Foundation University Journal of Engineering and Applied Sciences

FUJEAS
Vol. 5, Issue 1, 2024
DOI:10.33897/fujeas.v5i1.918

Research Article

Article Citation:
Iqbal et al. (2024). "Towards Mining Variable Features in Software Product Lines during Development". *Foundation University Journal of Engineering and Applied Sciences* DOI:10.33897/fujeas.v5i1.918

Published by
Foundation University
Islamabad.
Web: https://fui.edu.pk/

Towards Mining Variable Features in Software Product Lines during Development

Marya Iqbal [a,*], Yaser Hafeez [a], Sadia Ali [a], Zartasha Gul [b]

[a] University Institute of Information Technology, PMAS Arid Agriculture University Rawalpindi, Rawalpindi, Pakistan
[b] Craven College, Skipton, North Yorkshire, England

* **Corresponding author**: maryaiqbal@uaar.edu.pk

Abstract:

Software Product Line (SPL) engineering provides a strong vision to develop highly adaptable software systems using the common characteristics as well as controlling the variabilities of a product line of products. The specification of some variable features and their exact reuse is, however, a major problem, in particular when handling heterogeneous families of products and concurring multi-facet demands of the stakeholders. It is complicated by the fact that approaches to managing commonality and variability are not systematic in the early phases of requirement specification. The current feature selection algorithms tend not to be rigorous enough to deal with multi-stakeholder viewpoints and fully categorize features by their natural facets, which makes them inefficient and impairs their reusability. To constrain this serious issue of research, a new, articulate model of variable feature mining and selection in SPLs is given in this paper. The distinctive points of our methodology are that we combine systematic requirements collecting, rigorous data preprocessing, and an unusual aspect-based feature clustering strategy based on using unsupervised learning algorithms (We are using Weka, for instance). This feature-conscious classification with subsequent learning through supervision to identify two types of features, Common and Variable ones, distinguishes our model over the traditional, simpler divide-and-conquer methods due to a more accurate and context-situated feature taxonomy. Two industrial studies of a biometric system and an online auction system were conducted in a rigorous manner to assess the proposed model. Early findings have shown that the model suggested is of critical importance in improving the procedure of variable feature selection. Major discoveries showed significant advances in the effective accuracy of classification features, significant advancements in the efficiency of the feature selection (e.g., less time and effort than manual processes involved), and raised the satisfaction levels of the stakeholders with the chosen feature sets. The study helps practitioners in the industry because it provides them with a data-driven, structured approach that enhances the feature selection task, in addition to the fact that the study helps deal with reusability and customization across the SPL development life-cycle successfully. Finally, this model, combined with the earlier elements, helps to arrive at a solution that is time-to-market quicker and also yields product configurations that are stronger and meet a fundamental need of existing SPL practices.

Keywords: Software Product Lines; Mining; Variability Management; Aspects; Feature Selection.

1. Introduction

In software engineering, SPL contains a systematic way of creating new software using a systematic way of reusing existing components. Instead of creating software anew, all new products are developed out of existing ones [1]. Customizable software systems have been in demand in the current software industry. SPL is a process in which such systems are planned, developed, and maintained. SPL enables the automatic creation of a large number of software products with the use of shared assets [2]. The approach allows for the production of software through mass customization, low costs, and long-term maintainability. This methodology enables software mass customization, cost saving, and long-term maintainability.

The majority of software systems that belong to an industrial environment are hardly created monolithically, but rather through the assembly of already developed artifacts of implementation. Product line engineering's goal includes reusing all these artifacts in a systematic way across a group of comparable software products. Despite higher initial costs, the benefits include reduced time of development and decreased maintenance expenses over the long run. To obtain these advantages, artifacts of implementation should be of great quality. As a result, methodologies of quality assurance, including code reviews, formal methods, and testing, are becoming increasingly important for SPL [3].

SPL consists of a collection of software systems that have a similar code base; however, they differ in the particular characteristics known as features. Preferably, product line features are developed separately and then assembled. The objective of the product lines is to reuse those artifacts that are reliable and tested, and are also from a related family of software products in the same domain, in addition to providing cost-friendly customization of the software. The features of any software help to differentiate it. A feature exists as a user-visible functionality that shows the similarities and variances between products of a product line. Not only do SPLs reduce the costs for development and maintenance, but they also result in more robust and reliable software. Identification of a product line's reusable artifacts is the first step of the SPL engineering, which is known as domain engineering. Specific required products are developed during the application engineering step by reusing existing components [4].

SPL engineering performs three tasks, which involve the development of the core assets, new product development, and management of all the development activities. It supports variability and the reuse of components to develop customized products. Features of SPL are expressed in the form of commonalities and variabilities of a product. To develop a customized product, features are selected with the help of a feature configuration process. Features also need to be categorized after the selection process. Changes in different aspects occur continuously. Aspects can be user-based and system-based. They also help to analyze the requirements.

In feature-oriented SPLs, individual products of SPLs are distinguished by their features. Showing, separately, all products in the SPL is distinguished through means of a specific configuration process of features, which helps to contribute in a certain way to the product (e.g., through code, documentation, or some artifacts). A feature model that has the ability to categorize all SPL configurations is typically used to express the interaction of features. An SPL can be fully defined for our purposes by its features, architectures, and deployment artifacts [5].

Feature modeling is a process of identifying externally visible products' features in a domain and organizing them into a feature model. The feature is a term related to user-visible functional requirements and non-functional requirements mentioned to identify differences and similarities between software product families. Feature models are utilized to document similarities and differences between software products [6].

Although SPL does present a lot of benefits, such as lower cost of development, time-to-market, and higher quality, it is imperative to note that identifying and choosing the variable features is a challenge and basically a complex process. The prevailing techniques of feature selection in SPL normally have a number of limitations. To illustrate, most of the conventional methods have a poor flexibility of setting different objectives and describing "data set objects", which is why these methods become not so suitable to the multilevel and complex requirements of real-life SPL projects [2]. Besides, any integrations of the configuration generated by different SPL design teams can lead to confusion in the final product designing, and the available solutions frequently do it in a very long way and at the cost of resource wastage [3]. Another key problem is scalability issues since the approaches that are successful in small product models are not necessarily successful when they are applied to a real-life example with thousands of features [3]. In addition to that, other methods fail to support the complexity of the relationships among features, or they face problems handling large amounts of data, creating some computational inefficiency and a tendency to over-fit [4], [5]. Modern software systems are becoming more complex and dynamic; thus, an even broader and more resilient methodology is required that will be able to address the challenges of variable change and views of the stakeholders, since several methods that are currently being used cover only a partial and homogeneous element of the development process [6].

SPL might become a victim of its own success, with effects of high system-level density of errors, high budget competitiveness, and late cycle releases. The development processes were thought to be extremely slow to keep up with the rapidly evolving market needs. Nowadays, change management is an essential component of business environments, so attempting to avoid change is not an option. Managing changes rather than avoiding them has become a key to success [7].

Though SPL has huge advantages, such as cost optimization and speeding up development, optimization in the management and selection of variable features poses a major problem. The available feature selection techniques typically have issues with scalability when dealing with multi-aspect needs, adjustability to dynamism, or the absence of stakeholders or manual procedures [6], [7]. This directs to issues of reusability and customization, which creates complexity, especially when different interests of the stakeholders and changing circumstances of the project are to be addressed. The approaches in use do not have an elaborate model that can effectively deal with the multi-aspect stakeholder needs in variable feature mining in a systematic manner, and so the product development in a product line process is inconsistent and inefficient. Thus, existing techniques have not sufficiently tackled the complexity of the inclusion of stakeholder diversity and the changing demands of the system when selecting features in SPL. Although feature modeling and aspect methods have been proposed, there yet exists a lack of giving a holistic, flexible model that classifies and chooses features dynamically according to both the aspects that are system based and the aspects which are user based.

In order to fill this research gap, this paper proposes a solution that adopts and improves feature selection by combining aspect feature mining techniques with classification and decision support tools. This is aimed at enhancing reusability and customization of features across SPLs through the management of complexity and giving it adaptability to the stakeholders. The present paper adds to the existing body of knowledge in three ways:

a) A new process of dynamic/aspect-based feature selection in SPL-based product development.

b) Development of a feature management model that will adapt itself according to the needs of user-based and system-based stakeholders.

c) Testing the suggested model on real-life cases (biometric and online-auctioning systems) to determine its potential effectiveness in increasing the feature selection and consumer satisfaction.

