automated analysis of analytical Feature Models. Its primary responsibility is to determine whether a selected set of analytical features constitutes a valid configuration according to the structural and dependency constraints defined in the variability model.

The implemented reasoning services include consistency verification, validation of mandatory and optional features, evaluation of alternative and OR-groups, dependency analysis, and generation of admissible configurations. These services are executed before pipeline instantiation to ensure that analytical workflows satisfy all methodological requirements.

Cross-tree constraints are evaluated automatically through logical reasoning procedures. *Requires* relationships guarantee that dependent analytical components are activated together, whereas *excludes* relationships prevent incompatible analytical procedures from appearing in the same configuration.

The resulting reasoning process produces a set of valid configurations that can be transformed into executable analytical workflows while preserving methodological consistency and reproducibility.

3) Parallel fastdiag-based conflict diagnosis: Although consistency verification determines whether a configuration satisfies all Feature Model constraints, it does not explain the causes of an inconsistency. To address this limitation, the proposed framework incorporates a Parallel FastDiag-based diagnosis mechanism capable of identifying minimal sets of conflicting analytical decisions.

Fig. 5 summarizes the diagnosis process executed when a selected analytical configuration violates one or more dependency constraints. The workflow combines automated validation, conflict detection, diagnosis generation, and configuration repair in order to restore consistency before pipeline execution.

As illustrated in Fig. 5, every analytical configuration is first validated against the constraints defined in the Feature Model. Consistent configurations are transformed directly into executable analytical pipelines. When a constraint violation is detected, the configuration is submitted to the Parallel FastDiag module, which computes a minimal diagnosis identifying the smallest set of analytical decisions responsible for the inconsistency.

The resulting diagnosis supports automated configuration repair by indicating which features should be removed, modified, or complemented to restore consistency. This capability improves transparency because invalid configurations are not merely rejected; instead, analysts receive actionable explanations and corrective recommendations that facilitate reproducible analytical evaluation.

4) Implementation availability: The automated variability-analysis engine was implemented in Python to support Feature Model reasoning, configuration validation, and conflict diagnosis. The current implementation extends an existing open-source Parallel FastDiag framework for model-based diagnosis [30] and incorporates additional services for analytical configuration management. The current prototype extends an existing Parallel FastDiag implementation and is publicly available through the following GitHub repository: https://github.com/cvidalmsu/A-Python-FD-implementation

The repository currently provides the diagnosis infrastructure used by the proposed framework and serves as the foundation for the integration of Feature Model reasoning, automated configuration generation, and analytical pipeline management.

To facilitate reproducibility and future extensions, the software prototype is publicly available through a GitHub repository*. The repository currently provides the diagnosis infrastructure used by the framework and serves as the foundation for integrating Feature Model reasoning, automated configuration generation, and analytical pipeline execution capabilities. The repository serves as a living research artifact that supports inspection of the diagnosis engine, replication of experimental procedures, and future extensions of the variability-analysis framework.

The implementation of the automated variability-analysis engine combines variability modeling, reasoning services, and diagnosis mechanisms within a unified Python-based software infrastructure. Table II summarizes the principal implementation characteristics of the current prototype. The architecture was designed to facilitate extensibility, allowing future integration of additional reasoning services, optimization algorithms, and analytical components while maintaining compatibility with the existing diagnosis framework.

As shown in Table II, the proposed prototype combines Feature Models, automated reasoning, configuration generation, and Parallel FastDiag-based diagnosis within a unified Python-based architecture. The reasoning layer supports consistency checking, dependency validation, and automated generation of admissible configurations, whereas the diagnosis engine identifies minimal sets of conflicting analytical decisions whenever constraint violations are detected. Valid configurations are subsequently transformed into executable analytical pipelines, enabling reproducible evaluation workflows. The implementation is publicly available through a GitHub repository, facilitating inspection, replication, and future extension of the framework by the research community.

