International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 4, 1487-1511, 2026
https://doi.org/10.33889/IJMEMS.2026.11.4.061

# A Multi-Criteria Framework for Software Product Evaluation and Ranking Using Fuzzy d-Sets

**Nidhi Kadam**
Decision Sciences Area,
Indian Institute of Management Lucknow, Lucknow, Uttar Pradesh, India.
*Corresponding author*: nidhikadam@yahoo.co.in

**Garima Mittal**
Decision Sciences Area,
Indian Institute of Management Lucknow, Lucknow, Uttar Pradesh, India.
E-mail: garima@iiml.ac.in

(Received on January 23, 2026; Revised on May 17, 2026 & June 15, 2026 & June 24, 2026; Accepted on July 9, 2026)

**Abstract**
Most Information Technology (IT) organizations periodically review their products and services to aid systematic decision-making. Such decisions must balance budgetary and non-budgetary constraints to allocate resources effectively in the short and long term. While new IT products offer limited quantitative data, ongoing offerings yield substantial performance related information, which is used in this study to develop a framework for assessing existing Open-Source Software (OSS) products. As such data is dynamic and uncertain, we employ Interval-Valued Intuitionistic Fuzzy Parameterized Interval-Valued Intuitionistic Fuzzy Soft Sets (IVIFP-IVIFSS or d-sets) to model dual uncertainty and propose a maintenance-oriented decision-support framework. The study assesses OSS products under two complementary decision-making contexts: (i) unified long-term product ranking using d-sets and (ii) temporal short-term consensus ranking using d-sets and Borda aggregation. While the former context captures persistent maintenance-performance signals, the latter reveals localized operational dynamics relevant for interim decision-making. The study illustrates the effectiveness of the proposed ranking methodology through a case study using data from 16 OSS products in the Bugzilla repository, evaluated using four parameters. The framework is methodologically complex, but the resulting rankings are intuitive and actionable. In addition, a perturbation analysis was conducted with the Interval-Valued Intuitionistic Fuzzy Parameters (IVIFP) varied by ±5%, ±10% and ±15% to assess the robustness of the resulting d-set rankings. The results show that the proposed framework produces stable and interpretable rankings under dynamic and uncertain conditions. These findings indicate that d-sets are useful for uncertainty-aware performance evaluation and portfolio-level decision-making for IT products and services.

**Keywords-** Interval-valued intuitionistic fuzzy parameterized interval-valued intuitionistic fuzzy soft sets, d-sets, Open-source software, Performance evaluation, IT product ranking, Multi-criteria decision-making.

## 1. Introduction

Performance-based value assignment is widely applied across domains such as education (student assessment and instructional effectiveness), economics (value creation and productivity measurement), management (organizational, project, product and team evaluation), healthcare (treatment effectiveness), finance (investment valuation and risk assessment), public-sector governance (institutional and facility evaluation) and IT (software project evaluation and portfolio selection) (Amer et al., 2022; Carvalhosa & Lucas, 2026; Cosa & Torelli, 2024; de Araújo et al., 2024; Do et al., 2025; Klassen et al., 2010; Moran et al., 2019; Sahlin & Angelis, 2019; Saxena et al., 2022; Schleicher et al., 2018; Spitzer et al., 2023). Across these areas, decision-makers use performance parameters derived from historical data to guide future actions and resource allocation. Accordingly, a substantial body of the literature has examined the evaluation and improvement of performance management systems across sectors (de Melo Santos et al., 2025). Uncertainty in performance evaluation has been addressed using fuzzy-logic-based and data-driven approaches (Cakir & Adiguzel, 2020; Moran et al., 2019). Model-based fuzzy approaches offer flexibility for handling vague and imprecise information. In contrast, model-free approaches use historical or observed data for predictive modeling using artificial intelligence and genetic algorithms. These methodological paradigms have been employed in a broad spectrum of real-world problems such as task credit assignment (Shahar et al., 2019), programming feedback based on dynamic analysis (Gulwani et al., 2014), international assignment decision-making (Renshaw et al., 2021), student performance assessment in online learning environments using logistic mixed models (Spitzer et al., 2023), patient-physician assignment and outcome evaluation (Love et al., 2024), incentive design through field experiments (Delfgaauw et al., 2017), organizational credit and blame attribution using factorial designs (Crant & Bateman, 1993), and project and managerial performance evaluation using data envelopment analysis (Xu & Yeh, 2014).


**Figure 1.** Software Development Life Cycle (SDLC).

In the domain of IT, the Software Development Life Cycle (SDLC), as shown in **Figure 1**, comprises planning, requirements analysis, architecture design, implementation, testing, deployment and an extended maintenance period. The early phases typically last weeks or months, but maintenance may continue for years, including ongoing corrective and adaptive interventions. Thus, performance evaluation becomes particularly important during the software maintenance phase. In this phase, users report deviations from expected behavior or suggest enhancements. These are commonly known as software bugs and are logged systematically and managed via bug-tracking platforms such as Atlassian (2026), Bugzilla (2026).

Bugs move through a defined life cycle as illustrated in **Figure 2,** including submission (unconfirmed state), validation (new state), assignment, resolution, verification (closure) and potential reopening. Bug-report data generated during maintenance provides a rich basis for assessing software performance. Maintenance-driven evaluation considers both demand-side pressures (incoming bugs) and supply-side effectiveness (bug resolution), offering a more balanced and complete view of product health. However, bug life cycle data are uncertain due to (i) inconsistent reporting practices, (ii) variability of user behavior, (iii) incomplete or delayed updates and (iv) differences in developer response patterns over time. Ranking software products based on bug life cycle indicators helps decision-makers make informed decisions on software selection, adoption and resource allocation in OSS ecosystems. Furthermore, maintenance-based rankings bring out time-sensitive insights that enable decision-makers to track performance trends and identify early warning signs of declining quality in such environments.

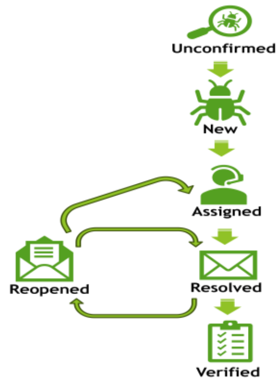


**Figure 2.** Typical bug life cycle.

Existing studies on OSS have primarily focused on developer productivity, defect prediction, bug triaging, or software evolution. The literature on IT product evaluation does not adequately address the comparative ranking of products or services based on maintenance-centric assessment under temporal uncertainty affecting both parameters and alternatives. To address these limitations, the present study investigates OSS maintenance evaluation from a product-level decision-making perspective under dynamic and uncertain maintenance conditions. The study seeks to address the following research questions (RQs):

**RQ1:** Which maintenance-centric parameters derived from bug life cycle data are most relevant for evaluating OSS product performance under uncertainty?

**RQ2:** How can simultaneous uncertainty in performance indicators and OSS products be modeled to derive robust comparative rankings?

**RQ3:** Can the proposed fuzzy decision-making framework support both long-term maintenance evaluation and short-term temporal performance analysis?

To address these research questions, this study identifies maintenance-oriented indicators from bug life cycle information, applies a fuzzy multi-criteria framework to investigate product ranking under dual uncertainty, and extends the proposed fuzzy framework across two complementary decision contexts: unified long-term evaluation and temporal short-term consensus analysis. It will be worth mentioning that the framework supports modeling uncertainty in both the parameters (performance indicators or criteria) and the alternatives (the products under study). To ensure empirical relevance and capture maintenance dynamics, the framework is applied to real bug-report data from 16 OSS products maintained using Bugzilla. This evaluation method addresses the limitations related to interpretability, utility and fairness reported in prior performance management studies (de Araújo et al., 2024). Specifically, to assist in decision-making under uncertainty, we employ Interval-Valued Intuitionistic Fuzzy Parameterized Interval-Valued Intuitionistic Fuzzy Soft Sets (IVIFP-IVIFSS), also referred to as d-sets (Aydın & Enginoğlu, 2021) to aggregate heterogeneous maintenance indicators and obtain comparative rankings.

The remainder of the paper is organized as follows. Section 2 presents the background and related work. The section presents a chronological development of fuzzy decision-making techniques. The section also presents the d-set methodology and its utility over other multi-criteria decision-making techniques for handling the dual uncertainty in the context of OSS products. Section 3 presents the OSS maintenance performance evaluation problem under uncertainty. Section 4 presents the technical details of the d-set methodology. These details are then summarized with the help of a flowchart. In Section 5, we present an empirical case study using OSS bug life cycle data. First, d-set-based unified product rankings are obtained, followed by temporal ranking analysis across multiple decision contexts, robustness analysis, results discussion, and a visual summary of the framework. The managerial implications of the study are discussed in Section 6. Section 7 presents the conclusions, outlines the study's limitations, and suggests future research directions.

## 2. Background and Related Work

Longitudinal studies of OSS indicate that historical trends in software metrics and quality attributes influence long-term system performance, indicating the importance of reliable indicators during the maintenance phase (Gradišnik et al., 2020). While version history primarily documents code evolution across releases, maintenance-related metrics more directly reflect the effort, resources and coordination involved in sustaining software systems and therefore offer a clearer proxy for team-level performance. Molnar & Motogna (2020) conducted a longitudinal analysis spanning eighteen years of software releases to assess long-term evolution patterns. In contrast, the present study uses recent, maintenance-oriented, latency-based bug life cycle information to evaluate software-level maintenance performance under evolving operational conditions, thereby reducing the uncertainty associated with long-term structural variability and historical evolution. Studies on OSS in literature have focused mainly on longitudinal software evolution, developer-level analytics, defect prediction, and project-specific maintenance analysis. However, the comparative ranking of OSS products based on maintenance-performance behavior remains largely unexplored. Furthermore, comparative evaluation under simultaneous uncertainty in both decision parameters and alternatives has received limited attention in fuzzy decision-making applications, mainly under temporal and longitudinal maintenance settings. These limitations form the motivation for the development of an uncertainty-aware decision framework that enables both long-term maintenance assessment and temporal performance analysis in OSS environments. The following subsections review the literature related to advancements in fuzzy decision-making techniques, d-set methodology and its applicability over other related Multi-Criteria Decision-Making (MCDM) methods.

### 2.1 Fuzzy Decision-Making

Classical approaches generally rely on crisp binary logic, which has limited ability to describe gradual transitions, partial truths, or the ambiguous information encountered in real-world decision problems. Fuzzy decision-making overcomes the limitations of traditional deterministic decision models by incorporating imprecision, uncertainty and subjective judgment. Fuzzy logic describes such phenomena using continuous membership functions, resulting in smoother transitions between states and a more realistic description of complex systems. This characteristic makes fuzzy decision-making appropriate for multi-criteria environments where inputs differ in reliability, scale, or completeness. By allowing uncertainty to be addressed in evaluation and aggregation processes, fuzzy approaches support more dependable optimization and ranking. Consequently, fuzzy decision-making frameworks are useful in practical decision-support applications in domains where exact quantification is infeasible, or data are incomplete or evolving.