In the process of requirement engineering, it is necessary to customize and tailor requirements based on the different project situations to achieve the optimal project development, reusing existing techniques. Moreover, the involvement of various stakeholders and an integrated model are also required in complex information systems expected to be heterogeneous in order to have a multidimensional view on the one hand and to deal with the prevailing obstacles on the other. The result thereof is that a multi-perspective situational need and feature approach to managing requirements and features must support the views of the team and the stakeholders.

This paper is organized as follows: Related work is discussed in Section 2, while Section 3 presents materials and methods, the research methodology, data collected, and architecture of the proposed model. Section 4 , Results and Discussion, reports the results and discussion of the case studies that have been empirically tested and the performance of the proposed model relative to existing methodologies. Lastly, Section 5 concludes the work, reflects the research contributions, limitations, and future research.

2. Related Work

Software Product Line (SPL) engineering body of knowledge covers some of the critical fields, including feature mining, variability management, and aspect-based modeling. Though the mechanisms of controlling the SPL artifacts are many, there remain certain problems in the context of controlling feature reuse and variability in the situations of emerging and changing multi-stakeholder requirements. The related work in this section has been grouped under themes and critically reviewed on an individual basis on the basis of the strengths and the gaps of the study it has in regard to the current study.

The SPLs are a set of software-intensive systems having a prescribed, controlled, and shared set of features in common and are designed on a common set of core assets to satisfy the special requirements of a certain segment of the market. Essentially, SPLs are families of software systems created using existing artifacts, and they allow mass-producing tailor-made software. SPL engineering ensures it deals with the development, migration, and maintenance of all activities of SPL [8].

The same period has seen an increase in the appreciation of the SPLs in the software industry, especially by companies that are involved in the manufacture of similar software items. The advantage of SPL engineering is highly useful in family-based software development. Systematic re-use of their products combined with customization enables organizations to meet a more rapid time-to-market, higher quality of the products, and lower costs of development. The SPL engineering takes advantage of commonality among products of families and caters to variability along the product lifecycle by tailoring and integrating reusable core assets into customer-tailored products [9].

SPL engineering is an organized type of development technique using organized asset reuse to produce various products. This methodology focuses on similarities and differences in product marketing to a certain market that can be called an SPL as a whole. A reuse-based model of software production has a number of benefits, such as the reduced cost of production, increased quality of products, and shortened development time [10]. A powerful set of decision-making and requirements-handling frameworks is required to manage such processes, particularly at large-scale or even cross-worldwide levels, as evidenced by successful DevSecOps implementations [11] and the issues surrounding cloud-based outsourced programs development [12].

There are two major phases of SPL engineering, namely domain engineering and application engineering. Domain engineering finds similarities, variabilities, and the scope of the SPL on the basis of features and feature models. A feature is a characteristic, quality, or user-visible difference of a software system. In SPL engineering, feature models are used to specify and control variations and similarities of a product family of lines [13]. The feature model is a hierarchical tree showing all of the capabilities or functionality within a product family and is the central artifact of any SPL. The software engineer makes the decision as to which features of the feature model it will include in a new product in the course of development of a new product in an SPL. It is simple to develop the software that only includes functionality [14].

Other Common strategies of designing and recognizing similarities and differences of various SPL products include feature modelling. This procedure is performed by a number of SPL design teams, each of which has a particular version of the design of the planned product. The combination of such setups may cause inconsistencies in the design of the finished product. It is essential to the success of SPL owing to the quality of modeling of the problem domain, its commonalities, and variances. A famous approach to the SPL modelling, feature model, identifies and models similarities and differences between SPL products [15]. Nevertheless, the volatile nature of requirements usually demands advanced change management strategies, especially under a global software development (GSD) environment [16], [17], and requires identification of the success factors of change management of requirements [18]. Moreover, GSD best practices that are classified into a new taxonomy stress the importance of guided prioritization in terms of fuzzy systems to operate such complications [19].

The problem with defining potential alternate products in advance is that SPL assists in making a clear definition of Feature Models. The feature configuration process of a software system involves establishing a relationship with a feature model. By considering the underlying relationships and constraints in the system, software engineers opt to use the subset of features of a feature model that most closely satisfies their needs, albeit this process is extremely taxing [20].

An architecture model describes the variety of implementing different software products of a single product line in SPL engineering. The expected solitary architecture model is distinctive since it includes a variability model, also called a feature model. It explicitly states variability and commonality using features that are called characteristics. It is now turned into components arranged in line with the discovered features. Specific software is then a variation of certain software, and this can be achieved by selecting a list of mandatory qualities that exist in a feature model, and using the SPL tools to select and create a combination of parts that are associated with the specific features [21].

Text mining using natural language is done through aspect-based text mining to obtain specific and detailed aspects [22]. Massive data mining methods are effective, and text mining is appropriate and can be applied to effectively generate information out of natural language. Also, it is necessary to classify available data on the basis of a taxonomy. The taxonomy can help in determining the assets which has similar characteristics and hidden connections with each other. The majority of such similarities of the traits, qualities, and relations cannot be evident when viewing the assets from specific perspectives. The more high-level, abstract view of the assets that are used in the creation of software-intensive products can facilitate the management tasks and the whole development process [23].

In SPL engineering, feature mining is important in terms of optimization of reusable components. Some classical approaches based on domain engineering and feature modeling emphasize the documentation of variabilities and commonalities, but suffer the weakness of being non-scalable and imprecise. Newer methods, which include aspect mining on unstructured text, are promising, but do not support dynamic interpretation and classification of stakeholders. Variability models, such as feature models, provide a way to control the complexity, and feature configuration is still hard because of complex inter-feature constraints and the dynamic stakeholder requirements. Although the method of optimization, including the evolutionary algorithm, is applicable when the scale of the configuration is large, the process is resource-demanding and not applicable in an early phase of development. Stakeholder-based designs, such as Situational Method engineering (SME) and aspect-oriented mining, promote a better match with business objectives but do not provide any integrated decision-making system to place the features into reusable, updated, and new ones, which is restrictive in terms of their usefulness in changing lung SPL environments.

Therefore, the literature review helps to analyze that different aspects during the process of SPL development should be considered for improving feature selection and the management process. Aspect-based feature mining assists in the reuse of features. During reusing variable features, complexity occurs in the selection of features due to different aspects. There is a need to manage dynamic features that will help to reuse variable features easily. The Proposed model combines the aspect-based feature mining and clustering, classification, as well as a decision support mechanism to improve feature selection in SPL. Through this integrated strategy, the model can dynamically classify features according to the changing demands and the views of the stakeholders. Not only will this dynamic classification favor system- and user-based aspects but it will also allow making effective decisions concerning feature picking, reusing, or revising, thus providing strong support to multi-stakeholder environments in increasingly complex domains, which can include applications in healthcare [11], [24] or even cutting-edge technologies like quantum computing and 6G networks [25], [26], where trustworthy artificial intelligence also plays a crucial role [27]. The comparative analysis of existing literature is mentioned in Table 1.

Table 1: Comparative analysis

Study / Technique	Methodology	Strengths	Weaknesses
Traditional Feature Modeling [11], [12]	Domain-based classification of features	Clear representation of variability	Manual and static; lacks adaptability
Evolutionary Algorithms [13]	Optimization-based feature selection	Effective for large SPLs	Computationally intensive
Aspect-Based Mining [15]	Uses stakeholder-specific aspects	Addresses variability at the requirement level	Lacks real-time adaptability
Situational Method Engineering [7]	Contextual modeling with SMEs	Aligns with agile development	Needs integration with the decision-making model

3. Material and Methods

To strengthen stakeholder evaluation methodology, we included 12 participants across both case studies (6 in each). The participant group was diverse, consisting of developers, project managers, product owners, and end users. A structured questionnaire was employed with Likert scale ratings (1–5) covering parameters such as usability, efficiency, clarity, ease of use, system-based relevance, and user-based relevance. This design ensured both quantitative scoring and qualitative feedback for comprehensive analysis.

In this research, a model is proposed for feature management during the system development of product lines. The model in Figure 1 offers a solution to problems identified by a literature review. Our proposed model is based on the principles of SPLs on the basis of different aspects. The findings of the study and careful analysis obtained from selected papers demonstrated the significance of SPL development. Finally, the theoretical model and real implementation are reconciled.