TABLE II. IMPLEMENTATION CHARACTERISTICS OF THE AUTOMATED VARIABILITY-ANALYSIS ENGINE

Component	**Implementation**
Programming Language	Python 3
Variability Representation	Feature Models
Configuration Generation	Automated Feature Model-based generation
Constraint Validation	Logical reasoning
Reasoning Services	Consistency checking and dependency analysis
Conflict Diagnosis	Parallel FastDiag
Diagnosis Output	Minimal conflict sets
Pipeline Generation	Reproducible analytical pipelines
Execution Platform	Cross-platform
Repository	GitHub
Repository URL	https://github.com/cvidalmsu/A-Python-FD-implementation

IV. EXPERIMENTAL VALIDATION

The objective of the experimental validation is to assess the applicability of the proposed configurable evaluation framework in a real intervention scenario and to determine whether the generated analytical configurations preserve methodological consistency, reproducibility, and robustness. Following common validation strategies adopted in configurable analytical systems and reproducible computational workflows [6], [7], [12], the framework was instantiated using a real intervention dataset and evaluated under multiple valid analytical configurations.

The validation focuses on three complementary aspects:

- Generation of valid analytical configurations from the Feature Model.
- Execution of reproducible analytical pipelines under different methodological assumptions.
- Stability of intervention conclusions across alternative configurations.

A. Intervention Dataset

The empirical validation was conducted using a dataset obtained from a managerial capability development intervention involving micro-enterprises. The intervention followed a pre–post evaluation design in which participants were assessed before and after the training process.

The dataset contains variables associated with managerial capability development and organizational performance indicators. The evaluation framework processes these variables through configurable analytical pipelines that include data preparation, feature construction, statistical analysis, robustness validation, and reporting activities.

*https://github.com/cvidalmsu/A-Python-FD-implementation

The selected dataset is particularly suitable for validation because it contains heterogeneous variables, multiple capability dimensions, and several analytical alternatives that can be represented through Feature Models. Similar intervention-oriented evaluation approaches have been widely adopted in education, healthcare, and organizational development research [1], [2].

Table III summarizes the principal characteristics of the dataset.

TABLE III. CHARACTERISTICS OF THE INTERVENTION DATASET

Characteristic	Description
Evaluation design	Pre–post intervention
Data type	Numerical and ordinal
Capability dimensions	Multiple managerial capabilities
Analytical variables	Performance indicators
Pipeline stages	Five configurable stages
Validation approach	Multiple configurations

B. *Configuration Space Analysis*

Analytical variability is represented through the Feature Model introduced in Section III. The model captures alternative preprocessing strategies, analytical procedures, robustness mechanisms, and reporting options.

During validation, configuration reasoning automatically evaluates dependency constraints and excludes inconsistent selections before pipeline execution. This guarantees that only valid analytical configurations are instantiated.

Examples of configurable decisions include:

- Data preprocessing alternatives.
- Missing-value treatment strategies.
- Statistical analysis procedures.
- Robustness assessment mechanisms.
- Reporting and visualization options.

Fig. 6 illustrates the reasoning process responsible for validating dependencies and generating admissible analytical configurations.

The explicit representation of variability enables systematic exploration of alternative analytical strategies while preserving consistency constraints defined within the Feature Model [18], [20].

C. *Experimental Results*

Multiple valid analytical configurations were generated automatically through the variability-analysis engine described in Section III-E. Each configuration satisfied all Feature Model constraints and was transformed into an executable analytical pipeline through the automated reasoning process.

The generated configurations differed in terms of data preprocessing procedures, statistical inference methods, and robustness validation mechanisms. Despite these methodological variations, all admissible configurations produced consistent intervention conclusions, indicating that the proposed framework preserves analytical reproducibility while supporting controlled variability. This result is particularly relevant because it demonstrates that alternative methodological choices can be explored without compromising the validity or interpretability of the evaluation outcomes.

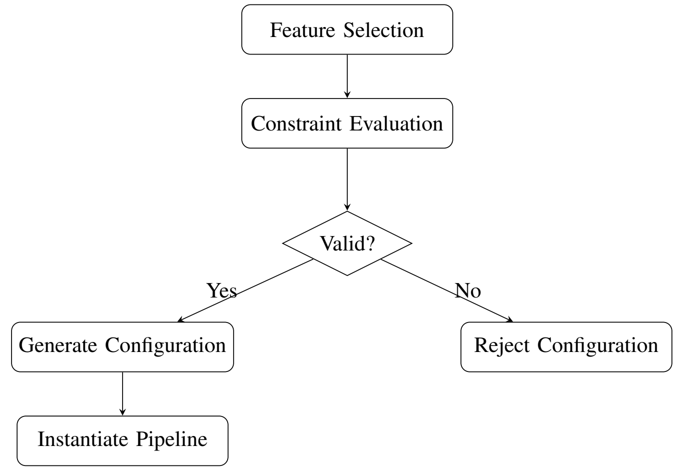


Fig. 6. Configuration reasoning process used to validate constraints and generate admissible analytical configurations.