Fuzzy sets, introduced by Zadeh (1965), enabled reasoning under uncertainty through membership functions. Several extensions were later proposed to address specific decision-making requirements. Soft sets (Molodtsov, 1999) manage uncertainty arising from multiple parameters without requiring membership functions, while rough sets (Pawlak, 1982) use lower and upper approximations. Vague sets (Gau & Buehrer, 1993) and Intuitionistic Fuzzy Sets (IFS) (Atanassov, 1986) further extended uncertainty modeling by introducing degrees of non-membership and indeterminacy.

Hybrid extensions of fuzzy sets have been proposed to increase flexibility and to address complexity in practical decision-making problems. Interval-Valued Fuzzy Sets (IVFS) assign intervals to membership degrees, allowing a range of possible values rather than a single point estimate (Grattan-Guinness, 1976). The Interval-Valued Intuitionistic Fuzzy Set (IVIFS) further extends this concept through representation of the degrees of membership, non-membership and indeterminacy as intervals (Atanassov & Gargov, 1989).

To address more complex forms of uncertainty, hybrid models combining fuzzy, intuitionistic, interval-valued and soft-set representations were developed. The Interval-Valued Intuitionistic Fuzzy Soft Set (IVIFSS), introduced by Jiang et al. (2010) and the Interval-Valued Intuitionistic Fuzzy Parameterized Soft Set (IVIFPSS) (Deli & Karataş, 2016) are examples of such extensions. Recently, Enginoğlu & Memiş (2020) applied fuzzy MCDM techniques, including criteria-weighted fuzzy soft min-max methods, to image denoising, illustrating the versatility of these approaches. These methods have been applied across a wide range of decision-making domains, addressing varying levels of uncertainty and vagueness in real-world scenarios.

## 2.2 d-Sets and Dual-Intuition Decision Framework

Dual-intuition-based methods enable modeling of uncertainty simultaneously in both parameters and alternatives. IVIFP-IVIFSS or d-sets, proposed by Aydın & Enginoğlu (2021), constitute one such model. In d-sets, both parameters and alternatives are represented using interval-valued-intuitionistic fuzzy information, making them suitable for soft decision-making problems involving vague and uncertain data. Randomness and ambiguity are two of the most common types of uncertainty. Since its proposal, fuzzy set theory has achieved profound success in dealing with uncertainty. Advanced intuitionistic fuzzy sets, such as d-sets, can capture real-world ambiguity more precisely and are more flexible and practical than traditional fuzzy set theory (Zhang et al., 2026). d-sets are the most comprehensive extension suited for modeling dual uncertainty through interval-valued degrees of membership, non-membership and indeterminacy (Memiş, 2023). However, the application of d-sets in IT remains limited, especially in the comparative evaluation and ranking of OSS products, where no prior study has employed this framework in the temporal and longitudinal maintenance settings considered in this study.

OSS performance data exhibit characteristics that align well with the assumptions underlying the d-set methodology. In practice, these data are high-volume, noisy, time-dependent, incomplete and rarely exhibit deterministic or strictly sequential patterns. Moreover, historical maintenance observations do not always suggest reliable predictors of future behavior because of evolving system architectures, developer dynamics and usage contexts. By modeling uncertainty in both parameters and alternatives, d-sets accommodate evolving and incomplete information without imposing restrictive temporal assumptions. These characteristics make the d-sets particularly suitable for performance evaluation in environments characterized by uncertainty, complexity and multidimensionality. Although many studies have examined fuzzy decision-making and multi-criteria performance evaluation, the systematic use of maintenance-centric indicators derived from bug life cycle data to evaluate and rank OSS systems remains limited. To the best of our knowledge, existing research has not integrated such indicators into a fuzzy decision-making-driven performance assessment frameworks. While OSS has been examined in relation to developer productivity, bug prediction and automated bug assignments, the present work focuses exclusively on product-level performance evaluation.

**Table 1** presents a chronological overview of major fuzzy set generalizations and hybrid extensions, including d-sets. By enabling both parameters (bug-related attributes) and alternatives (OSS projects) to be modeled with greater fidelity, d-sets provide a flexible basis for performance-based value assignments during software maintenance. This unique capability is critical for OSS evaluation, where bug-related metrics are often incomplete, time-dependent and noisy. Other fuzzy or hybrid soft-set models cannot adequately represent this multidimensional uncertainty.

**Table 1.** Chronological overview of fuzzy decision-making methods and their extensions.

| Fuzzy method | Category | Key contribution | Applications |
|---|---|---|---|
| Zadeh Fuzzy Sets (ZFS) (Zadeh, 1965). | Generalization | Single membership function. | Classification, control systems, AI. |
| L-Fuzzy Sets (Goguen, 1967). | Generalization | Membership over lattices/posets. | Algebraic fuzzy modelling. |
| Fuzzy Linguistic Term Sets (FLTS) (Zadeh, 1975). | Generalization | Linguistic membership terms. | Human-centric decision making. |
| Interval-Valued Fuzzy Sets (IVFS) (Grattan-Guinness, 1976) | Hybrid | Interval-valued membership. | Control systems, pattern recognition. |
| Rough Sets (Pawlak, 1982). | Generalization | Lower and upper approximations without membership. | Data mining, Decision-support System. |
| Intuitionistic Fuzzy Sets (IFS) (Atanassov, 1986). | Generalization | Membership and non-membership. | Decision-making, risk analysis. |
| Interval-Valued Intuitionistic Fuzzy Sets (IVIFS) (Atanassov & Gargov, 1989). | Hybrid | Interval membership and non-membership. | MCDM, reliability analysis. |
| Vague Sets (Gau & Buehrer, 1993). | Generalization | Interval truth and falsity. | Evaluation systems. |
| Soft Sets (SS) (Molodtsov, 1999). | Generalization | Parameter-based representation without membership. | Decision-support system. |
| Fuzzy Soft Sets (FSS) (Maji et al., 2002). | Hybrid | Fuzzy membership on parameters. | Medical diagnosis. |
| Fuzzy Parameterized Fuzzy Soft Sets (FPFSS) (Çağman et al., 2010). | Hybrid | Fuzzy parameters and fuzzy sets. | MCDM. |
| Interval-Valued Intuitionistic Fuzzy Soft Sets (IVIFSS) (Jiang et al., 2010). | Hybrid | Interval-valued intuition for alternatives. | Decision evaluation. |
| Probabilistic Soft Sets (Zhu & Wen, 2010). | Hybrid | Probability-based association. | Decision analysis. |
| Fuzzy Parameterized Interval-Valued Fuzzy Soft Sets (FPIVFSS) (Alkhazaleh et al., 2011a). | Hybrid Dual | Interval-valued fuzzy parameters. | Uncertain decision-support systems. |
| Fuzzy Parameterized Soft Sets (FPSS) (Çağman et al., 2011). | Hybrid | Fuzzy parameters. | MCDM. |
| Hesitant Fuzzy Sets (HFS) (Torra, 2010). | Hybrid | Multiple membership values. | Group decision-making. |
| Multi-Fuzzy Sets (MFS) (Sebastian & Ramakrishnan, 2011). | Hybrid | Sequence of membership functions. | Advanced fuzzy modelling. |
| Possibility Fuzzy Soft Sets (Alkhazaleh et al., 2011b). | Hybrid | Possibility and fuzzy membership. | Medical diagnosis. |
| Soft Expert Sets (Alkhazaleh & Salleh, 2011). | Hybrid | Multiple experts. | Expert-based decision-making. |
| Soft Multiset Theory (Alkhazaleh et al., 2011c). | Hybrid | Multiple subsets. | Complex decision-support systems. |
| Bipolar Fuzzy Parameterized Soft Sets (BFPSS) (Abdullah et al., 2014). | Hybrid | Positive and negative memberships. | Medical diagnosis. |
| Intuitionistic Fuzzy Parameterized Soft Sets (IFPSS) (Deli & Çağman, 2015). | Hybrid Dual | Intuitionistic fuzzy parameters. | Decision-making. |
| Multi-Interval-Valued Fuzzy Soft Sets (Alkhazaleh, 2015). | Hybrid | Multiple interval memberships. | Uncertain decision-support systems. |
| Interval-Valued Intuitionistic Fuzzy Parameterized Soft Sets (IVIFPSS) (Deli & Karataş, 2016). | Hybrid Dual | Interval intuition on parameters. | Decision-making under uncertainty. |
| Hypersoft Sets (Smarandache, 2018). | Hybrid | Sub-attribute parameters. | Multi-attribute decision-making (MADM). |
| Interval-Valued Fuzzy Adaptive Ratio Assignment (IVF-ARA) (Dahooie et al., 2018). | Hybrid | Interval-valued fuzzy multi-attribute evaluation. | Construction and civil project evaluation. |
| Intuitionistic Fuzzy Parameterized Intuitionistic Fuzzy Soft Sets (IFPIFS-Sets) (Enginoğlu & Arslan, 2019). | Hybrid Dual | Dual intuitionistic fuzziness. | Soft decision-making. |
| Fuzzy Parameterized Intuitionistic Fuzzy Soft Sets (Sulukan et al., 2019). | Hybrid Dual | Parameterized intuitionistic fuzziness. | Performance evaluation. |
| d-sets (IVIFP-IVIFSS) (Aydın & Enginoğlu, 2021). | Hybrid Dual | Interval-valued intuition in parameters and alternatives. | Performance-based ranking, soft decision-making. |
| Effective Fuzzy Soft Sets (EFSS) (Alkhazaleh, 2022). | Hybrid | Efficiency-oriented fuzzy modelling. | Medical decision-making. |
| Picture Fuzzy Parameterized Picture Fuzzy Soft Sets (PFPPFS) (Memiş, 2023). | Hybrid | Picture fuzzy MCDM. | Image noise removal, construction and civil project evaluation. |

### 2.3 Applicability of d-Sets Over Other MCDM Methods

We compare widely used MCDM techniques in the literature for their ability to handle fuzzy uncertainty, indeterminacy and temporal robustness. The comparison focuses on OSS product ranking and maintenance evaluation, where decision inputs are often noisy, incomplete and evolving. Classical MCDM methods such as Analytic Hierarchy Process (AHP) (Saaty, 1980), Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS) (Hwang & Yoon, 1981), and Data Envelopment Analysis (DEA) (Charnes et al., 1978) use crisp numerical inputs or deterministic evaluation structures. AHP is based on pairwise comparisons and consistency verification; TOPSIS evaluates alternatives using geometric distances from the ideal and anti-ideal solutions; and DEA measures relative efficiency without modeling uncertainty or indeterminacy. The applicability of these methods is limited for product ranking in complex OSS environments, as they do not support fuzzy uncertainty. Extensions of these methods have been developed to cater to fuzzy uncertainty. For example, fuzzy AHP (Chang, 1996; Zadeh, 1965) and fuzzy TOPSIS (Chen, 2000) use linguistic and fuzzy representations to handle imprecise judgments. However, these methods increase computational complexity due to pairwise comparisons, normalization, defuzzification and exhibit sensitivity to weighting and scaling schemes (Saxena et al., 2022; Singh et al., 2023). Note that OSS product ranking is characterized by dual uncertainty, both at the parameter level and at the level of alternatives. Hybrid approaches such as fuzzy AHP-TOPSIS increase flexibility but do not fully address the propagation of uncertainty across parameters and alternatives (Do et al., 2025). The d-set methodology adopted in the present study, combines IVIFS with soft set theory, enabling simultaneous representation of membership, non-membership and indeterminacy. d-set intervals preserve uncertainty across the decision space without information loss (Aydın & Enginoğlu, 2021). Unlike conventional fuzzy MCDM approaches, d-sets support the aggregation of heterogeneous and parameterized information, making them more suitable for OSS environments where evaluation attributes, such as the number of bugs in various statuses, are inconsistent, volatile and time-dependent. Further, d-sets provide a unified mechanism to handle both parameter-level and alternative-level uncertainty simultaneously. This is especially relevant for OSS product ranking, where evaluation dimensions are dynamic. Existing empirical studies indicate that interval-based soft computing frameworks produce more stable and expert-aligned rankings under high uncertainty conditions (Aydın & Enginoğlu, 2021). Overall, the d-set methodology gives a structured well suited for OSS product ranking and maintenance evaluation under fuzzy, incomplete and dynamic information environments.