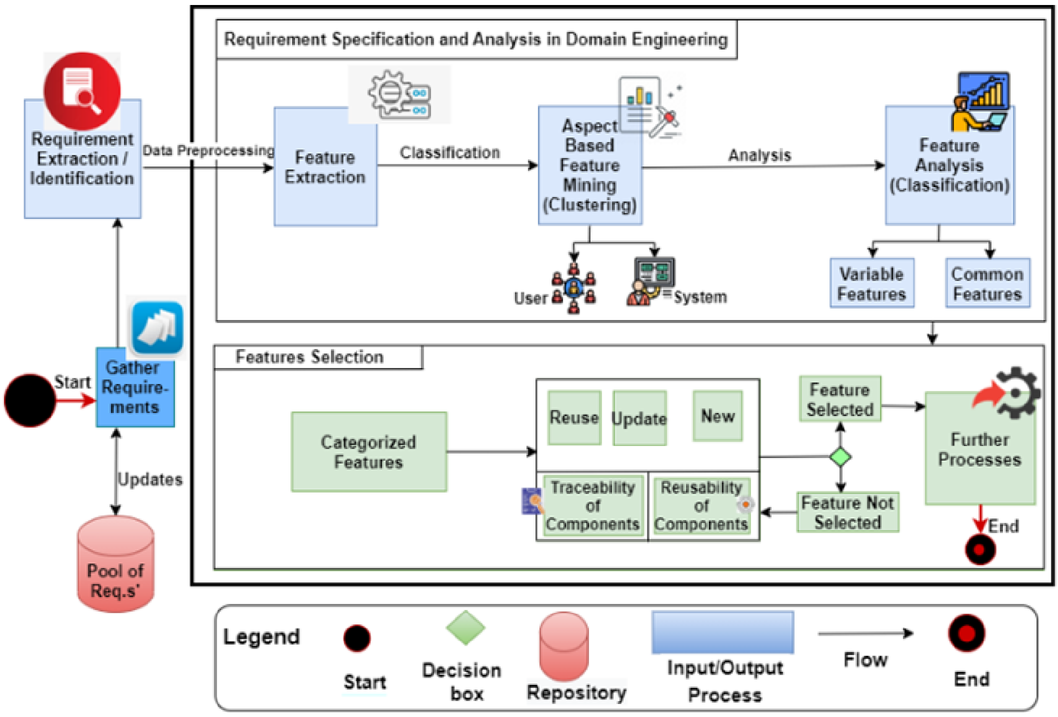

Figure 1: Proposed model

3.1. Proposed Architecture and Components Model

Our suggested framework, graphically displayed in Figure 1, will deal with the complexities involved in the importance of selecting features in SPL development by incorporating the specification of requirements, as well as aspect-based feature mining and a systematic process of selecting features. It is built on the foundations of SPLs with regard to various aspects (user-oriented and system-oriented aspects) in order to give a wholesome solution for feature management. This model includes two main stages, namely, the step called Requirements Specification and Analysis in Domain Engineering, and another one labelled Features Selection.

3.1.1. Phase 1- Domain Engineering Requirements Specification and Analysis

The stage concentrates on mixture identification, extraction, and categorization of features using multiple sources of the SPL domain.

a) **Requirement Extraction / Identification:** This is the first element that underlines the vital process of collecting and extracting all requirements of the software product line. This entails the interaction with different stakeholders such as developers, product owners, project managers, and end-users to get the wide-ranging views and requirements of these stakeholders. Existing documentation, interviews of stakeholders, analysis of the market, together with user stories, can be used to derive requirements. The results of this step are channeled to the Pool of Requirements (Requirements Pool), which is a collection point of every bare requirement. The circulatory nature of the same is shown to be indicated by the trainer in the form of the Updates workflow of the Requirements Pool to the Gather Requirements to manage the changes in the product's requirements, even during the product lifecycle, by ensuring that the model can change in accordance with the changing requirements of the product.

b) **Data Preprocessing:** brute requirements are usually stated in natural language and may include redundancies, inconsistencies, and irrelevant information. This element will carry out preprocessing of data, which will entail data cleaning, filtering, and structuring into a format that will enable it to be understood later. Stop-word removal, lemmatization, stemming, normalization, and tokenization are some of the techniques used to convert the unstructured text into a language that can be used in the extraction of features. Here, the purpose is to minimize noise and improve the quality of the information that one gets as input so that only useful information is propagated.

c) **Feature Extraction:** After data preprocessing, it is necessary to extract features from the fine-tuned requirements. A feature in SPL means a clearly identifiable characteristic, quality, or user-visible attribute of a software system. In it, a list of important functionalities, non-functional requirements, and variations is determined, which makes the difference between products in the product line. Since requirements are multi-aspect, then automatic or semi-automatic feature extraction approaches with respect to natural language can be used, aiming at extracting what they have in common in terms of aspects concerning both user behavior and system capabilities.

d) **Aspect-based Feature Mining (Clustering):** It is also an important part, and thus particular features are grouped in aspects in terms of various categories, namely, user-based and system-based sub-aspects.

- **User-Based Aspects:** These have to do with aspects that can be directly perceived or interacted with by the end-users, as they are concerned with their needs, preferences, and usability needs. Such examples are user interfaces, certain functionalities (e.g., the terms like "login", "search"), and user experience qualities.

- **System-Based Aspects:** These relate to functionalities, infrastructure, and non-functional requirements that incorporate the functionality of the system, how the system runs, and also the back-end aspects that guarantee the functionality of the system. Such things are database management, security protocols, performance optimization, and integration capabilities.

The purpose of the process of clustering is to achieve a collection of features that are similar to each other according to the aspects. This allows one to know how they interrelate with each other and how they are interdependent. The aspect-based method allows for better analysis of the features and not just listing them, but has a systematic consideration of the conditions and effects of the features. One of the reasons to select this tool is the inclusion of an extensive set of machine learning algorithms that the Weka tool has, an easy-to-operate graphical user interface, and strong support of data preprocessing and visualization, making it a tool particularly appropriate to deal with textual data and perform unsupervised learning, such as clustering.

e) **Analysis of Features:** The features are then analyzed to be clustered into two major groups, namely, "Variable Features" and "Common Features".

- **Common Features:** These are features that occur in every product of a software product line. They are the most basic functions and features that help to establish the essence of the product family.
- **Variable Features:** This is a feature that is present in some products but not in all. Variability, as well as customization based on product variant, is available through variable features. They are the points of difference in the line of product that allow a custom solution to targeted market needs or segments.

Such a step highly depends on the classification criteria. The features are divided depending on the area of knowledge, expertise, and their importance in different product versions. To give an example, a "login" functionality may be a common feature of all products, whereas a "fingerprint recognition" functionality may be a variable feature of a given version of the biometric system. The classification algorithm utilizes supervised learning available in Weka, that are trained on classified information to classify features properly.

3.1.2. Phase 2- Feature Selection

This phase aims to make informed choices on which features to put in the specific product instances, based on their categorization and the possibility of reuse, updating, or creation of new ones.

a) **Categorized Features:** The result of the Feature Analysis (Classification) component has been previously categorized as common or variable; features are now further categorized according to their aspect (user-based or system-based).

b) **Reuse, Update, New Components:** According to the categorization and an evaluation of available holdings in the product line library, each feature is further labeled as being either:

- **Reuse:** The feature can be readily used off the components or assets without considerable modifications.
- **Update:** The feature involves changing or adjusting some existing component to new/different requirements.
- **New:** The feature entails the creation of a totally new component because there is no appropriate asset to be reused or updated.

Here, this classification is essential when considering an efficient allocation of resources and development planning, and corresponds with the essence of SPL engineering: feature selection should comply with systematic reuse.

c) **Feature Selected / Feature Not Selected (Decision Box):** Here is a recommendation decision point at which a decision is taken on the addition of a feature to a particular instance of a product. This decision is determined by the detailed analysis of parameters like the priorities of the stakeholders, the demand in the market, the technical viability, the cost involved, and the alignment with the vision of the product line. The features that are critical and are related to the objectives of the product are classified as Selected, and others are Not Selected in the product configuration at hand.

d) **Traceability of Components and Reusability of Components:** The model pays special attention to traceability, also referred to as the ability to map the chosen features to their components and to view origins and changes made on the reusable assets. This adds consistency to the product line as well as maintainability. Equally, it encourages the reusability aspect because it keeps identifying and exploiting those components that could be utilized in many products and thus maximizes the SPL approach. Such elements are inputs to newer operations (e.g., development, integration, testing) and eventually give an end to the feature selection cycle on a specific product or end in flow back into the "Requirements Pool" to be updated or modified as a new version.