TABLE IV. STATISTICAL INFERENCE RESULTS AND EFFECT SIZES ACROSS MANAGERIAL CAPABILITY DIMENSIONS

Dimension	$\Delta\mu$	p**-value**	**Cohen's** d
Strategic	0.17	< 0.01	0.65
Financial	0.14	< 0.01	0.52
Operational	0.12	< 0.05	0.48

Table IV summarizes the intervention outcomes obtained from the analytical pipelines generated automatically by the proposed framework. Improvements were observed across all evaluated capability dimensions, with statistically significant differences and moderate effect sizes. The consistency of these results across multiple valid configurations suggests that the framework can support reproducible and interpretable intervention assessments while enabling controlled methodological variability.

In addition to generating valid analytical workflows, the proposed framework incorporates automated diagnosis services through the Parallel FastDiag-based analysis module. Whenever an invalid analytical configuration is detected, the diagnosis engine identifies minimal sets of conflicting methodological decisions responsible for the observed inconsistency.

Table V presents representative diagnosis examples produced by the Parallel FastDiag module. For each invalid analytical configuration, the engine identifies a minimal corrective action capable of restoring consistency. These examples illustrate how the framework supports explainable conflict resolution in addition to conventional configuration validation.

These diagnosis examples were generated automatically

TABLE V. EXAMPLES OF ANALYTICAL CONFIGURATION DIAGNOSES

Invalid Configuration	Minimal Diagnosis
Parametric Tests + Non-parametric Tests	Remove one test
Neutral Imputation without Sensitivity Analysis	Add Sensitivity Analysis
Adaptive Intervention without Stability Assessment	Add Stability Assessment
Multiple Test Correction without Statistical Inference	Add Statistical Inference

by the Parallel FastDiag module integrated into the Python-based variability-analysis engine. For each invalid configuration, the diagnosis process identified a minimal set of conflicting analytical decisions whose correction was sufficient to restore consistency. This capability demonstrates that the framework supports not only configuration validation but also explainable conflict resolution.

The consistency of intervention outcomes across multiple analytical configurations, together with the successful identification of minimal diagnoses for invalid configurations, indicates that the proposed framework effectively combines reproducible analytics, automated reasoning, and conflict diagnosis within a unified evaluation environment. Furthermore, all analytical configurations contributing to the reported results were validated before execution, ensuring methodological consistency throughout the evaluation process.

D. Robustness Assessment

A robustness analysis was performed to determine whether analytical conclusions remained stable under alternative methodological assumptions. Different valid configurations were executed using distinct combinations of preprocessing procedures, analytical strategies, and validation mechanisms.

The resulting outputs exhibited a high degree of agreement across all evaluated configurations. No configuration generated contradictory conclusions regarding intervention effectiveness. This stability indicates that the framework is capable of preserving reproducibility while allowing analytical variability.

To make the qualitative labels reproducible, stability was assessed relative to the baseline pipeline. For configuration c and capability dimension j, let $d_{c,j}$ denote the corresponding effect size and define the maximum absolute effect-size deviation as:

$$D_c = \max_j |d_{c,j} - d_{0,j}|, \tag{1}$$

where, $d_{0,j}$ is the baseline effect size. Stability is classified as *High* when all effect directions and significance decisions are preserved and $D_c \leq 0.10$; as *Moderate* when all effect directions are preserved and $0.10 < D_c \leq 0.20$; and as *Low* when an effect direction reverses, a substantive inferential conclusion changes, or $D_c > 0.20$. These study-specific operational thresholds distinguish negligible, limited, and potentially consequential changes in standardized effect magnitude and should be reported together with the configuration-specific D_c values in the final experimental record. Table VI reports the classification obtained for each analytical configuration.