### 2.4 Research Gap and Contribution of the Present Study

Although OSS maintenance datasets are widely available through bug-tracking repositories, prior studies have largely focused on developer-level analysis, defect prediction, bug triaging, and project-specific maintenance analytics, while the ranking of OSS products based on maintenance behavior remains largely unexplored. In addition, despite the growing adoption of fuzzy decision-making approaches, the application of the d-set methodology to establish OSS product rankings has not been previously investigated. Existing d-set studies have predominantly considered static evaluation settings, whereas OSS maintenance data are temporal, evolutionary and longitudinal. Unlike prior studies, the present work brings together maintenance-oriented bug life cycle dynamics with dual-uncertainty modeling under both long-term strategic assessment and short-term temporal analysis, thereby enabling a broader and more realistic evaluation of OSS products.

## 3. Problem Statement

- This study addresses the evaluation and ranking of maintenance performance of long-running OSS products using bug life cycle indicators extracted from historical bug repository data.
- The decision space consists of multiple software products (alternatives) such as Bugzilla, Mozilla QA, GeckoView, NSS and WebExtensions, each exhibiting distinct and evolving maintenance characteristics

over time. These products are continuously updated through bug reporting and resolution activities, resulting in heterogeneous maintenance patterns across the system.

- In this study, the evaluation of the OSS products is performed using multiple maintenance-related parameters derived from bug life cycle states, namely: New (N), Fixed (F), Assigned (A) and Reopened (R) (refer to **Figure 2**). These parameters capture different aspects of maintenance effort, resolution efficiency, backlog evolution and regression behavior across products. Since these indicators are extracted from real-world bug repositories, they are typically dynamic, incomplete and noisy.
- The older information regarding bugs may become unavailable due to overwriting of statuses caused by reopenings, reassignments and resolutions. To account for temporal evolution, the dataset is structured into bimonthly time windows, where each product is evaluated across successive observation periods (e.g., Set I: 09/23-10/23 and Set II: 11/23-12/23). This temporal segmentation captures the evolving nature of OSS maintenance, where only recent bug statuses are available in the data under study.
- Consequently, the decision problem can be formulated as a multi-criteria, multi-product and time-evolving ranking problem, in which each OSS product is evaluated against multiple, potentially conflicting parameters derived from bug life cycle states. The uncertainty arises from missing values, evolving bug statuses, and asynchronous updates across repositories. Since these uncertainties affect all products uniformly within the same observational framework, the evaluation remains unbiased across alternatives.
- The uncertainty also arises at the level of performance indicators and products due to fluctuations in reporting, resolution delays and incomplete records. This structure leads to an MCDM formulation, where products represent alternatives, bug life cycle indicators represent parameters, and temporal snapshots represent dynamic evaluation layers.

The resulting decision environment is therefore characterized by fuzziness, indeterminacy and temporal variability, motivating the use of advanced uncertainty-aware models, such as the d-set method for robust OSS product ranking under both long-term and short-term decision contexts.

## 4. Methodology

We present the details of the d-set methodology for evaluating software maintenance performance. The application of the d-set methodology leads to the adaptation of an otherwise static evaluation model into a dynamic performance ranking system tailored to OSS environments. Let $U$ denote the universal set of OSS product alternatives to be ranked, and let $E$ be the set of evaluation criteria (parameters). The following steps outline the process for evaluating product ranks.

**Step I- Construction of IVIFP Set**

An IVIFP set is characterized by assigning to each element an interval-valued membership degree and an interval-valued non-membership degree, both lying in $[0,1]$ (Atanassov & Gargov, 1989). These intervals represent the possible ranges of membership and non-membership values, allowing the model to capture uncertainty more flexibly than classical intuitionistic fuzzy sets do. Aydın & Enginoğlu (2021) defined the IVIFP set $\kappa$ as follows:

$$\kappa = \left\{ {}^{\alpha(x)}_{\beta(x)} x : x \in E \right\} \tag{1}$$

where, $\alpha(x)$ and $\beta(x)$ denote the interval-valued membership ($\alpha(x) = [\alpha^{-}(x), \alpha^{+}(x)]$) and non-membership ($\beta(x) = [\beta^{-}(x), \beta^{+}(x)]$) functions for parameter $x \in E$. Here $\alpha^{-}$ and $\alpha^{+}$denote the lower and upper bounds of the interval-valued membership function and $\beta^{-}$and $\beta^{+}$ denote the lower and upper bounds of the interval-valued non-membership function. Equations (2) and (3) give the detailed computation of $\alpha(x)$ and $\beta(x)$, respectively:

$$\alpha(x) = \left[\frac{\underset{n}{min}\ \mu_n^x}{\underset{n}{max}\ \mu_n^x + \underset{n}{max}\ \nu_n^x + min\left(\underset{\kappa\in I}{min}\ \pi_\kappa^x, \underset{t\in J}{min}\ \pi_t^x\right)}, \frac{\underset{n}{max}\ \mu_n^x}{\underset{n}{max}\ \mu_n^x + \underset{n}{max}\ \nu_n^x + min\left(\underset{\kappa\in I}{min}\ \pi_\kappa^x, \underset{t\in J}{min}\ \pi_t^x\right)}\right] \quad (2)$$

$$\beta(x) = \left[\frac{\underset{n}{min}\ \nu_n^x}{\underset{n}{max}\ \mu_n^x + \underset{n}{max}\ \nu_n^x + min\left(\underset{\kappa\in I}{min}\ \pi_\kappa^x, \underset{t\in J}{min}\ \pi_t^x\right)}, \frac{\underset{n}{max}\ \nu_n^x}{\underset{n}{max}\ \mu_n^x + \underset{n}{max}\ \nu_n^x + min\left(\underset{\kappa\in I}{min}\ \pi_\kappa^x, \underset{t\in J}{min}\ \pi_t^x\right)}\right] \quad (3)$$

where, $\mu_n^x$, $\nu_n^x$, $\pi_n^x$ are the membership, non-membership and indeterminacy degrees of the parameter $x \in E$ for the $n$-th observation, respectively. $I$, $J$ are the index sets of maximum membership and non-membership values, respectively: $I = \{k: \max_n \mu_n^x = \mu_k^x\}, J = \{t: \max_n \nu_n^x = \nu_t^x\}$.

**Step II- Normalization and Construction of IVIFSS**

Alternatives are first normalized to the interval $[0, 1]$ using benefit- or cost-based min-max normalization methods. This process makes different features comparable while preserving proportions.

$$Normalized\ Benefit = \frac{B_i - B_{min}}{B_{max} - B_{min}} \quad (4)$$

$$Normalized\ Cost = \frac{C_{\max} - C_i}{C_{\max} - C_{\min}} \quad (5)$$

where, $B_i$ denotes the value of a benefit-type parameter for alternative $i$, with $B_{min}$ and $B_{max}$ as the minimum and maximum values of that parameter across all the alternatives. Similarly, $C_i$ denotes the value of a cost-type parameter, with $C_{\min}$ and $C_{\max}$ being the corresponding minimum and maximum values across all the alternatives.

Let $u \in U$ denote an alternative under comparative evaluation and let $x \in E$ denote an evaluation parameter. Then the IVIFSS is defined as: $f = \{({}^{\alpha(x)}_{\beta(x)}x, \{{}^{\omega(u)}_{\theta(u)}u: u \in U\}): x \in E\}$ using the interval-valued membership ($\omega_x$(u)) and non-membership ($\theta_x(u)$) degrees of alternative $u$ for parameter $x$ (Aydın & Enginoğlu, 2021):

$$\omega_x(u) = \left[\frac{\min_n(\mu_u^x)_n}{\max_n(\mu_u^x)_n + \max_n\{1-(\mu_u^x)_n\}}, \frac{\max_n(\mu_u^x)_n}{\max_n(\mu_u^x)_n + \max_n\{1-(\mu_u^x)_n\}}\right] \quad (6)$$

$$\theta_x(u) = \left[\frac{\min_n\{1-(\mu_u^x)_n\}}{\max_n(\mu_u^x)_n + \max_n\{1-(\mu_u^x)_n\}}, \frac{\max_n\{1-(\mu_u^x)_n\}}{\max_n(\mu_u^x)_n + \max_n\{1-(\mu_u^x)_n\}}\right] \quad (7)$$

where, $(\mu_u^x)_n$ corresponds to the $n$-th observation of alternative $u$ and parameter $x$. The unified IVIFSS: $f^* = \{{}^{\omega^*(u)}_{\theta^*(u)}u: u \in U\}$, aggregates information from all parameters to compute interval-valued membership and non-membership degrees $\omega^* = [\omega_-^*, \omega_+^*]$ and $\theta^* = [\theta_-^*, \theta_+^*]$) respectively, for each alternative. These intervals are obtained from the IVIFP and IVIFSS lower and upper bounds defined in Equations (2), (3), (6) and (7):

$$\omega^*(u) = \frac{1}{|\kappa|_a}\sum_{x\in E} \alpha\,(x)\,\omega(u),\ \ \theta^*(u) = \frac{1}{|\kappa|_a}\sum_{x\in E} \beta\,(x)\,\theta(u) \quad (8)$$

where, $|\kappa|_a$ is the average cardinality for $|E| = m$ parameters calculated using Equations (2) and (3); and allows comparison of sets without ambiguity (Huang et al., 2013):

$$|\kappa|_a = \frac{1}{2}\sum_{i=1}^{m}\left(1 + \frac{\alpha^-(x_i)+\alpha^+(x_i)}{2} - \frac{\beta^-(x_i)+\beta^+(x_i)}{2}\right) \quad (9)$$

**Step III- Formation of d-Set (Aydın &Enginoğlu, 2021)**
The lower and upper score bounds $s^{-}(u_k)$ and $s^{+}(u_k)$ for the $k$-th alternative for $u_k \in U$ are obtained from Equation (8) as follows:

$$s^{-}(u_k) = \omega_{-}^{*}(u_k) - \theta_{+}^{*}(u_k),\ s^{+}(u_k) = \omega_{+}^{*}(u_k) - \theta_{-}^{*}(u_k) \tag{10}$$

The interval-valued decision assessment $d(u_k)$ for the $k$-th alternative is obtained by normalizing the scores obtained from Equation (10). This final decision set is called the d-set:

$$d(u_k) = \left[\frac{s^{-}(u_k)+\left|\min\limits_{i}\{s^{-}(u_i)\}\right|}{\max\limits_{i}\{s^{+}(u_i)\}+\left|\min\limits_{i}\{s^{-}(u_i)\}\right|}, \frac{s^{+}(u_k)+\left|\min\limits_{i}\{s^{-}(u_i)\}\right|}{\max\limits_{i}\{s^{+}(u_i)\}+\left|\min\limits_{i}\{s^{-}(u_i)\}\right|}\right] \tag{11}$$

**Step IV- Lexicographic Ranking of d-Set**
To rank the products represented by interval-valued d-sets, lexicographic ordering is applied to the two-tuple values (lower and upper bounds) of d-sets obtained from Equation (11). Lexicographic ordering is applied by first comparing lower bounds and, in the case of ties, upper bounds (Xu & Yager, 2006). This ordering makes it possible to obtain a consistent and deterministic ranking of alternatives represented by interval scores despite interval-valued outcomes.