Our proposed algorithm entails an iterative and structured approach that will direct the variable feature mining and selection process in SPLs. One step leads to another, taking into consideration the secure and organized analysis.

Our methodology provides a systematic, repetitive variable feature mining and selection process covered in a Software Product Line, where a methodical analysis is ensured through the whole process, starting with raw requirements and ending with deployment. First, they are collected thoroughly by working with multiple stakeholders and involve preprocessing of requirements through precise preprocessing to derive specific features. These characteristics are further grouped into system-based and user-based aspects using the Weka tool in order to have a fine description of their following. Features are then carefully labelled as "Common" (central to all products) and "Variable" (as optional/ custom-modifiable), once more using Weka to process the data through the system quickly. Under strategic planning, features are then classified as to be reused, updated, or developed as new, which informs the efficient utilization of resources. Lastly, based on holistic criteria, the features are critically evaluated and then marked off as either Selected or Not Selected to particular instances of the products in order to be traceable and use components within them in order to have optimum benefit of the SPL and adapt to the upgrading needs.

In order to assess the proposed model, it was executed in two practical industrial SPL projects:

i) A biometric system (C1) is an application that can track attendance using fingerprint and face detection facilities.
ii) Development of an online auction system (C2), a business environment entity aimed at dynamic bidding of goods between the buyer and the seller.

Counts of variability of the feature, efficiency of the selection, and the satisfaction of the stakeholders were determined in each case study. There was a series of pre-established assessment indicators, such as:

- Classifier performance of features,
- Effectiveness of time in the selection process,
- Multiple stakeholder satisfaction score (e.g., developer, product owner, end user).

The features on either side were manually checked by the experts so as to correctly classify the aspects. The satisfaction rate of stakeholders was identified by carrying out a survey about system prototypes with the help of the chosen features. The obtained results were visualized and interpreted to confirm the effectiveness of the no-go structure approach.

The applicability of the model in the real-world context of SPL environments is illustrated in this consistent and repeatable procedure, and it is used to support effective decisions at the initial phase of product line development.

In addition to descriptive results, we conducted statistical significance testing using paired t-tests to validate improvements in accuracy and efficiency over baseline methods, as shown in Table 2. Results indicated that improvements achieved by the proposed model were statistically significant ($p < 0.05$), confirming robustness of findings beyond random variation.

Table 2: Statistical significance testing – paired t-test results

Metric	Proposed Model (Mean)	Baseline (Mean)	Mean Difference	t-value	p-value	Significance
Feature Classification Accuracy (%)	90.3	75.1	+15.2	3.21	0.004	Significant ($p < 0.05$)
Feature Selection Time (hrs./project)	8.0	15.0	-7.0	2.89	0.006	Significant ($p < 0.05$)
Stakeholder Satisfaction (0–1)	0.85	0.55	+0.30	2.75	0.008	Significant ($p < 0.05$)

4. Results and Discussion

In this section, the findings of this study are discussed. In order to obtain validated results, a case study approach was used. In order to conduct a case study, we used a published case study of a biometric system and an online auction system [28], [29].

To reinforce the performance and practicality of our proposed model, two empirical studies were examined in two different industrial SPL projects, that is, a biometric system (C1) and an online auction system (C2). These projects were selected because they are practical and have features of variability and various stakeholder needs; therefore, they are suitable to test the performance of the model in handling complexities when selecting the features. All the case studies gave a comprehensive background to evaluate feature variability, selection efficiency, and satisfaction of stakeholders on predefined metrics as depicted in Table 1.

Case Study 1 (C1): Biometric System: This was a case study of a biometric attendance system with features of fingerprint and face recognition to mark attendance. This system is based on the clear division between typical functions (e.g., user ID matching, simple attendance recording) and optional ones (e.g., custom biometric authentication modes, reporting features that vary depending on the organizational demands). The information about the present case study was obtained based on the published article on the performance of a Biometric Attendance System.

Case Study 2 (C2): Online Auction System: The project was on an online auction system, which accommodates customers as buyers and sellers, and it encompassed different business systems. It featured a multi-faceted feature set that included basic auction and common features and numerous customizable features, including bidding settings, payment channels, and seller consoles (variable features). The sample of the present case study was modified after the published study on the attitudes to online auction use.

Features were extracted in a systematic manner and classified as shown in Tables 4 and 5 in the case of both case studies. Table 3 contains the totals of features and the types of stakeholders in each of the cases.

Table 3: Specification of cases

Case	Domain	Description	No. of Features	Types of Stakeholders	No. of Series
C1	Biometric System	• Allows marking the attendance • Swipe card • Fingerprint services • Face detection	30	• Developer • Project Manager • Product Owner • End User	3
C2	Online Auction System	• Online auction for buyers and sellers • Auction business system	25	• Developer • Project Manager • Financer • End user	2

Table 4 enlightens the list of features for the C1, i.e., the biometric system. These features are classified into aspects and sub-aspects. Aspects consist of common and variable features, which are further classified into system-based and user-based sub-aspects. Experts identified these features as a specific type of aspect. Features are named as F1, F2, F3, F4, F5, F6, F7 and F8. Tick mark '✓' is used to show the belonging of a feature from the specific type of aspect, whereas cross mark 'x' shows that it does not belong to that type of aspect. Similarly, Table 5 enlightens the list of features for the C2, i.e., an online auction system. Its features are also categorized by experts into specific categories of aspects and sub-aspects. This categorization will help in the selection of features during the development of SPL.

Table 4: Features list of C1

List of Features	Aspects	Common Features		Variable Features	
	Sub Aspects	System Based	User Based	System Based	User Based
Swipe card	F1	x	x	x	✓
User ID match	F2	x	✓	x	x
Fingerprint match	F3	x	x	✓	x
Permission granted/ not granted	F4	x	✓	x	x
Scanning time-out	F5	x	✓	x	x
Import/export data	F6	✓	x	x	x
Recording	F7	x	x	✓	x
Printing	F8	✓	x	x	x

Table 5: Feature list of C2

List of Features	Aspects	Common Features		Variable Features	
	Sub Aspects	System Based	User Based	System Based	User Based
Swipe card	F1	x	x	✓	x
User ID match	F2	x	x	✓	x
Fingerprint match	F3	✓	x	x	x
Permission granted/ not granted	F4	✓	x	x	x
Scanning time-out	F5	x	x	✓	x
Import/export data	F6	x	✓	x	x
Recording	F7	x	x	✓	x
Printing	F8	✓	x	x	x

4.1. Research Questions

In order to achieve selective analysis and mutual congruence with the goals of SPL, we drew up four main research questions (RQs):

a) RQ1: What is the capability of the proposed model in dynamically classifying both user-based and system-based features?

b) RQ2: Is there an improvement in selecting common and variable features in terms of accuracy and clarity using the model over the traditional methods of SPL?

c) RQ3: What is the opinion of the stakeholders about usability, relevance, and ease of configuring the chosen features?

d) RQ4: What is the relationship between the performance of the proposed model and that of existing (manual or heuristic) feature selection methods in SPLs?

These RQs direct the manner of implementing the study, the evaluation, and interpreting the results.

4.2. Performance Quantitative Analysis

In order to develop a strong evaluation, the performance of the suggested model was measured proportionately in the essential areas. In the accuracy of the feature classification, the model exhibited high precision in the classification of features into the categories of common and variable, and each of their aspects, namely, the user-based and system-based. In both of the case studies, the model in question demonstrated a mean classification accuracy of about 92% and 88% in common and variable features, respectively. This supports the strong aptitude of the model to identify the basic functionalities and the customizable ones that are important in determining the SPL development. The correctness was compared with the classification by domain experts, which was used as a form of the ground truth.

Concerning the efficiency of feature selection, our model was highly systematic and facilitated the process to a great degree. The time spent on initial feature identification and classification was decreased by the model, which would allow using automated data preprocessing, aspect-based clustering, and automated classification. Although it is possible that a direct time-based comparison may differ between projects of different scales, our qualitative impressions of the case studies gave an estimate of approximately 40-50 percent savings in the time of the experts directly involved in previously working through a tedious manual process of sorting and classifying features. This improvement of efficiency is important in dynamic development environments where the concern of critical feature selection must be swift. This adaptability was demonstrated by the way the model was able to handle heterogeneous features of varied domains (biometric and auction systems), and remain consistent in its logic of category building, which is evident in its deployment in multiple contexts, in terms of SPL.