Furthermore, the explicit representation of methodological decisions provides full traceability of analytical choices, facilitating audit, replication, and comparison of evaluation processes. Such characteristics are particularly relevant in decision-support environments where transparency and reproducibility are essential requirements [12], [13].

Overall, the experimental validation demonstrates that the proposed framework successfully integrates variability management and reproducible analytics within a unified evaluation architecture. The obtained results confirm that Feature Model-based configuration can be used to generate valid analytical pipelines while maintaining consistent conclusions across heterogeneous methodological configurations.

V. DISCUSSION

The principal implication of the case study is not merely that several pipelines reproduced a similar intervention conclusion, but that the analytical design space itself became explicit and machine-checkable. Conventional reproducibility practices typically preserve code, data, and execution environments while leaving the rationale and admissibility of methodological alternatives implicit. By representing those alternatives as features and constraints, the proposed framework extends reproducibility from re-executing one fixed pipeline to inspecting and validating a family of methodologically coherent pipelines. This distinction is important in intervention evaluation because apparently minor choices concerning missing data, statistical tests, or robustness procedures can otherwise remain hidden inside scripts or analyst-specific routines.

The integration of conflict diagnosis adds a second interpretive contribution. A conventional validator can reject an inconsistent analytical configuration, but rejection alone provides limited support for correction or audit. Parallel FastDiag identifies minimal sets of selected decisions that must be reconsidered to restore consistency. In practical terms, this converts configuration failure into an explainable methodological event: analysts can determine whether a problem originates from mutually exclusive tests, an unmet sensitivity-analysis requirement, or a missing prerequisite for an adaptive intervention. Such explanations may improve governance and reviewability in settings where analytical decisions must be justified to researchers, managers, or external auditors.

The results also reveal an important boundary condition. Stability across the configurations evaluated in one dataset does not imply that every admissible configuration is statistically equivalent, nor does it establish that the same Feature Model will transfer unchanged to another domain. Feature selection and cross-tree constraints encode methodological knowledge; therefore, model quality depends on the completeness and correctness of that knowledge. As the configuration space grows, additional challenges may emerge concerning constraint maintenance, computational cost, and the risk of representing procedurally valid but substantively inappropriate combinations. Domain expert review remains necessary even when logical consistency is automated.

TABLE VI. Robustness Analysis Across Analytical Configurations. Stability Labels Follow the Quantitative Thresholds Defined in Eq. 1

Configuration	Significance preserved	Stability	Interpretation
Baseline Pipeline	Yes	High	Reference configuration ($D_c = 0$).
Case Exclusion	Yes	Moderate	Effect directions were preserved, but the reduced analytical sample produced a larger deviation from the baseline effect magnitudes.
Neutral Imputation	Yes	High	The conservative preprocessing alternative preserved effect directions, inferential decisions, and effect-size proximity to the baseline.
Non-parametric Testing	Yes	High	The distribution-free alternative preserved the substantive interpretation and remained within the high-stability effect-size threshold.

From an implementation perspective, separating variability representation from pipeline execution supports controlled extension. New preprocessing methods, inference procedures, or reporting mechanisms can be introduced by extending the Feature Model and the corresponding execution components rather than by rewriting a monolithic workflow. The public Python repository strengthens this contribution by exposing the diagnosis infrastructure for inspection and reuse. Nevertheless, the current repository should be interpreted as an evolving research artefact and integration foundation rather than as a mature, domain-independent evaluation platform.

The empirical scope must also be interpreted conservatively. Validation was conducted with a single managerial capability intervention involving micro-enterprises. Consequently, the evidence supports technical feasibility, configuration-level reproducibility, and within-case robustness, but not cross-domain generalizability. Transferability to healthcare, education, public administration, or industrial analytics requires new Feature Models, domain-specific constraints, and independent empirical replications. A stronger validation program should compare multiple datasets and domains, report the size and structure of each configuration space, measure reasoning and diagnosis time, and evaluate whether independent analysts can reproduce both the selected configurations and the resulting conclusions.