For any two alternatives $u_p$ and $u_q$:

$$u_p \succ u_q \Leftrightarrow \left(d^{-}(u_p), d^{+}(u_p)\right) >_{\text{lex}} \left(d^{-}(u_q), d^{+}(u_q)\right) \tag{12}$$

where the lexicographic order is given by:

$$(a_1, a_2) >_{\text{lex}} (b_1, b_2) \Leftrightarrow (a_1 > b_1) \text{ or } (a_1 = b_1 \text{ and } a_2 > b_2) \tag{13}$$

**Figure 3** presents the d-set-based framework as a sequential workflow that puts together the major computational steps of the ranking process.

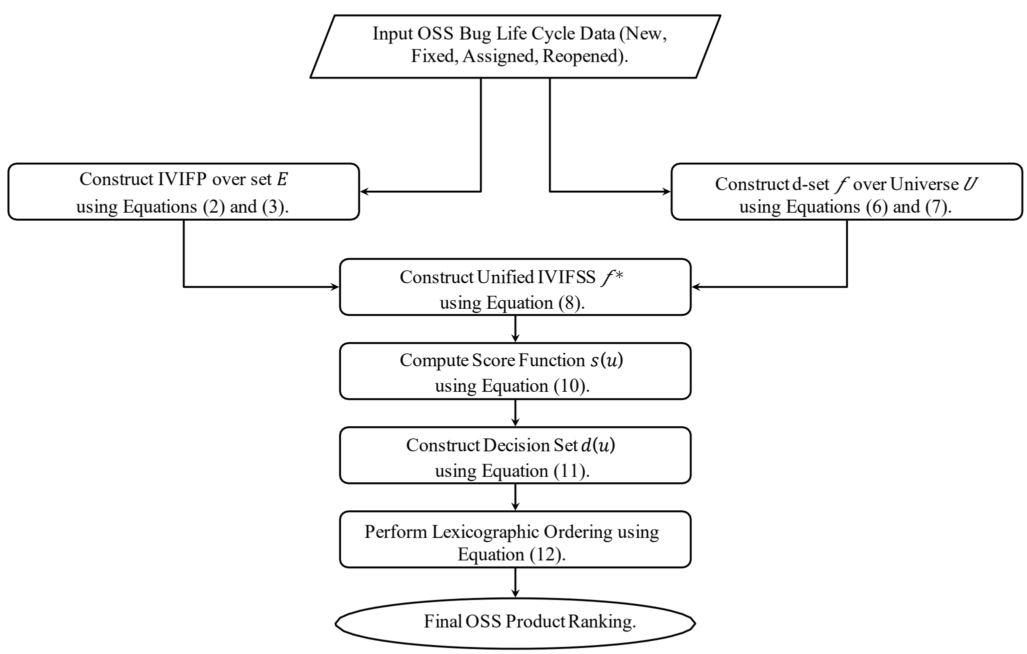


**Figure 3.** Flowchart of the proposed d-set-based OSS product ranking framework.

## 5. Empirical Case Study: Results and Discussion

This section presents an empirical case study on OSS maintenance using bug life cycle data from the Bugzilla repository. The analysis is conducted under two complementary evaluation case contexts to examine maintenance performance from both unified and temporal perspectives. Case context 1 applies the d-set methodology to derive the ranking $R^*$ using maintenance information collected over the complete one-year observation period. Case context 2 performs temporal ranking analysis across segmented bimonthly decision windows to evaluate ranking consistency over shorter observation periods, followed by Borda-based consensus aggregation to obtain an overall temporal ranking. Together, these two case contexts enable the assessment of both long-term maintenance performance and short-term temporal fluctuations in OSS environments. To assess the robustness of the proposed ranking framework, a sensitivity analysis was also performed. Finally, the overall workflow of the proposed framework is summarized at the end of this section. The following subsections present empirical analysis and results.

### 5.1 Data Preprocessing

Data for OSS products were collected from Bugzilla over a one-year period (September 1, 2023, to August 31, 2024). This period served as the evaluation horizon for assessing bug handling and ranking from an annual performance perspective. Following preliminary analysis, 16 products were retained as they exhibited relatively low sparsity and sufficient data density. Four parameters were derived from the bug life cycle: (i) newly raised bugs, reflecting maintenance demand and perceived software quality; (ii) fixed bugs, representing resolution effectiveness and backlog reduction; (iii) currently assigned bugs, indicating work-in-progress and resource allocation; and (iv) reopened bugs, suggesting rework requirements and prior fix quality. No extensive data-cleaning procedures were applied, except for filtering bugs based on the identified parameters (new, fixed, assigned and reopened). The alternatives set and evaluation parameters are defined as follows:

- The alternative set consists of OSS products from Bugzilla: $U = \{u_1, u_2, \ldots u_{16}\}$, where $u_1$ = "bugzilla.mozilla.org", $u_2$ = "CA Program", $u_3$ = "Calendar", $u_4$ = "Conduit", $u_5$ ="Data Platform and Tools", $u_6$ = "Developer Infrastructure", $u_7$ = "GeckoView", $u_8$ = "MailNews Core", $u_9$ = "Mozilla QA", $u_{10}$ = "mozilla.org", $u_{11}$ = "NSS", $u_{12}$ = "Release Engineering", $u_{13}$ = "Remote Protocol", $u_{14}$= "support.mozilla.org", $u_{15}$ = "WebExtensions" and $u_{16}$ = " Webtools ".

- The decision parameter set used to evaluate the alternatives is defined as $E = \{x_1, x_2, x_3, x_4\}$, where $x_1$ = "Number of new bugs (*N*)", $x_2$ = "Number of fixed bugs (*F*)", $x_3$ = "Number of assigned bugs (A)" and $x_4$ = "Number of reopened bugs (*R*)".

### 5.2 Case Context 1: d-Set-Based Unified Product Ranking

We present the details of the d-set-based product ranking framework, which ranks 16 OSS products across four parameters over a one-year observation period. Data from consecutive months were integrated and aligned at the product level to derive the unified ranking- $R^*$ over the complete observation horizon. Bug records were then grouped into four mutually exclusive categories: new, fixed, assigned and reopened. Computations were performed using Microsoft Excel 365 and the R programming language, and the code is available in a public repository (refer to the Code Availability section). **Table 2** reports the resulting bimonthly dataset used for d-sets-based ranking over the one-year observation period.

**Table 2.** Bimonthly bug count by product for status: $N = New$, $F = Fixed$, $A = Assigned$, $R = Reopened$ (6 sets, 2 months each).

| Product | Set I (09/23-10/23) | | | | Set II (11/23-12/23) | | | |
|---|---|---|---|---|---|---|---|---|
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 6 | 32 | 2 | 0 | 4 | 43 | 1 | 0 |
| CA Program | 2 | 34 | 2 | 0 | 0 | 22 | 3 | 0 |
| Calendar | 26 | 16 | 0 | 0 | 15 | 9 | 0 | 1 |
| Conduit | 19 | 24 | 0 | 2 | 5 | 29 | 0 | 0 |
| Data Platform and Tools | 20 | 79 | 1 | 0 | 57 | 112 | 2 | 2 |
| Developer Infrastructure | 23 | 62 | 2 | 0 | 10 | 43 | 0 | 2 |
| GeckoView | 20 | 30 | 2 | 1 | 22 | 36 | 1 | 2 |
| MailNews Core | 38 | 30 | 1 | 3 | 62 | 42 | 1 | 2 |
| Mozilla QA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Mozilla.org | 0 | 66 | 0 | 0 | 3 | 88 | 0 | 0 |
| NSS | 49 | 29 | 1 | 0 | 311 | 61 | 16 | 7 |
| Release Engineering | 11 | 54 | 2 | 0 | 7 | 68 | 3 | 1 |
| Remote Protocol | 21 | 50 | 0 | 0 | 24 | 58 | 1 | 1 |
| support.mozilla.org | 9 | 18 | 1 | 1 | 3 | 25 | 0 | 1 |
| WebExtensions | 38 | 40 | 1 | 3 | 50 | 39 | 3 | 2 |
| Webtools | 6 | 9 | 0 | 0 | 4 | 12 | 0 | 0 |
| **Product** | **Set III (01/24-02/24)** | | | | **Set IV (03/24-04/24)** | | | |
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 144 | 51 | 6 | 2 | 10 | 88 | 1 | 2 |
| CA Program | 1 | 27 | 7 | 0 | 1 | 29 | 9 | 0 |
| Calendar | 34 | 8 | 0 | 0 | 20 | 12 | 0 | 1 |
| Conduit | 27 | 34 | 2 | 1 | 34 | 54 | 1 | 1 |
| Data Platform and Tools | 46 | 110 | 0 | 2 | 34 | 122 | 1 | 1 |
| Developer Infrastructure | 35 | 47 | 1 | 1 | 23 | 61 | 1 | 1 |
| GeckoView | 26 | 25 | 3 | 1 | 29 | 40 | 0 | 0 |
| MailNews Core | 107 | 46 | 1 | 3 | 93 | 50 | 2 | 2 |
| Mozilla QA | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Mozilla.org | 3 | 86 | 0 | 0 | 1 | 84 | 0 | 0 |
| NSS | 23 | 37 | 4 | 1 | 15 | 22 | 7 | 0 |
| Release Engineering | 11 | 67 | 0 | 1 | 12 | 116 | 2 | 0 |
| Remote Protocol | 33 | 74 | 1 | 1 | 22 | 64 | 0 | 0 |
| support.mozilla.org | 10 | 48 | 3 | 1 | 4 | 27 | 1 | 2 |
| WebExtensions | 133 | 57 | 14 | 3 | 52 | 34 | 4 | 4 |
| Webtools | 18 | 9 | 0 | 1 | 9 | 11 | 0 | 0 |
| **Product** | **Set V (05/24-06/24)** | | | | **Set VI (07/24-08/24)** | | | |
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 18 | 107 | 2 | 3 | 5 | 70 | 2 | 1 |
| CA Program | 1 | 144 | 3 | 1 | 2 | 63 | 26 | 0 |
| Calendar | 78 | 23 | 2 | 1 | 28 | 14 | 1 | 2 |
| Conduit | 33 | 87 | 0 | 0 | 28 | 56 | 0 | 0 |
| Data Platform and Tools | 30 | 79 | 3 | 1 | 51 | 66 | 3 | 0 |
| Developer Infrastructure | 38 | 50 | 1 | 3 | 20 | 33 | 1 | 1 |
| GeckoView | 110 | 177 | 1 | 5 | 116 | 40 | 1 | 9 |
| MailNews Core | 143 | 56 | 9 | 6 | 108 | 54 | 4 | 6 |
| Mozilla QA | 0 | 53 | 7 | 0 | 4 | 96 | 8 | 1 |
| mozilla.org | 1 | 112 | 0 | 0 | 3 | 72 | 0 | 0 |
| NSS | 19 | 74 | 6 | 2 | 9 | 38 | 1 | 1 |
| Release Engineering | 19 | 112 | 3 | 1 | 20 | 60 | 2 | 5 |
| Remote Protocol | 62 | 69 | 1 | 2 | 52 | 90 | 1 | 6 |
| support.mozilla.org | 4 | 75 | 0 | 0 | 10 | 31 | 2 | 1 |
| WebExtensions | 96 | 85 | 1 | 4 | 79 | 52 | 15 | 9 |
| Webtools | 26 | 53 | 0 | 2 | 12 | 31 | 1 | 0 |

The stepwise details of the d-set ranking framework are as follows.