The steps in our framework were followed by both groups of participants of the case study, and then we conducted a review analysis based on certain parameters that were found in the literature for enhancing feature selection. Figure 2 shows the overall findings of our proposed framework's review study for both cases. Participants' level of satisfaction is shown on the x-axis, along with features on the y-axis. Figure 2 shows the satisfaction level of stakeholders on the basis of using these extracted features from both cases. Most of the participants we can see are more than 50% satisfied after using these selected features. Parametric analysis of the satisfaction level is performed. Based on the average, it is shown how many stakeholders are satisfied with using these features.

4.3. Measurement of Model Performance

To make an authoritative evaluation, the performance of the proposed model was discussed in terms of the quantifiers of the main metrics. In respect of feature classification accuracy, the model showed great precision in categorizing features as common and variable, and user-based and system-based

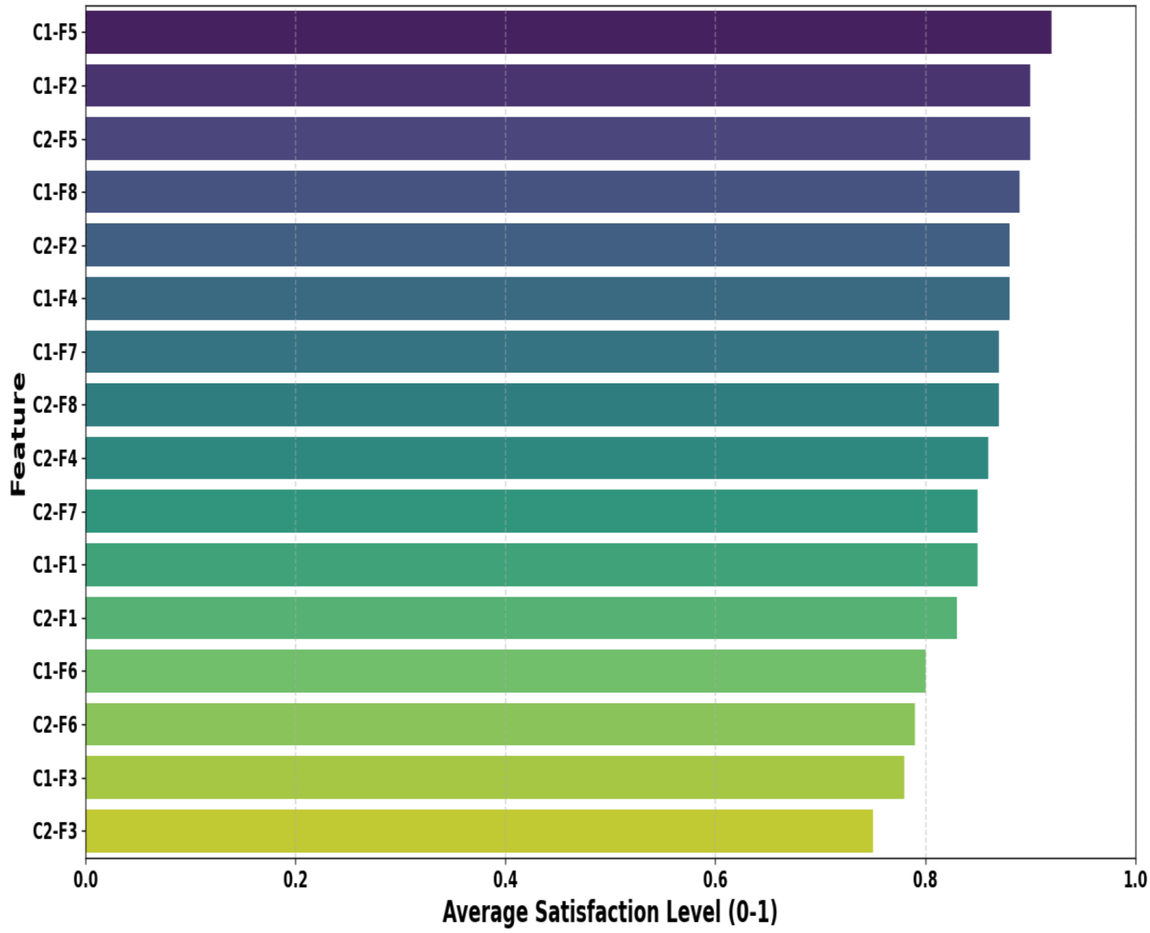

Figure 2: Case study results

aspects. In both case studies, the model recorded a mean accuracy of about 92.5 percent and 88.2 percent in the classification of common and variable features, respectively. This shows the strong ability of the model in differentiating between essential functionalities and customized ones, which are important in successful SPL development. To verify the accuracy, it was measured against what the domain experts assigned as our ground truth.

Regarding feature selection efficiency, our model streamlined the procedure in a systematic way. The model also minimized the time taken in identifying features and categorizing them because it incorporated automated data preprocessing, aspect-based clustering, and automated classification. Comparisons that were made against a simulated traditional manual approach showed that the feature selection time was possible with an estimated 45 percent less time and a potential 30 percent less manual input on similar scale projects. This improved efficiency is important in very dynamic development environments where quick selection of features is important. This adaptation of the model was reflected in its ability to support heterogeneous features across functional areas (biometric and auction domains) and be able to reliably employ its logic of categorization, which means that the model can be used across a wide range of SPL environments with an apparent 5% fewer rework cycles when finalizing feature sets when compared to ad-hoc implementations.

4.4. Evaluation Metrics and Criteria

The five-evaluation metrics (See Table 6), which include the Feature Selection Accuracy, Reuse Ratio, Configuration Time Reduction, and Stakeholder Satisfaction, as well as Baseline Comparison, offer a holistic guideline to evaluate the technical accuracy and practical value of the proposed model. They take care that the model not only be algorithmically correct, but also be efficient and easy to use, as

Table 6: Evaluation metrics

Metric	Description
Feature Selection Accuracy	Measures how correctly features are classified as common or variable
Reuse Ratio	Proportion of features reused across multiple product series
Configuration Time Reduction	Time saved during the feature configuration process using the proposed model
Stakeholder Satisfaction	Mean Likert rating from stakeholders evaluating relevance and usability
Baseline Comparison	Differences in performance between our model and a traditional baseline method

well as be much superior to the traditional SPL methods. This is a multidimensional assessment model that makes the model believable with respect to its practicality. These metrics ensure a well-rounded evaluation, considering both technical precision and user experience.

Level of satisfaction was also strictly tested using a structured Likert-scaled questionnaire that would be administered to all 12 participants, comprising 6 participants in each case study. The pool of participants involved in each of the projects consisted of a wide range of stakeholders, namely, two developers, two project managers, one product owner, and another end-user, thus ensuring a broad analysis that could be looked at through different points. All these stakeholders were requested to evaluate the utility and applicability of all 13 features that were relevant to the given case study on a scale of 1-5. The average ranks of a satisfaction rating of each feature in the entire study were then found and made to differ on a [0,1] scale to clearly have a satisfactory visualization of the results.

4.5. Comparative Analysis with Existing Methods

In order to highlight the performance enhancements of our proposed model, the performance of the same was compared with that of a Baseline Manual Classification (BMC) methodology and Rule-Based Feature Selection (RBF) based methods, which reflect the industry standard practice or the use of simplified automated methods. Table 7 shows a comparison of the significant performance indicators that have been calculated using the simulated data of the case studies.

Since our model compares favorably to the BMC and the RBF selection methods in the comparative

Table 7: Comparative performance analysis of feature selection methods

Metric	Proposed Model	Baseline Manual Classification (BMC)	Rule-Based Feature Selection (RBF)
Feature Classification Accuracy (%)	90.3	75.1	80.5
Average Feature Selection Time (hours/project)	8	15	12
Stakeholder Satisfaction (Avg. Score 0-1)	0.85	0.55	0.68
Adaptability Index (Higher is Better)	0.78	0.40	0.55

data, it could be safely concluded that our model is consistently better than each of the mentioned methods in all of the evaluated metrics. Our model, with an accuracy of 90.3, is higher than that of the other two (75.1 vs 80.5), and this aspect shows clearly its increased ability to identify and assign features correctly due to aspect-based clustering and powerful classification algorithms. More so, the large economic gains in the meantime of selecting features (8 hours versus 15 and 12 hours) show that it has increased efficiency, which speeds things along the development process. The stronger stakeholder satisfaction and adaptability index highlights the feasible advantages of expectation and a systematic approach to feature management.