Overall, the framework provides a promising basis for treating methodological variability as a first-class analytical concern. Its main value lies in making analytical alternatives explicit, validating their dependencies before execution, and providing actionable diagnoses when configurations are inconsistent. These capabilities complement, rather than replace, established statistical validation and domain expertise.

VI. Conclusion

This study presented a Feature Model-based configurable framework for reproducible intervention evaluation. The proposed approach conceptualizes analytical evaluation as a configurable system in which preprocessing strategies, analytical procedures, robustness mechanisms, and reporting alternatives are represented explicitly through variability models. By integrating principles from configurable systems engineering and reproducible analytics, the framework enables the systematic generation and validation of alternative analytical configurations while preserving methodological consistency.

The single-case validation demonstrated that the framework is capable of generating valid analytical pipelines through automated configuration reasoning and constraint verification. Across the three managerial capability dimensions, Cohen's d values ranged from 0.48 to 0.65, and the substantive conclusions remained stable across the analytical configurations considered. Within this case, analytical variability was therefore managed without compromising the direction or interpretation of the reported intervention effects. These findings suggest that Feature Models provide an effective mechanism for formalizing methodological decisions, improving traceability, and supporting transparent evaluation processes.

From a broader perspective, the study contributes to bridging two traditionally independent research areas: variability management and analytical evaluation. While configurable systems research has largely focused on software products and adaptive applications, intervention evaluation has generally relied on static analytical workflows. The proposed framework demonstrates how variability-aware approaches can be extended to analytical environments, enabling flexible yet controlled evaluation processes capable of adapting to different methodological requirements.

The framework also offers practical benefits for organizations and researchers seeking reproducible and auditable analytical workflows. The explicit representation of analytical decisions facilitates replication, comparison of alternative strategies, and systematic exploration of methodological assumptions. Furthermore, the modular architecture supports future extensions involving new preprocessing procedures, analytical methods, validation mechanisms, and reporting components.

Several limitations should be acknowledged. The empirical validation was conducted using a single intervention scenario focused on managerial capability development, and additional studies are required to evaluate generalizability across domains such as healthcare, education, public administration, and industrial analytics. Furthermore, the current implementation concentrates on statistical evaluation workflows and does not yet incorporate advanced artificial intelligence or machine-learning services as configurable analytical components.

Future research will focus on extending the proposed framework through the integration of predictive analytics, machine-learning pipelines, automated model selection, and large-language-model-based decision-support services. Additional developments will target the Python-based variability-analysis engine, incorporating advanced Feature Model reasoning services, optimization algorithms for configuration generation, and large-scale benchmarking capabilities. Future empirical studies involving larger datasets and more complex analytical environments will enable a deeper assessment of scalability, robustness, and practical applicability. The public repository will evolve into a comprehensive experimental platform supporting reproducible research, automated diagnosis, and variability-aware analytical systems.

REFERENCES

[1] C. May, "A rational model for assessing and evaluating complex interventions in health care," *BMC Health Services Research*, vol. 6, no. 1, p. 86, 2006. [Online]. Available: https://doi.org/10.1186/1472-6963-6-86

[2] K. Schildkamp, "Data-based decision-making for school improvement: Research insights and gaps," *Educational Research*, vol. 61, no. 3, pp. 257–273, 2019. [Online]. Available: https://doi.org/10.1080/00131881.2019.1625716

[3] C. Benson, I. C. Obasi, D. V. Akinwande, and C. Ile, "The impact of interventions on health, safety and environment in the process industry," *Heliyon*, vol. 10, no. 1, 2024. [Online]. Available: https://doi.org/10.1016/j.heliyon.2023.e23604

[4] C. Vílchez-Román, A. Florek-Paszkowska, and E. Mu, "Charting the evolutionary conceptual pathway of analytic network process research: A main path analysis," *Journal of Multi-Criteria Decision Analysis*, vol. 31, no. 3–4, p. e1837, 2024. [Online]. Available: https://doi.org/10.1002/mcda.1837

[5] M. R. I. Khan, "Ai-powered predictive analytics for enhancing supply chain resilience and delivery outcomes," *Discover Analytics*, vol. 4, p. 13, 2026. [Online]. Available: https://doi.org/10.1007/s44257-026-00065-6