**Step I- Construction of IVIFP Set**

The most recent dataset (Set VI) was used to construct the IVIFP set. The use of recent data is critical in this context as bug status in Bugzilla is updated on an ongoing basis, with earlier states often being overwritten. The membership ($\mu$), non-membership ($\nu$) and indeterminacy ($\pi$) components for the Set VI values were mapped using a structured assignment scheme based on the relative magnitude of observed values, as reported in **Table 3**. To ensure consistency, a monotonic assignment principle is followed for each parameter: the most desirable values are associated with stronger membership, the least desirable with non-membership and intermediate or ambiguous values contribute to indeterminacy, thereby ensuring a consistent ordering from observed values to IVIFP s-tuples. For example, for the benefit-type parameter *fixed bugs* ($F$), higher values were allocated to the membership component: $\mu_{VI}^{F} = (70,72,90,96)$, intermediate values contribute to the indeterminacy component $\pi_{VI}^{F} = (63,56,66,40,54,60,52)$ and lower values were assigned to the non-membership component $\nu_{VI}^{F} = (14,33,38,31,31)$. On the other hand, for the cost-type parameter *new bugs* ($N$), lower values were associated with the membership component $\mu_{VI}^{N} = (5,2,3,9,4)$, intermediate values contribute to the indeterminacy component $\pi_{VI}^{N} = (28,28,20,20,10,12)$, whereas higher values were assigned to the non-membership component $\nu_{VI}^{N} = (51,116,108,52,79)$. The assignment process therefore follows predefined rules instead of relying on arbitrary choices, enabling replicability across comparable datasets.

**Table 3.** Ordered s-tuples for membership, non-membership and indeterminacy degrees of parameters N, F, A and R.

| N | F | A | R |
|---|---|---|---|
| $\mu_{VI}^{N}$ (5,2,3,9,4) | $\mu_{VI}^{F}$ (70,72,90,96) | $\mu_{VI}^{A}$ (26,15) | $\mu_{VI}^{R}$ (1,0,0,0,1,0,1,1,0,1) |
| $\pi_{VI}^{N}$ (28,28,20,20,10,12) | $\pi_{VI}^{F}$ (63,56,66,40,54,60,52) | $\pi_{VI}^{A}$ (4,8) | $\pi_{VI}^{R}$ (2,5) |
| $\nu_{VI}^{N}$ (51,116,108,52,79) | $\nu_{VI}^{F}$ (14,33,38,31,31) | $\nu_{VI}^{A}$ (2,1,0,3,1,1,0,1,2,1,2,1) | $\nu_{VI}^{R}$ (6,6,9,9) |

Using Equations (2) and (3), the IVIFP set $\kappa$ over the parameter set $E$ was obtained as follows:

$$\kappa = \left\{\frac{[0.015,0.067]}{[0.074,0.207]}N,\ \frac{[0.402,0.552]}{[0.230,0.379]}F,\ \frac{[0.455,0.788]}{[0.121,0.242]}A,\ \frac{[0.000,0.083]}{[0.167,0.417]}R\right\}.$$

**Table 4** presents the lower and upper bounds of set $\kappa$ used to compute the normalization factor.

**Table 4.** $\alpha(x)$ and $\beta(x)$ *for* normalization factors.

| | N | F | A | R |
|---|---|---|---|---|
| $\alpha^{-}$ | 0.015 | 0.402 | 0.455 | 0.000 |
| $\alpha^{+}$ | 0.067 | 0.552 | 0.788 | 0.083 |
| $\beta^{-}$ | 0.074 | 0.230 | 0.121 | 0.167 |
| $\beta^{+}$ | 0.207 | 0.379 | 0.242 | 0.417 |

**Step II- Normalization and Construction of IVIFSS**

Using Equations (4) and (5), the bimonthly data in **Table 2** were normalized. Assigned and fixed bugs are treated as benefit-type parameters, whereas new and reopened bugs are treated as cost-type parameters. The normalized dataset is presented in **Table 5**.

**Table 5.** Normalized parameter values across six sets of 16 products.

| Product | Set I | | | | Set II | | | |
|---|---|---|---|---|---|---|---|---|
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 0.981 | 0.181 | 0.923 | 0.000 | 0.987 | 0.243 | 0.962 | 0.000 |
| CA Program | 0.994 | 0.192 | 0.923 | 0.000 | 1.000 | 0.124 | 0.885 | 0.000 |
| Calendar | 0.916 | 0.090 | 1.000 | 0.000 | 0.952 | 0.051 | 1.000 | 0.111 |
| Conduit | 0.939 | 0.136 | 1.000 | 0.222 | 0.984 | 0.164 | 1.000 | 0.000 |
| Data Platform and Tools | 0.936 | 0.446 | 0.962 | 0.000 | 0.817 | 0.633 | 0.923 | 0.222 |
| Developer Infrastructure | 0.926 | 0.350 | 0.923 | 0.000 | 0.968 | 0.243 | 1.000 | 0.222 |
| GeckoView | 0.936 | 0.169 | 0.923 | 0.111 | 0.929 | 0.203 | 0.962 | 0.222 |
| MailNews Core | 0.878 | 0.169 | 0.962 | 0.333 | 0.801 | 0.237 | 0.962 | 0.222 |
| Mozilla QA | 1.000 | 0.000 | 1.000 | 0.000 | 1.000 | 0.000 | 1.000 | 0.000 |
| Mozilla.org | 1.000 | 0.373 | 1.000 | 0.000 | 0.990 | 0.497 | 1.000 | 0.000 |
| NSS | 0.842 | 0.164 | 0.962 | 0.000 | 0.000 | 0.345 | 0.385 | 0.778 |
| Release Engineering | 0.965 | 0.305 | 0.923 | 0.000 | 0.977 | 0.384 | 0.885 | 0.111 |
| Remote Protocol | 0.932 | 0.282 | 1.000 | 0.000 | 0.923 | 0.328 | 0.962 | 0.111 |
| support.mozilla.org | 0.971 | 0.102 | 0.962 | 0.111 | 0.990 | 0.141 | 1.000 | 0.111 |
| WebExtensions | 0.878 | 0.226 | 0.962 | 0.333 | 0.839 | 0.220 | 0.885 | 0.222 |
| Webtools | 0.981 | 0.051 | 1.000 | 0.000 | 0.987 | 0.068 | 1.000 | 0.000 |
| **Product** | **Set III** | | | | **Set IV** | | | |
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 0.537 | 0.288 | 0.769 | 0.222 | 0.968 | 0.497 | 0.962 | 0.222 |
| CA Program | 0.997 | 0.153 | 0.731 | 0.000 | 0.997 | 0.164 | 0.654 | 0.000 |
| Calendar | 0.891 | 0.045 | 1.000 | 0.000 | 0.936 | 0.068 | 1.000 | 0.111 |
| Conduit | 0.913 | 0.192 | 0.923 | 0.111 | 0.891 | 0.305 | 0.962 | 0.111 |
| Data Platform and Tools | 0.852 | 0.621 | 1.000 | 0.222 | 0.891 | 0.689 | 0.962 | 0.111 |
| Developer Infrastructure | 0.887 | 0.266 | 0.962 | 0.111 | 0.926 | 0.345 | 0.962 | 0.111 |
| GeckoView | 0.916 | 0.141 | 0.885 | 0.111 | 0.907 | 0.226 | 1.000 | 0.000 |
| MailNews Core | 0.656 | 0.260 | 0.962 | 0.333 | 0.701 | 0.282 | 0.923 | 0.222 |
| Mozilla QA | 0.997 | 0.000 | 1.000 | 0.000 | 1.000 | 0.006 | 1.000 | 0.000 |
| mozilla.org | 0.990 | 0.486 | 1.000 | 0.000 | 0.997 | 0.475 | 1.000 | 0.000 |
| NSS | 0.926 | 0.209 | 0.846 | 0.111 | 0.952 | 0.124 | 0.731 | 0.000 |
| Release Engineering | 0.965 | 0.379 | 1.000 | 0.111 | 0.961 | 0.655 | 0.923 | 0.000 |
| Remote Protocol | 0.894 | 0.418 | 0.962 | 0.111 | 0.929 | 0.362 | 1.000 | 0.000 |
| support.mozilla.org | 0.968 | 0.271 | 0.885 | 0.111 | 0.987 | 0.153 | 0.962 | 0.222 |
| WebExtensions | 0.572 | 0.322 | 0.462 | 0.333 | 0.833 | 0.192 | 0.846 | 0.444 |
| Webtools | 0.942 | 0.051 | 1.000 | 0.111 | 0.971 | 0.062 | 1.000 | 0.000 |
| **Product** | **Set V** | | | | **Set VI** | | | |
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 0.942 | 0.605 | 0.923 | 0.333 | 0.984 | 0.395 | 0.923 | 0.111 |
| CA Program | 0.997 | 0.814 | 0.885 | 0.111 | 0.994 | 0.356 | 0.000 | 0.000 |
| Calendar | 0.749 | 0.130 | 0.923 | 0.111 | 0.910 | 0.079 | 0.962 | 0.222 |
| Conduit | 0.894 | 0.492 | 1.000 | 0.000 | 0.910 | 0.316 | 1.000 | 0.000 |
| Data Platform and Tools | 0.904 | 0.446 | 0.885 | 0.111 | 0.836 | 0.373 | 0.885 | 0.000 |
| Developer Infrastructure | 0.878 | 0.282 | 0.962 | 0.333 | 0.936 | 0.186 | 0.962 | 0.111 |
| GeckoView | 0.646 | 1.000 | 0.962 | 0.556 | 0.627 | 0.226 | 0.962 | 1.000 |
| MailNews Core | 0.540 | 0.316 | 0.654 | 0.667 | 0.653 | 0.305 | 0.846 | 0.667 |
| Mozilla QA | 1.000 | 0.299 | 0.731 | 0.000 | 0.987 | 0.542 | 0.692 | 0.111 |
| mozilla.org | 0.997 | 0.633 | 1.000 | 0.000 | 0.990 | 0.407 | 1.000 | 0.000 |
| NSS | 0.939 | 0.418 | 0.769 | 0.222 | 0.971 | 0.215 | 0.962 | 0.111 |
| Release Engineering | 0.939 | 0.633 | 0.885 | 0.111 | 0.936 | 0.339 | 0.923 | 0.556 |
| Remote Protocol | 0.801 | 0.390 | 0.962 | 0.222 | 0.833 | 0.508 | 0.962 | 0.667 |
| support.mozilla.org | 0.987 | 0.424 | 1.000 | 0.000 | 0.968 | 0.175 | 0.923 | 0.111 |
| WebExtensions | 0.691 | 0.480 | 0.962 | 0.444 | 0.746 | 0.294 | 0.423 | 1.000 |
| Webtools | 0.916 | 0.299 | 1.000 | 0.222 | 0.961 | 0.175 | 0.962 | 0.000 |