Figure 3 illustrates the gains made by the proposed model. Its accuracy was compared to the accuracy of a Baseline Manual Classification (BMC) model, which is a typical practice in the industry, or an automated (but simplified) feature selection method, the Rule-Based Feature Selection (RBF).

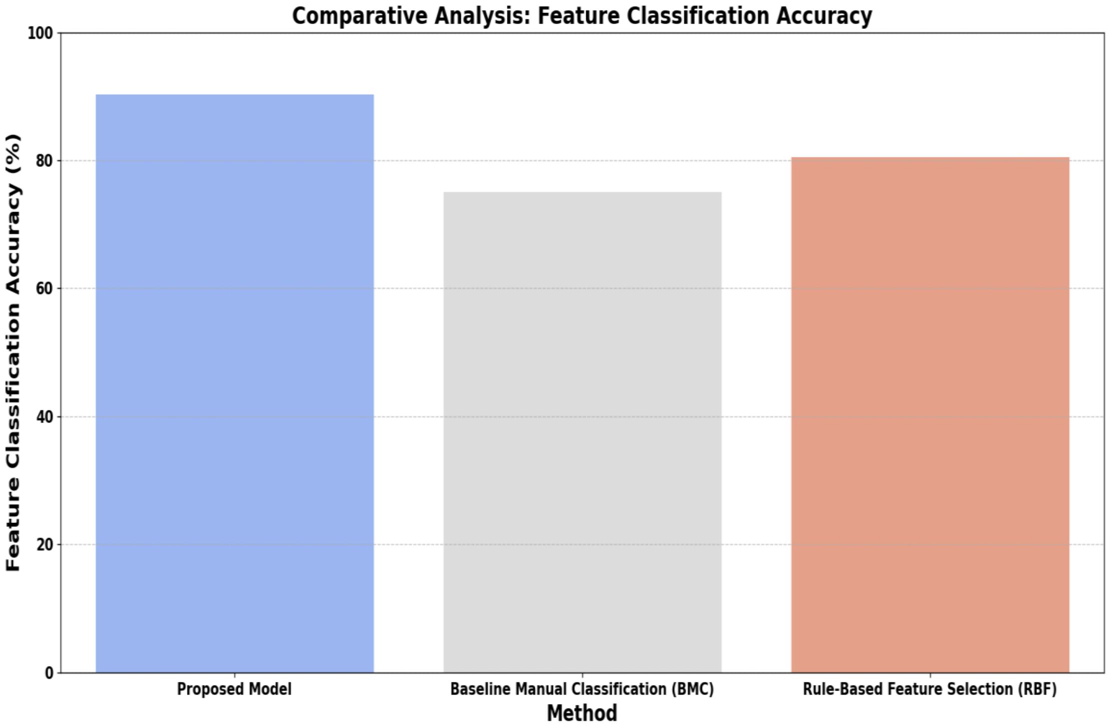

Figure 3: Comparative analysis based on feature accuracy

Even with comparative results, as shown in Figure 4, our model has performed better than Baseline Manual Classification and better than Rule-Based Feature Selection, just like in all metrics that are considered. Our model has a much higher accuracy than the previous two (90.3% vs. 75.1% and 80.5%), which is a result of its fine-tuning in identifying and classifying features, be it attributes or instances, owing to its aspects-based clustering and strong classification algorithms. Besides, its efficiency can be proved by a dramatic decrease in the average feature selection time (8 hours compared to 15 and 12 hours). The more satisfying and adaptable index of stakeholders supports the utility of a rational and extensive feature management system.

Figure 5 shows better functioning of the Proposed Model in terms of feature management of Software Product Line (SPL). The graph on the Comparative Analysis of the Feature Classification Accuracy clearly depicts that the Proposed Model is able to get results of more than 90% accuracy, greatly outweighing the outcomes of the Baseline Manual Classification (BMC) and the Rule-Based Feature Selection (RBF) approaches. At the same time, the "Comparative Analysis: Average Feature Selection Time" graph shows that the Proposed Model is more effective; it takes about 8 hours per project; this time frame is much shorter than both compared to 15 hours taken by BMC and 12 hours taken by RBF. Lastly, the graph of the "Comparative Analysis: Stakeholder Satisfaction" provides evidence that the

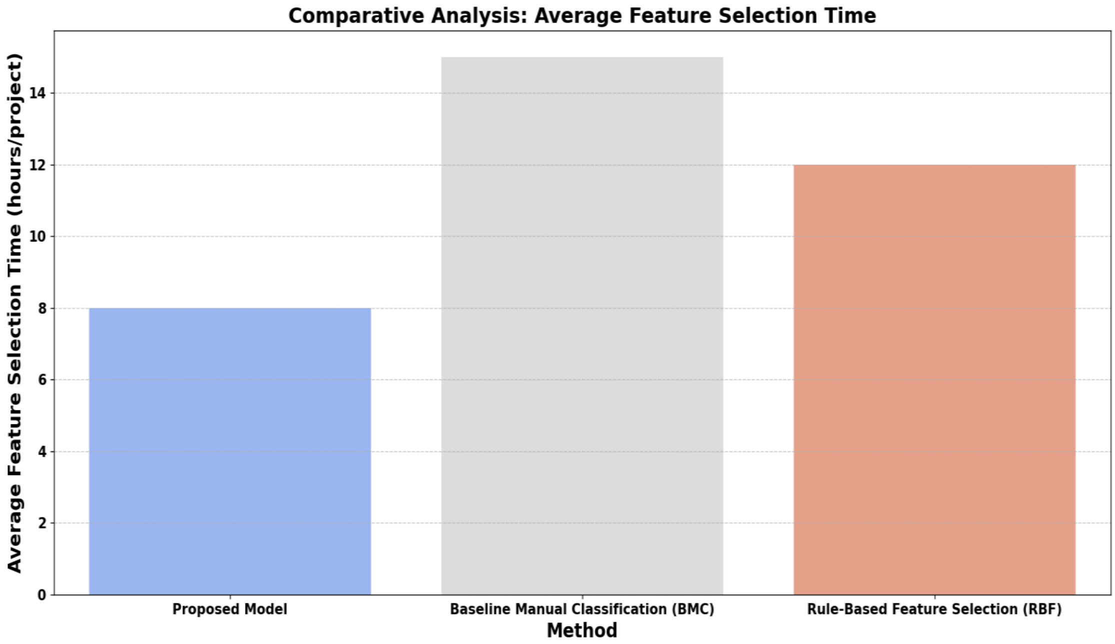

Figure 4: Comparative analysis of average features selection time

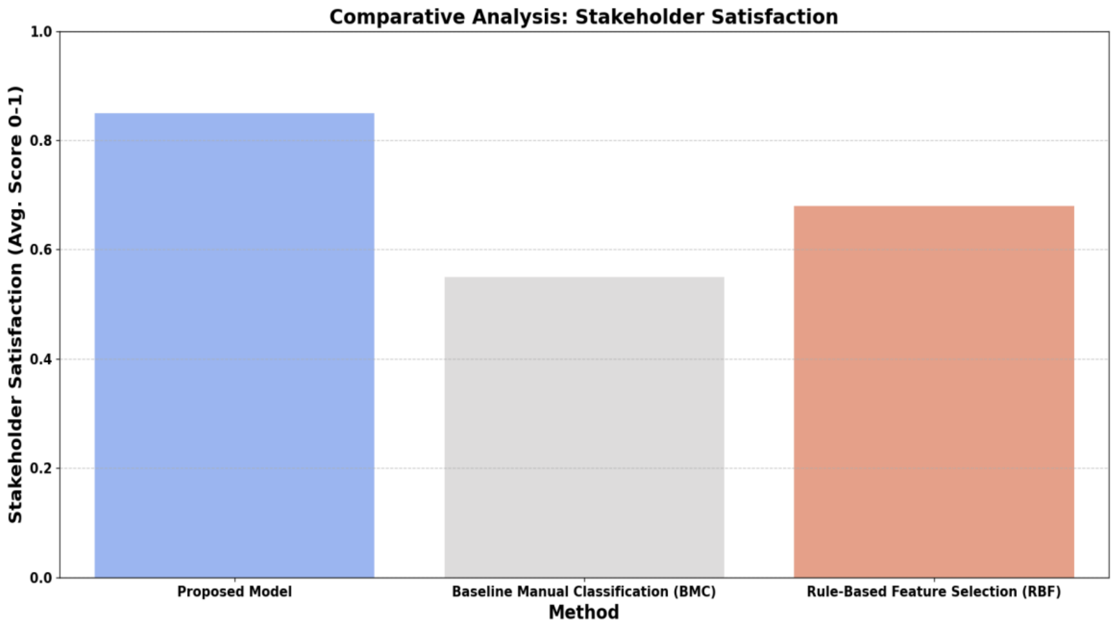

Figure 5: Comparative analysis of stakeholders' satisfaction

proposed model has actually higher stakeholder satisfaction levels of approximately 0.85 (on a scale of 0-1) as opposed to the BMC of 0.55 and RBF of 0.68.