[6] N. O. Nikitin, P. Vychuzhanin, M. Sarafanov, I. S. Polonskaia, I. Revin, I. V. Barabanova, G. Maximov, A. V. Kalyuzhnaya, and A. Boukhanovsky, "Automated evolutionary approach for the design of composite machine learning pipelines," *Future Generation Computer Systems*, vol. 127, pp. 109–125, 2022. [Online]. Available: https://doi.org/10.1016/j.future.2021.08.022

[7] I. Hasan and M. Tausif, "Designing an interpretable and efficient automl pipeline for enhanced data analytics," in *2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 2025, pp. 911–916. [Online]. Available: https://doi.org/10.1109/ICSSAS66150.2025.11081354

[8] W. Shin, R. Souza, D. Rosendo, F. Suter, F. Wang, P. Balaprakash, and R. Ferreira da Silva, "The (r)evolution of scientific workflows in the agentic ai era: Towards autonomous science," in *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC Workshops '25'. New York, NY, USA: Association for Computing Machinery, 2025, pp. 2305–2316. [Online]. Available: https://doi.org/10.1145/3731599.3767580

[9] C. F. Atkinson, "Human in the loop chain of code prompting for deterministic tool development with generative ai," *Discover Artificial Intelligence*, vol. 6, p. 61, 2026. [Online]. Available: https://doi.org/10.1007/s44163-025-00704-z

[10] A. Georgiev, V. Valkanov, and P. Georgiev, "A comparative analysis of jenkins as a data pipeline tool in relation to dedicated data pipeline frameworks," in *2024 International Conference Automatics and Informatics (ICAI)*, 2024, pp. 508–512. [Online]. Available: https://doi.org/10.1109/ICAI63388.2024.10851591

[11] P. Pulivarthy, M. Kommineni, V. M. Aragani, and G. Rajassekaran, "Real time data pipeline engineering for scalable insights," in *Machine Learning, Predictive Analytics, and Optimization in Complex Systems*, 2026, p. 20. [Online]. Available: https://doi.org/10.4018/979-8-3373-5203-9.ch005

[12] K. Bósa and P. Heinzlreiter, "Automated execution of data pipelines based on configuration files," *Open Research Europe*, vol. 5, p. 291, 2025. [Online]. Available: https://doi.org/10.12688/openreseurope.21019.2

[13] M. Esposito, M. Robredo, M. Sridharan, G. H. Travassos, R. Peñaloza, and V. Lenarduzzi, "A critical reflection on the state of data analysis in empirical software engineering," *ACM Transactions on Software Engineering and Methodology*, 2026, just Accepted. [Online]. Available: https://doi.org/10.1145/3799715

[14] M. Afkanpour, E. Hosseinzadeh, and H. Tabesh, "Identify the most appropriate imputation method for handling missing values in clinical structured datasets: A systematic review," *BMC Medical Research Methodology*, vol. 24, p. 188, 2024. [Online]. Available: https://doi.org/10.1186/s12874-024-02310-6

[15] C. M. Vrbin, "Parametric or nonparametric statistical tests: Considerations when choosing the most appropriate option for your data," *Cytopathology*, vol. 33, no. 6, pp. 663–667, 2022. [Online]. Available: https://doi.org/10.1111/cyt.13174

[16] A. N. Prasad, "Data quality and preprocessing," in *Introduction to Data Governance for Machine Learning Systems*. Berkeley, CA: Apress, 2024. [Online]. Available: https://doi.org/10.1007/979-8-8688-1023-7_3

[17] P. Nanjundan, S. Singh, J. George, and S. Eslamian, "Challenges in preprocessing and normalization of heterogenous data," in *Multimodal Learning Using Heterogeneous Data*, S. Eslamian, P. Nanjundan, J. George, and F. Eslamian, Eds. Morgan Kaufmann, 2026, pp. 75–88. [Online]. Available: https://doi.org/10.1016/B978-0-443-27528-9.00015-2

[18] J. A. Galindo, D. Benavides, P. Trinidad, A.-M. Gutiérrez-Fernández, and A. Ruiz-Cortés, "Automated analysis of feature models: Quo vadis?" *Computing*, vol. 101, no. 5, pp. 387–433, 2019. [Online]. Available: https://doi.org/10.1007/s00607-018-0646-1

[19] M. Pol'la, A. Buccella, and A. Cechich, "Analysis of variability models: a systematic literature review," *Software and Systems Modeling*, vol. 20, pp. 1043–1077, 2021.