**Table 6** summarizes the minimum and maximum parameter values across all products and sets.

**Table 6.** Minimum and maximum values of the four parameters across all products.

| Product | Minimum | | | | Maximum | | | |
|---|---|---|---|---|---|---|---|---|
| | N | F | A | R | N | F | A | R |
| bugzilla.mozilla.org | 0.537 | 0.181 | 0.038 | 0.667 | 0.987 | 0.605 | 0.231 | 1.000 |
| CA Program | 0.994 | 0.124 | 0.077 | 0.889 | 1.000 | 0.814 | 1.000 | 1.000 |
| Calendar | 0.749 | 0.045 | 0.000 | 0.778 | 0.952 | 0.130 | 0.077 | 1.000 |
| Conduit | 0.891 | 0.136 | 0.000 | 0.778 | 0.984 | 0.492 | 0.077 | 1.000 |
| Data Platform and Tools | 0.817 | 0.373 | 0.000 | 0.778 | 0.936 | 0.689 | 0.115 | 1.000 |
| Developer Infrastructure | 0.878 | 0.186 | 0.000 | 0.667 | 0.968 | 0.350 | 0.077 | 1.000 |
| GeckoView | 0.627 | 0.141 | 0.000 | 0.000 | 0.936 | 1.000 | 0.115 | 1.000 |
| MailNews Core | 0.540 | 0.169 | 0.038 | 0.333 | 0.878 | 0.316 | 0.346 | 0.778 |
| Mozilla QA | 0.987 | 0.000 | 0.000 | 0.889 | 1.000 | 0.542 | 0.308 | 1.000 |
| mozilla.org | 0.990 | 0.373 | 0.000 | 1.000 | 1.000 | 0.633 | 0.000 | 1.000 |
| NSS | 0.000 | 0.124 | 0.038 | 0.222 | 0.971 | 0.418 | 0.615 | 1.000 |
| Release Engineering | 0.936 | 0.305 | 0.000 | 0.444 | 0.977 | 0.655 | 0.115 | 1.000 |
| Remote Protocol | 0.801 | 0.282 | 0.000 | 0.333 | 0.932 | 0.508 | 0.038 | 1.000 |
| support.mozilla.org | 0.968 | 0.102 | 0.000 | 0.778 | 0.990 | 0.424 | 0.115 | 1.000 |
| WebExtensions | 0.572 | 0.192 | 0.038 | 0.000 | 0.878 | 0.480 | 0.577 | 0.778 |
| Webtools | 0.916 | 0.051 | 0.000 | 0.778 | 0.987 | 0.299 | 0.038 | 1.000 |

Based on Equations (6) and (7), the IVIFSS $f$ is constructed as shown in **Table 7**.

**Table 7.** IVIFSS $f$ interval values for parameters N, F, A and R.

| Product | N | | | | F | | | |
|---|---|---|---|---|---|---|---|---|
| | $\omega^-$ | $\omega^+$ | $\theta^-$ | $\theta^+$ | $\omega^-$ | $\omega^+$ | $\theta^-$ | $\theta^+$ |
| bugzilla.mozilla.org | 0.005 | 0.045 | 0.001 | 0.066 | 0.051 | 0.234 | 0.064 | 0.218 |
| CA Program | 0.015 | 0.066 | 0.000 | 0.001 | 0.030 | 0.266 | 0.025 | 0.197 |
| Calendar | 0.009 | 0.053 | 0.003 | 0.043 | 0.017 | 0.066 | 0.184 | 0.334 |
| Conduit | 0.012 | 0.060 | 0.001 | 0.021 | 0.040 | 0.200 | 0.086 | 0.242 |
| Data Platform and Tools | 0.011 | 0.056 | 0.004 | 0.034 | 0.114 | 0.289 | 0.054 | 0.181 |
| Developer Infrastructure | 0.012 | 0.059 | 0.002 | 0.023 | 0.064 | 0.166 | 0.128 | 0.265 |
| GeckoView | 0.007 | 0.048 | 0.004 | 0.059 | 0.031 | 0.297 | 0.000 | 0.175 |
| MailNews Core | 0.006 | 0.044 | 0.007 | 0.071 | 0.059 | 0.152 | 0.137 | 0.275 |
| Mozilla QA | 0.014 | 0.066 | 0.000 | 0.003 | 0.000 | 0.194 | 0.068 | 0.246 |
| mozilla.org | 0.015 | 0.066 | 0.000 | 0.002 | 0.119 | 0.277 | 0.067 | 0.189 |
| NSS | 0.000 | 0.033 | 0.001 | 0.105 | 0.039 | 0.178 | 0.103 | 0.257 |
| Release Engineering | 0.013 | 0.063 | 0.002 | 0.013 | 0.091 | 0.268 | 0.059 | 0.195 |
| Remote Protocol | 0.010 | 0.055 | 0.004 | 0.037 | 0.093 | 0.229 | 0.092 | 0.222 |
| support.mozilla.org | 0.014 | 0.065 | 0.001 | 0.007 | 0.031 | 0.177 | 0.100 | 0.258 |
| WebExtensions | 0.006 | 0.045 | 0.007 | 0.068 | 0.060 | 0.206 | 0.093 | 0.238 |
| Webtools | 0.013 | 0.061 | 0.001 | 0.016 | 0.016 | 0.132 | 0.129 | 0.288 |
| **Product** | **A** | | | | **R** | | | |
| | $\omega^-$ | $\omega^+$ | $\theta^-$ | $\theta^+$ | $\omega^-$ | $\omega^+$ | $\theta^-$ | $\theta^+$ |
| bugzilla.mozilla.org | 0.015 | 0.152 | 0.078 | 0.196 | 0.000 | 0.063 | 0.000 | 0.104 |
| CA Program | 0.018 | 0.410 | 0.000 | 0.116 | 0.000 | 0.075 | 0.000 | 0.042 |
| Calendar | 0.000 | 0.056 | 0.104 | 0.225 | 0.000 | 0.068 | 0.000 | 0.076 |
| Conduit | 0.000 | 0.056 | 0.104 | 0.225 | 0.000 | 0.068 | 0.000 | 0.076 |
| Data Platform and Tools | 0.000 | 0.082 | 0.096 | 0.217 | 0.000 | 0.068 | 0.000 | 0.076 |
| Developer Infrastructure | 0.000 | 0.056 | 0.104 | 0.225 | 0.000 | 0.063 | 0.000 | 0.104 |
| GeckoView | 0.000 | 0.082 | 0.096 | 0.217 | 0.000 | 0.042 | 0.000 | 0.208 |
| MailNews Core | 0.013 | 0.209 | 0.061 | 0.178 | 0.000 | 0.045 | 0.026 | 0.192 |
| Mozilla QA | 0.000 | 0.185 | 0.064 | 0.185 | 0.000 | 0.075 | 0.000 | 0.042 |
| mozilla.org | 0.000 | 0.000 | 0.121 | 0.242 | 0.000 | 0.083 | 0.000 | 0.000 |
| NSS | 0.011 | 0.307 | 0.030 | 0.148 | 0.000 | 0.047 | 0.000 | 0.182 |
| Release Engineering | 0.000 | 0.082 | 0.096 | 0.217 | 0.000 | 0.054 | 0.000 | 0.149 |
| Remote Protocol | 0.000 | 0.029 | 0.112 | 0.233 | 0.000 | 0.050 | 0.000 | 0.167 |
| support.mozilla.org | 0.000 | 0.082 | 0.096 | 0.217 | 0.000 | 0.068 | 0.000 | 0.076 |
| WebExtensions | 0.011 | 0.295 | 0.033 | 0.152 | 0.000 | 0.036 | 0.021 | 0.234 |
| Webtools | 0.000 | 0.029 | 0.112 | 0.233 | 0.000 | 0.068 | 0.000 | 0.076 |

The unified IVIFSS $f^*$, computed using Equation (8), is presented in **Table 8**. Using Equation (9), the average cardinality was calculated as 2.1309.

**Table 8.** Unified IVIFSS $f^*$.

| Product | $\omega_-'$ | $\omega_+'$ | $\theta^*_-$ | $\theta^*_+$ |
|---|---|---|---|---|
| bugzilla.mozilla.org | 0.0334 | 0.2321 | 0.0670 | 0.2741 |
| CA Program | 0.0293 | 0.3832 | 0.0119 | 0.1671 |
| Calendar | 0.0122 | 0.1142 | 0.1367 | 0.3182 |
| Conduit | 0.0245 | 0.1804 | 0.0897 | 0.2644 |
| Data Platform and Tools | 0.0586 | 0.2320 | 0.0726 | 0.2383 |
| Developer Infrastructure | 0.0358 | 0.1614 | 0.1100 | 0.2899 |
| GeckoView | 0.0177 | 0.2195 | 0.0468 | 0.3097 |
| MailNews Core | 0.0370 | 0.2109 | 0.1080 | 0.3363 |
| Mozilla QA | 0.0068 | 0.2441 | 0.0621 | 0.2232 |
| mozilla.org | 0.0627 | 0.2001 | 0.0883 | 0.2033 |
| NSS | 0.0233 | 0.2654 | 0.0629 | 0.3248 |
| Release Engineering | 0.0489 | 0.2184 | 0.0734 | 0.2694 |
| Remote Protocol | 0.0484 | 0.1703 | 0.0980 | 0.3091 |
| support.mozilla.org | 0.0211 | 0.1835 | 0.0925 | 0.2616 |
| WebExtensions | 0.0365 | 0.2733 | 0.0722 | 0.3246 |
| Webtools | 0.0136 | 0.1366 | 0.1136 | 0.2880 |

**Step III- Formation of d-Set**

**Table 9** reports the lower and upper bounds of the product scores and the resulting decision set, computed from Equations (10) and (11). **Figure 4** illustrates the interval-valued decision scores derived from the unified d-set.