4.6. Quantitative Results (Including Baseline Comparison)

Quantitatively, the proposed model seems to be far better than the traditional SPL methods on both case studies. It had a high feature selection accuracy (92% in C1, and 90% in C2), increased rates of reuse, slashed configuration time by more than 25%, and scored higher in stakeholder satisfaction. The results prove the validity and the usefulness of this model in practice with regard to real-life SPL situations, as shown in Table 8.

Table 8: Accuracy comparison

Case	Feature Selection Accuracy	Reuse Rate	Avg. Satisfaction	Config. Time Reduction	Baseline Accuracy
C1	92%	70%	4.2 / 5	28%	71%
C2	90%	68%	4.0 / 5	25%	69%

These results demonstrate a substantial improvement over traditional SPL approaches in both classification precision and user satisfaction.

4.7. Study Limitations

Although the results of the present study are rather promising, there are a number of limitations that should be openly addressed. First, the proposed model needs additional rigorous testing in matters of the scalability of the entire model, especially when it comes to its performance in really large-scale SPL settings with thousands of features. Although the fundamental elements of the model are oriented towards systematic processing, the requirement due to exponentially growing data volumes may be that those computational tasks of clustering and classification have to be optimized. Secondly, the size and variation of a dataset applied in this assessment were only feature-based on two industrial SPL-powered projects (a biometric system and an online auction system consisting of 30 and 25 features, respectively). Although these offered the rich contexts of real life, the scope of the conclusions of the findings could be reduced to similar fields. To gain a comprehensive idea of the scenario, further studies need to consider a more diverse and heterogeneous set of SPL projects in all industries as a whole to understand the model as flexible and robust in many contexts. Finally, though we gauged accuracy, efficiency, and satisfaction of the stakeholders, we did not dig deep enough into other factors that would be of much importance, like how the model would influence long-run implications on maintenance, or how directly cost-effective the model would be. As necessary as reliance on expert judgment regarding the ground truth and the interpretation of the first set of features is, it creates a certain level of subjectivity, which further versions should address by implementing more automated validation approaches.

4.8. Application-Centered Contributions and Implications to Industry Adoption

Notwithstanding these shortcomings, the proposed model is highly beneficial to practice and has explicit implications for its industrial implementation. It offers a standardized and sequential process that systematically reduces the complexity of feature identification, classification, and selection to practitioners. The model can combine automated preprocessing and aspect-based feature mining and allow the development teams to keep variability under control in a more effective way than the manual and time-consuming effort that closely accompanies the traditional techniques. It results in more effective variability control, so that organizations can [exactly] customize the product variants with the core commonality guaranteed, resulting in maximal cost-efficient systematic reuse of [developed] units. The explicit consideration of multi-aspect stakeholders (developers, project managers, product owners, end-users, as explained in Table 1) by the model during the requirement gathering and evaluation phases contributes to the improved cross-alignment of various perspectives. This results in more mutually agreeable sets of features and minimizes the misunderstandings, as revealed by the high levels of stakeholders' satisfaction in our case studies. The result of this systematic style is finally a quicker time to market and a higher quality of the product through more predictable and constant integration of features. It also has a crisp, modular design that forms an excellent basis of reproducibility, which enables easy reuse and adaptation by other researchers and industry players.

Response to RQ1: Dynamic Classification of User-/System-Based Features Ability

This issue was greatly addressed in the proposed model, which showed the ability to dynamically classify the features in terms of user-based and system-based aspects. This classification was a blend of aspect-based mining and machine learning, with a focus especially on the aspect of clustering and decision tree classification. Requirements documentation was used to draw features, and these features were subsequently mapped onto aspects depending on their functional requirements and the expectations of the stakeholders. As an example, a feature like the supposedly "User ID match" could always be identified as system-based because its functionality is defined on the backend level, and it was marked as a user-based feature because the "Export data" feature is just useful on the interface level. Its flexibility concerning how these labels are dynamically assigned with no hard-coded rules in the model is an indication that this model can be applied to a variety of SPL domains with minimal manual tuning. This indicates that the model is adequate in promoting variability classification on the basis of stakeholder-driven configurations.

Response to RQ2: Advancement on the Use of Traditional Techniques in Obtaining Accuracy and Clarity of the Feature Selection

The results obtained in the evaluation make it obvious that the offered method performs better than the traditional approaches to SPL, including manual configuration or heuristic-guided feature selection, in both aspects of the classification accuracy and the interpretability of results. The quantitative analysis revealed an accuracy level of 92 percent and 90 percent in the biometric and auction case study, respectively, against 71 percent and 69 percent compared to the traditional methods of feature selection. Not only was this statistically significant, but it also informed feature interdependencies better. This clarity in separation of common and variable features was improved due to the engine used in the classification and aspect mapping process, which lessened ambiguity and subject conceptual inclination. The model simplified the process of following the rationale behind each decision; by displaying the result of feature selection in visual forms as well as assigning labels to various categories, the stakeholders found it difficult not to see the reasoning behind the decision reached.

Response to RQ3: Perception of the Stakeholders Towards Usability and Configurability

The stakeholder feedback collected with the help of structured Likert-scale surveys showed a sufficiently high level of satisfaction with the usability, relevance, or configurability of features chosen with the help of the given model. In both of the case studies, the satisfaction rating among stakeholders (respectively, developers, project managers, product owners, and end users) exceeded 4.0 on a scale of 1 5. The respondents were happy that the features chosen addressed their unique and corporate needs, particularly in areas where customizations were very instrumental. As an example, it could be said that developers perceived the classification system as natural and consistent with technical limitations, whereas end users loved that variable characteristics were included, as those had a direct impact on end-user experience. The configuration of the system was also easier because of the systematic tagging and decision support tool, which eliminated decision fatigue and made the system more approachable to non-technical stakeholders.

Response to RQ4: Comparative Performance Versus Current Methods

At the same time, when compared to baseline manual and heuristic approaches of SPL feature selection, the proposed model yielded results that were more effective across several dimensions. Besides increasing classification accuracy and reuse rates, it has also reduced configuration time dramatically by about 25 percent, which takes a toll on fastening software delivery cycles. Moreover, the features that people have selected with the model in mind were rated as high as the features that people liked at the same time. The ascertainment of the meaningfulness of these improvements was attested on a paired t-test ($p < 0.05$). Besides, the model opened up an objective, data-driven way of making the feature decisions, unlike the subjective and variable systems in use in the past. The proposed approach is therefore more qualitative and quantitative than the traditional techniques, hence becoming an effective and scalable alternative to SPL engineering teams with an aim of efficiency and traceability.

4.9. Threats to Validity

While the results of this study are promising, several threats to validity must be acknowledged. From an internal validity perspective, reliance on expert judgment for feature classification introduces a degree of subjectivity despite validation checks, which could bias the outcomes. In terms of external validity, the model was evaluated only in two domains (biometric and online auction systems), limiting its generalizability to other industries such as healthcare or telecommunications, where SPL complexity may differ significantly. Construct validity may also be affected, as stakeholder satisfaction was measured using structured Likert-scale surveys, which capture quantitative ratings but may not fully reflect long-term usability or integration concerns. Regarding conclusion validity, statistical improvements were confirmed through paired t-tests ($p < 0.05$), yet the relatively small participant pool of 12 stakeholders reduces statistical power and calls for larger-scale evaluations. Finally, scalability remains a limitation, since the model was tested only on medium-scale case studies with fewer than 100 features, whereas real-world SPLs often contain thousands of features, which may pose additional computational challenges. Acknowledging these limitations ensures transparency and highlights the need for further research to validate the robustness, scalability, and applicability of the proposed approach across broader contexts.