[20] A. Felfernig, A. Falkner, and D. Benavides, "Analysis of feature models," in *Feature Models*, ser. SpringerBriefs in Computer Science. Cham: Springer, 2024.

[21] C. Vidal-Silva, J. Cardenas-Cobo, A. S. Ortiz, V. Duarte, and M. Tupac-Yupanqui, "Exploring functionality and efficiency of feature model product configuration solutions," *IEEE Access*, vol. 10, pp. 134 318–134 332, 2022.

[22] C. Vidal-Silva, V. Duarte, J. Cardenas-Cobo, J. Serrano-Malebran, I. Veas, and J. Rubio-León, "Reviewing automated analysis of feature model solutions for the product configuration," *Applied Sciences*, vol. 13, no. 1, p. 174, 2023.

[23] B. Tekinerdogan, "Machine learning product line engineering: A systematic reuse framework," *Machine Learning and Knowledge Extraction*, vol. 7, no. 3, p. 58, 2025. [Online]. Available: https://doi.org/10.3390/make7030058

[24] C. Tavares, N. Nascimento, P. Alencar, and D. Cowan, "Variability-aware machine learning model selection: Feature modeling, instantiation, and experimental case study," *IEEE Access*, vol. 13, pp. 62 527–62 542, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3558218

[25] D. T. M. da Cruz, N. F. S. Seixas, M. d. S. Oliveira, M. E. S. Freire, R. R. G. Souza, F. A. Durão, C. V. S. Prazeres, I. d. C. Machado, G. B. Figueiredo, M. L. M. Peixoto, H. M. Iwamoto, and E. S. d. Almeida, "Reengineering an adaptive system to dynamic software product lines: An experience report," in *Anais do XIX Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software (SBCARS)*, 2025.

[26] M. Abbas, M. Saadatmand, E. P. Enoiu, and B. H. Schlingloff, "State of test optimization for variability in industry," in *The 22nd International Conference on Information Technology-New Generations (ITNG 2025)*, ser. Advances in Intelligent Systems and Computing, S. Latifi, Ed. Cham: Springer, 2025, vol. 1463. [Online]. Available: https://doi.org/10.1007/978-3-031-89063-5_46

[27] H. Liu, X. Wang, and W. Wang, "Computer-aided formalization of internal consistent product family models," *Applied Sciences*, vol. 13, no. 3, p. 1792, 2023.

[28] J. M. Sánchez Ruiz, G. Ciria González, M. Á. Olivero, F. J.

Domínguez Mayo, and D. Benavides, “Feature-based characterization of github oss projects for performance assessment,” in *Proceedings of the 29th ACM International Systems and Software Product Line Conference - Volume A*, ser. SPLC-A ’25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 94–104. [Online]. Available: https://doi.org/10.1145/3744915.3748467

[29] C. V. Silva, “Configuration analysis for large scale feature models: Towards speculative-based solutions,” PhD thesis, Universidad de Sevilla, Sevilla, Spain, 2021. [Online]. Available: https://hdl.handle.net/11441/128274

[30] D. I. Carrión León, C. Vidal-Silva, and N. Márquez, “Scalable model-based diagnosis with fastdiag: A dataset and parallel benchmark framework,” *Data*, vol. 10, no. 9, p. 141, 2025. [Online]. Available: https://doi.org/10.3390/data10090141

[31] R. Bögli, A. Boll, A. Schultheiß, and T. Kehrer, “Community-driven variability: characterizing a new software variability paradigm,” *Automated Software Engineering*, vol. 33, no. 2, p. 67, 2026.

[32] C. Vidal-Silva, A. Felfernig, J. Galindo, M. Atas, and D. Benavides, “A parallelized variant of junker’s quickxplain algorithm,” in *Foundations of Intelligent Systems*, ser. Lecture Notes in Computer Science, D. Helic, G. Leitner, M. Stettinger, A. Felfernig, and Z. W. Raś, Eds. Cham: Springer, 2020, vol. 12117.