**Table 9.** $s(u_k)$ and decision set $d(u_k)$ for unified IVIFSS $f^*$.

| Product | $s^-(u)$ | $s^+(u)$ | $d^-(u)$ | $d^+(u)$ |
|---|---|---|---|---|
| bugzilla.mozilla.org | -0.2407 | 0.1652 | 0.0964 | 0.6956 |
| CA Program | -0.1378 | 0.3713 | 0.2483 | 1.0000 |
| Calendar | -0.3060 | -0.0225 | 0.0000 | 0.4185 |
| Conduit | -0.2399 | 0.0907 | 0.0976 | 0.5857 |
| Data Platform and Tools | -0.1797 | 0.1594 | 0.1864 | 0.6871 |
| Developer Infrastructure | -0.2540 | 0.0514 | 0.0767 | 0.5277 |
| GeckoView | -0.2921 | 0.1726 | 0.0205 | 0.7066 |
| MailNews Core | -0.2993 | 0.1029 | 0.0099 | 0.6037 |
| Mozilla QA | -0.2164 | 0.1820 | 0.1322 | 0.7205 |
| mozilla.org | -0.1406 | 0.1118 | 0.2442 | 0.6168 |
| NSS | -0.3014 | 0.2025 | 0.0067 | 0.7507 |
| Release Engineering | -0.2205 | 0.1450 | 0.1261 | 0.6658 |
| Remote Protocol | -0.2607 | 0.0723 | 0.0669 | 0.5585 |
| support.mozilla.org | -0.2405 | 0.0911 | 0.0967 | 0.5862 |
| WebExtensions | -0.2881 | 0.2011 | 0.0264 | 0.7487 |
| Webtools | -0.2744 | 0.0230 | 0.0466 | 0.4857 |

**Step IV- Lexicographic Ranking of d-Set**

Using the lexicographic ordering of the interval-valued decision scores, the final unified ranking $R^*$ was obtained as follows:

CA Program > mozilla.org > Data Platform and Tools > Mozilla QA > Release Engineering > Conduit > support.mozilla.org > bugzilla.mozilla.org > Developer Infrastructure > Remote Protocol > Webtools > WebExtensions > GeckoView > MailNews Core > NSS > Calendar.

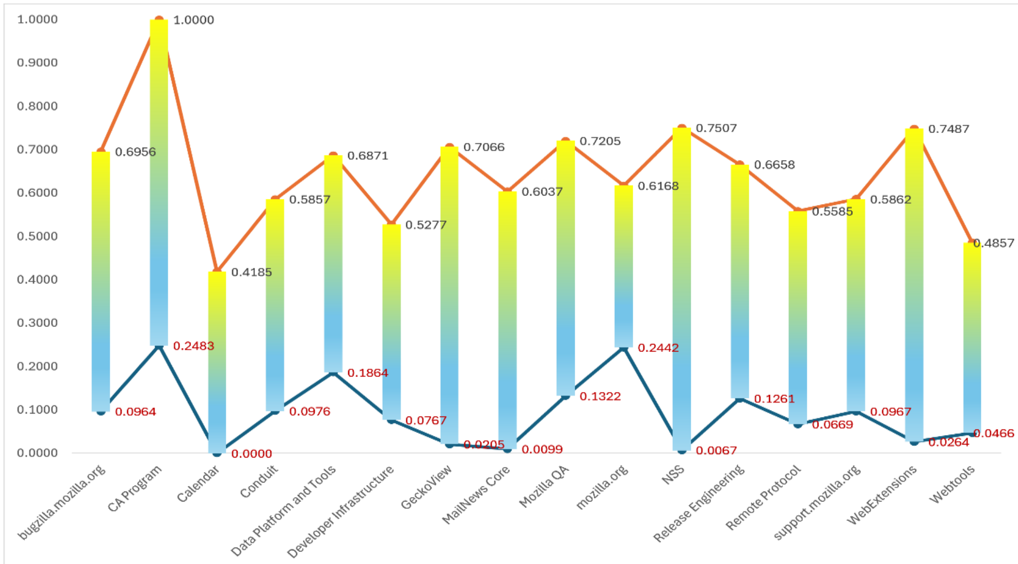


**Figure 4.** Interval-valued decision scores $\left(d^-(u_p), d^+(u_p)\right)$ for the unified IVIFSS $f^*$.

The framework produces interval-valued decision scores, allowing clear and consistent prioritization under uncertainty. Product ordering is primarily driven by higher lower-bound values of the decision intervals, providing deterministic and interpretable ranking. The interval-valued structure of the d-set represents the uncertainty and variation inherent in OSS bug repository data, including fluctuations in reporting behavior, resolution delays, and evolving maintenance dynamics over time. Unlike deterministic and point-estimate-based approaches, the employed framework accounts for uncertainty in both the parameters and the alternatives across multiple stages.

### 5.3 Case Context 2: Temporal Ranking and Consensus Aggregation

From an operational perspective, decision-makers often require interim performance monitoring to address concerns such as changes in resource allocation, escalation handling, maintenance evaluation, release planning, or shifting organizational focus across products. To address this, in case context 2, a temporal evaluation is carried out to examine maintenance-performance behavior over shorter time intervals. For this purpose, the OSS maintenance data for one year was segmented into six bimonthly decision windows. Product rankings $R_1$ to $R_6$ were then independently computed for each period using the d-set ranking framework. Each period was divided into four two-week datasets, and the d-set method was applied as in case context 1. The six temporal rankings capture localized maintenance dynamics and are reported in **Table 10**. If a decision-maker wishes to assess product performance over one year, the consensus representation of these temporal rankings can be computed using the Borda rank aggregation method (Borda, 1781). For this purpose, the consensus score for each product $i$, is computed as follows:

$$BO_i = \sum_{t=1}^{T} r_{it} \tag{14}$$

where, $r_{it}$ is the ordinal rank assigned to product $i$ in the temporal window $t$ and $T$ is the total number of windows. The final consensus ranking is obtained by ordering products in ascending order of $BO_i$, where lower values indicate stronger overall performance consensus across temporal windows.

The ranking produced in case context 1 ($R^*$) and the Borda consensus ranking (refer to **Table 10**) for the same time frame demonstrate moderate-to-strong agreement, with Spearman's $\rho = 0.75$ suggesting that both case contexts capture largely consistent maintenance-performance behavior. This agreement also suggests that the case context 2 evaluation framework based on Borda rank aggregation performs well in reducing temporal noise and captures persistent maintenance-performance signals across temporal windows. Borda aggregation favors positional consistency across temporal windows and implicitly treats equal ordinal separation between adjacent ranks. In contrast, the d-set approach preserves interval-valued uncertainty and performance proximity information during evaluation and ranking. As a result, the d-set-based ranking offers clearer distinction among closely ranked products under uncertain maintenance conditions. In practical OSS maintenance environments, decision-makers may require both long-horizon evaluation and shorter-period analysis depending on operational objectives. **Figure 5** illustrates the temporal evolution of product rankings across multiple decision periods, together with the ranking $R^*$.

**Table 10.** Product rankings across bimonthly windows, aggregated Borda ranking, and the baseline ranking $R^*$.

| Product | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | Borda rank | $R^*$ |
|---|---|---|---|---|---|---|---|---|
| bugzilla.mozilla.org | 6 | 3 | 10 | 7 | 7 | 4 | 5 | 8 |
| CA Program | 5 | 4 | 4 | 3 | 3 | 2 | 2 | 1 |
| Calendar | 12 | 16 | 11 | 16 | 14 | 14 | 16 | 16 |
| Conduit | 16 | 8 | 14 | 10 | 8 | 6 | 12 | 6 |
| Data Platform and Tools | 4 | 1 | 2 | 2 | 9 | 5 | 3 | 3 |
| Developer Infrastructure | 1 | 15 | 9 | 8 | 15 | 10 | 10 | 9 |
| GeckoView | 9 | 9 | 12 | 9 | 11 | 16 | 13 | 13 |
| MailNews Core | 11 | 10 | 15 | 14 | 12 | 15 | 15 | 14 |
| Mozilla QA | 15 | 13 | 13 | 12 | 1 | 1 | 8 | 4 |
| mozilla.org | 2 | 2 | 1 | 4 | 2 | 3 | 1 | 2 |
| NSS | 8 | 14 | 8 | 5 | 5 | 8 | 7 | 15 |
| Release Engineering | 3 | 5 | 5 | 1 | 6 | 13 | 4 | 5 |
| Remote Protocol | 7 | 6 | 7 | 6 | 10 | 11 | 6 | 10 |
| support.mozilla.org | 10 | 12 | 6 | 13 | 4 | 12 | 9 | 7 |
| WebExtensions | 13 | 7 | 3 | 15 | 13 | 9 | 11 | 12 |
| Webtools | 14 | 11 | 16 | 11 | 16 | 7 | 14 | 11 |

## 5.4 Sensitivity Analysis

To evaluate the robustness of the d-set-based ranking framework, a perturbation analysis was conducted by introducing controlled variations in the IVIFP set $(\alpha, \beta)$ reported in **Table 4**. For the sensitivity analysis, data for the entire one-year observation period, as in case context 1, were used. The rankings were recomputed under perturbation levels of $\pm 5\%$, $\pm 10\%$ and $\pm 15\%$ and compared with the case context 1 ranking $R^*$. The results in **Table 11** show an overall high degree of stability across all perturbation levels, with Spearman's $\rho$ consistently demonstrating a strong correlation between the obtained rank and $R^*$. Spearman's rank correlation coefficient is commonly employed to evaluate ranking consistency and pairwise ordering agreement across decision methods (Wu & Tiao, 2018). There are minor local rank variations among closely ranked alternatives; however, the overall ranking structure largely remains unaffected by moderate perturbations. These findings confirm the robustness, stability and reliability of the framework for decision-making under uncertainty.