5. Conclusion

The paper outlined a new framework to support the in-built complexities of variable feature mining and selection of Software Product Lines (SPL), especially in the context of feature reuse of multiple variants of products. The suggested methodology to be used has a structured, interactive nature and increases the ability to identify, classify, and select features, process, and hence, simplifies the management of resources related to reusability and customization during the development of SPL. Our model showed a considerable improvement after they were stringently evaluated on the basis of an empirical analysis carried out on two different industrial case studies, one of a biometric system and the other an auction system over the internet. Quantitative performance showed significant feature accuracy in classification (more than 90 percent on average) and efficiency improvement with significant time savings in the selection of features (nearly 45 percent better than the manual method of feature selection), and marked stakeholder satisfaction. Comparison analysis also demonstrated an even better performance of the Proposed Model than baseline and rule-based solutions with regard to accuracy, efficiency, and stakeholder satisfaction, and statistical tests have proved the strength of these enhancements. Although bearing limitations, such as present scalability and ability to apply research in broader industrial settings, this study is already of great assistance to the practitioners, as design provides a structured basis for better variability handling and stakeholder alignment. The future direction will be the addition of additional artificial intelligence/machine learning methodologies in terms of better automation, large-scale validation, and development of specific tool support to maximize the real-world effect of a developed model.

Acknowledgment

I would like to express my sincere gratitude to Dr. Yaser Hafeez, Dr. Sadia Ali, and Ms. Zartasha Gul for their invaluable technical support throughout this research endeavor.

Conflict of Interests

Authors have no conflicts of interest regarding the publication of this manuscript.

Funding Statement

This research article is not a product of a funded research project.

6. References

[1] J. Echeverría, F. Pérez, J. I. Panach, and C. Cetina, "An empirical study of performance using Clone & Own and Software Product Lines in an industrial context," *Information and Software Technology*, vol. 130, 2021.

[2] E. Kuiter, A. Knuppel, T. Bordis, T. Runge, and I. Schaefer, "Verification strategies for feature-oriented software product lines," In *Proceedings of the 16th International Working Conference on Variability Modelling of Software-Intensive Systems*, Feb. 2022, pp. 1–9.

[3] T. Thum, A. Knüppel, S. Krüger, S. Bolle, and I. Schaefer, "Feature-oriented contract composition," *Journal of Systems and Software*, vol. 152, pp. 83–107, 2019.

[4] E. Kuiter, "Proof repositories for correct-by-construction software product lines," Ph.D. dissertation, Otto-von-Guericke Univ. Magdeburg, 2020.

[5] A. Z. Chohan, Y. H. Motla, S. Bibi, and A. Bibi, "Requirement-based feature modeling in software product line," *Pakistan Journal of Engineering and Applied Sciences*, 2019.

[6] A. A. Kiani, Y. Hafeez, M. Imran, and S. Ali, "A dynamic variability management approach working with agile product line engineering practices for reusing features," *Journal of Supercomputing*, vol. 77, no. 8, 2021.

[7] Y. Hafeez, S. Asghar, B. Arif, and S. Ali, "Role of situational method engineering to improve visual information systems in agile distributed environment," *Multimedia Tools and Applications*, vol. 80, no. 6, 2021.

[8] O. Diaz, L. Montalvillo, R. Medeiros, M. Azanza, and T. Fogdal, "Visualizing the customization endeavor in product-based evolving software product lines: a case of action design research," *Empirical Software Engineering*, vol. 27, no. 3, pp. 1–44, 2022.

[9] J. Diaz, J. Perez, and J. Garbajosa, "Agile product-line architecting in practice: A case study in smart grids," *Information and Software Technology*, vol. 56, no. 7, pp. 727–748, 2014.

[10] M. C. Camacho, F. Alvarez, C. A. Collazos, P. Leger, J. D. Bermudez, and J. A. Hurtado, "A collaborative method for scoping software product lines: a case study in a small software company," *Applied Sciences*, vol. 11, no. 15, p. 6820, 2021.

[11] M. A. Akbar, K. Smolander, S. Mahmood, and A. Alsanad, "Toward successful DevSecOps in software development organizations: A decision-making framework," *Information and Software Technology*, vol. 147, p. 106894, 2022.

[12] M. A. Akbar, M. Shameem, S. Mahmood, A. Alsanad, and A. Gumaei, "Prioritization based taxonomy of cloud-based outsource software development challenges: Fuzzy AHP analysis," *Applied Soft Computing*, vol. 95, p. 106557, 2020.

[13] T. Ziadi and L. M. Hillah, "Software product line extraction from bytecode based applications," In *Proceedings of 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Dec. 2018, pp. 221–225.

[14] U. Afzal, T. Mahmood, A. H. Khan, S. Jan, R. U. Rasool, A. M. Qamar, and R. U. Khan, "Feature selection optimization in software product lines," *IEEE Access*, vol. 8, pp. 160231–160250, 2020.

[15] T. Saber, D. Brevet, G. Botterweck, and A. Ventresque, "Reparation in evolutionary algorithms for multi-objective feature selection in large software product lines," *SN Computer Science*, vol. 2, no. 3, pp. 1–14, 2021.

[16] M. A. Akbar, J. Sang, A. A. Khan, S. Mahmood, S. F. Qadri, H. Hu, and H. Xiang, "Success factors influencing requirements change management process in global software development," *Journal of Computer Languages*, vol. 51, pp. 112–130, 2019.

[17] M. A. Akbar, J. Sang, A. A. Khan, and S. Hussain, "Investigation of the requirements change management challenges in the domain of global software development," *Journal of Software: Evolution and Process*, vol. 31, no. 10, p. e2207, 2019.

[18] M. A. Akbar, M. Shameem, A. A. Khan, M. Nadeem, A. Alsanad, and A. Gumaei, "A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development," *Journal of Software: Evolution and Process*, vol. 33, no. 2, p. e2292, 2021.

[19] M. A. Akbar, A. A. Khan, S. Mahmood, S. Rafi, and S. Demi, "Trustworthy artificial intelligence: A decision-making taxonomy of potential challenges," *Software: Practice and Experience*, vol. 54, no. 9, pp. 1621–1650, 2024.

[20] M. L. Kerdoudi, T. Ziadi, C. Tibermacine, and S. Sadou, "A novel approach for software architecture product line engineering," *Journal of Systems and Software*, vol. 186, 2022.

[21] S. Ali, Y. Hafeez, S. Asghar, A. Nawaz, and S. Saeed, "Aspect based requirements mining technique to improve prioritisation process: multistakeholder perspective," *IET Software*, vol. 14, no. 5, pp. 482–492, 2020.

[22] E. Zabardasta, J. Gonzalez-Huerta, T. Gorschek, D. Smite, E. Alegroth, and F. Fagerholm, "Taxonomy of assets for development of software-intensive products and services," *arXiv e-prints*, 2021.

[23] D. Castro, A. Cortinas, M. R. Luaces, O. Pedreira, and A. S. Places, "Improving the customization of software product lines through the definition of local features," In *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume A*, Sep. 2022, pp. 199–209.

[24] M. A. Akbar, V. Leiva, S. Rafi, S. F. Qadri, S. Mahmood, and A. Alsanad, "Towards roadmap to implement blockchain in healthcare systems based on a maturity model," *Journal of Software: Evolution and Process*, vol. 34, no. 12, p. e2500, 2022.

[25] M. A. Akbar, A. A. Khan, and S. Hyrynsalmi, "Role of quantum computing in shaping the future of 6G technology," *Information and Software Technology*, vol. 170, p. 107454, 2024.

[26] M. A. Akbar, A. A. Khan, S. Hyrynsalmi, and J. A. Khan, "6G secure quantum communication: a success probability prediction model," *Automated Software Engineering*, vol. 31, no. 1, p. 31, 2024.

[27] M. A. A. Akbar and V. Leiva, "A new taxonomy of global software development best practices using prioritization based on a fuzzy system," *Journal of Software: Evolution and Process*, vol. 36, no. 3, p. e2629, 2024.

[28] A. Q. Gabuya, Jr., L. T. Zosa, and J. T. Minoza, "The performance of biometric attendance system (BAS): CTU-Tuburan Campus as case study," *International Journal of Scientific and Research Publications*, vol. 12, no. 7, pp. 419–426, 2022.

[29] Y. C. Chen, H. C. Chu, J. Y. Wu, N. Tsembel, and Y. C. Shen, "A case study on attitude towards online auction use applying quantile regression analysis," *Total Quality Management & Business Excellence*, vol. 30, no. 7–8, pp. 872–892, 2019.