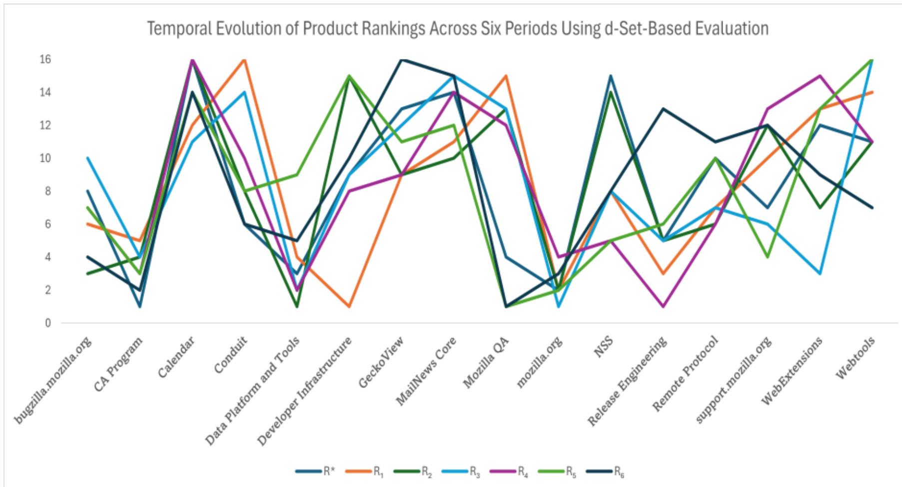


**Figure 5.** Temporal comparison of product rankings across six periods ( $R_1$-$R_6$) alongside the ranking $R^*$.

**Table 11.** Ranking stability of $R^*$ under different perturbation levels.

| **Product** | **Baseline Rank** | **5% Perturbation** | **10% Perturbation** | **15% Perturbation** |
|---|---|---|---|---|
| bugzilla.mozilla.org | 8 | 8 | 8 | 8 |
| CA Program | 1 | 1 | 1 | 1 |
| Calendar | 16 | 16 | 16 | 16 |
| Conduit | 6 | 6 | 7 | 7 |
| Data Platform and Tools | 3 | 3 | 3 | 3 |
| Developer Infrastructure | 9 | 9 | 9 | 9 |
| GeckoView | 13 | 13 | 13 | 12 |
| MailNews Core | 14 | 14 | 15 | 15 |
| Mozilla QA | 4 | 4 | 4 | 4 |
| mozilla.org | 2 | 2 | 2 | 2 |
| NSS | 15 | 15 | 14 | 14 |
| Release Engineering | 5 | 5 | 5 | 5 |
| Remote Protocol | 10 | 10 | 10 | 10 |
| support.mozilla.org | 7 | 7 | 6 | 6 |
| WebExtensions | 12 | 12 | 12 | 13 |
| Webtools | 11 | 11 | 11 | 11 |
| **Spearman's $\rho$** | — | 1 | 0.9941 | 0.9912 |

## 5.5 Proposed Framework for OSS Maintenance Performance Evaluation

This section summarizes the OSS product performance evaluation and ranking framework proposed in this study. **Figure 6** depicts the structured approach proposed for evaluating OSS maintenance performance under uncertainty using bug life cycle data consists of data preparation, uncertainty modeling, ranking generation, temporal evaluation, and robustness assessment within a unified decision-support process. The analysis begins with the collection and preprocessing of bug life cycle data from Bugzilla, where maintenance indicators are defined by four parameters: newly reported, fixed, assigned and reopened bugs. These parameters collectively represent several aspects of maintenance demand, resolution effectiveness, workload status, and bug resolution quality. To address the uncertainty and variability in data, the framework employs IVIFP representations and develops d-sets (IVIFP-IVIFSS) for performance evaluation. Reliability is further examined through sensitivity analysis conducted by perturbing the IVIFP set and comparing the resulting d-set rankings with the baseline ranking and evaluating the stability of the framework under moderate variations in the underlying uncertainty specifications.

The framework uses two evaluation settings:

- **Case context 1** generates a unified ranking. It combines six bimonthly maintenance timeframes to produce a one-year ranking of OSS products. A perturbation analysis is performed to verify the robustness of the ranks obtained using fuzzy d-set approach.
- **Case context 2** assesses individual performances of six bimonthly timeframes. Each bimonthly window in turn combines four biweekly timeframes. This setting enables the analysis of short-term maintenance changes and time-based variations among the individual ranks. The resulting rankings are later aggregated into a one-year timeframe using the Borda count method. This generates a consensus ranking that validates ranking consistency with baseline ranking of case context 1.

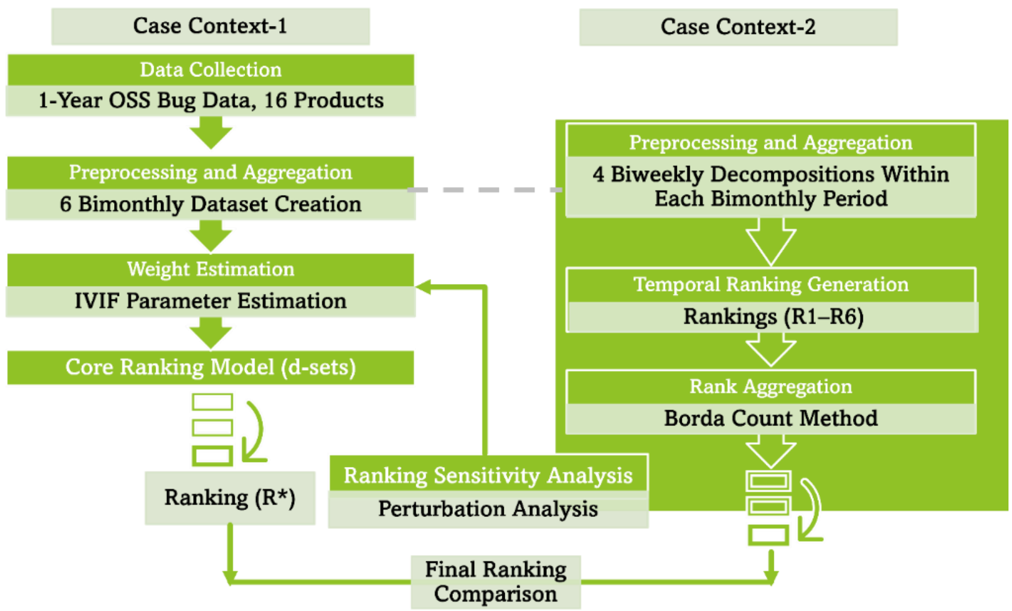


**Figure 6.** Proposed framework for OSS product performance evaluation and ranking.

Overall, the framework provides a systematic workflow that connects repository-based maintenance data to uncertainty-aware decision analysis. It enables both long-term and temporal assessments of OSS maintenance performance while ensuring the robustness of the resulting rankings.

## 6. Managerial Insights

The ranking framework transforms results into actionable insights for various stakeholders:

- For sponsors, the rankings can support funding decisions by identifying products needing more investment due to maintenance risks, and the ones that merit continued investment due to stable performance and value.
- For enhancement leads, the rankings aid planning and control with respect to sprint review, release planning, service monitoring, issue escalation, and resource adjustment.
- The framework presented in this study supports long-term strategic assessment through the unified ranking and short-term assessments, helping support managers distinguish persistent maintenance patterns and interim changes across decision windows.
- Time-based changes may reflect organizational priorities, release cycle pressures, workload shifts and changes in development focus across products. Module lead maintainers can plan maintenance

activities based on fluctuations in rankings over time. They may focus on lower-ranked components that show instability while giving less attention to consistently high-performing products.

- For development teams, the interval-valued scores improve decision confidence by separating stable products from those with high variability, guiding level of monitoring, backup planning, and corrective actions.
- For portfolio managers, the framework supports prioritization across multiple products under resource constraints while aligning maintenance efforts with strategic importance and risk exposure.

Overall, the proposed approach supports the transition from reactive bug fixing toward proactive evidence-driven maintenance, improving reliability, cost efficiency, and long-term sustainability while offering a practical decision-support tool for complex uncertain software maintenance environments.

## 7. Conclusion

This study presents a maintenance-focused performance evaluation of OSS systems using bug life cycle data in the d-set decision-making framework. It proposes a structured approach for modeling uncertain and dynamic maintenance data to obtain comparative product rankings. The empirical case study, using Bugzilla data, yields clear, stable and actionable rankings, showing the framework’s practical use for practical decision-support in uncertain and evolving software maintenance environments. The results show that the d-set methodology generates stable rankings under simultaneous uncertainty in parameters and alternatives across unified and temporal decision contexts. The baseline ranking $R^*$ reflects consistent maintenance performance across the full observation period, while temporal ranking analysis provides insights into shorter-term operational variations and evolving maintenance dynamics. Perturbation analysis further indicates the stability and robustness of the resulting rankings under moderate variations in input parameters, supporting the framework's reliability for decision-making under uncertainty.

One limitation is that the analysis is constrained to a one-year temporal window uses a single recent snapshot of bug data but lacks full historical state data. In addition, only a limited set of maintenance parameters has been examined, and additional factors such as team size, project maturity, and release cadence are excluded. Despite these limitations, the framework can be applied more broadly in OSS and industrial software maintenance areas.

Future research may extend this work by using longer time periods and more maintenance and quality metrics, including historical state-transition data and different weighting and aggregation schemes. These extensions may further improve the usability of d-set-based evaluation frameworks for maintenance governance, portfolio prioritization, and long-term sustainability assessment in complex software environments.

**Conflicts of Interest**

The authors confirm that there is no conflict of interest to declare for this publication.

**Acknowledgments**

This research did not receive any specific grant from funding agencies in the public, commercial, or non-profit sectors. The authors would like to thank the editor and the anonymous reviewers for their comments that helped improve the quality of this work.

**Code Availability**

To ensure full transparency and reproducibility, the implementation of the proposed d-sets-based ranking framework, along with the complete dataset, processing steps and analysis code, is publicly available in a version-controlled anonymized repository: https://github.com/nvkadam/oss-maintenance-ranking-framework. It incorporates the model implementation, experiments and results reported in this study.

**AI Discourse**

The author(s) confirm that generative AI tools were not used during the preparation of this article.

## References

Abdullah, S., Aslam, M., & Ullah, K. (2014). Bipolar fuzzy soft sets and its applications in decision making problem. *Journal of Intelligent & Fuzzy Systems*, *27*(2), 729-742.

Alkhazaleh, S. (2015). The multi-interval-valued fuzzy soft set with application in decision making. *Applied Mathematics*, *6*(8), 1250-1262.

Alkhazaleh, S. (2022). Effective fuzzy soft set theory and its applications. *Applied Computational Intelligence and Soft Computing*, *2022*(1), 6469745.

Alkhazaleh, S., & Salleh, A.R. (2011). Soft expert sets. *Advances in Decision Sciences*, *2011*, 757868.

Alkhazaleh, S., Salleh, A.R., & Hassan, N. (2011a). Fuzzy parameterized interval-valued fuzzy soft set. *Applied Mathematical Sciences*, *5*(67), 3335-3346.

Alkhazaleh, S., Salleh, A.R., & Hassan, N. (2011b). Possibility fuzzy soft set. *Advances in Decision Sciences*, *2011*, 479756.

Alkhazaleh, S., Salleh, A.R., & Hassan, N. (2011c). Soft multisets theory. *Applied Mathematical Sciences*, *5*(72), 3561-3573.

Amer, F., Hammoud, S., Khatatbeh, H., Lohner, S., Boncz, I., & Endrei, D. (2022). A systematic review: the dimensions to evaluate health care performance and an implication during the pandemic. *BMC Health Services Research*, *22*(1), 621.

Atanassov, K.T. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, *20*(1), 87-96.

Atanassov, K.T., & Gargov, G. (1989). Interval valued intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 31(3), 343-349.

Atlassian, Jira, Available at: https://www.atlassian.com/software/jira, Accessed: 2026.

Aydın, T., & Enginoğlu, S. (2021). Interval-valued intuitionistic fuzzy parameterized interval-valued intuitionistic fuzzy soft sets and their application in decision-making. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 1541-1558.

Borda, J.C. (1781). Mémoire sur les élections au scrutiny. *Histoire de l'Académie Royale des Sciences*, *102*, 657-665.

Bugzilla, Available at: https://bugzilla.mozilla.org, Accessed: 2026.

Çağman, N., Çıtak, F., & Enginoğlu, S. (2010). Fuzzy parameterized fuzzy soft set theory and its applications. *Turkish Journal of Fuzzy Systems*, *1*(1), 21-35.

Çağman, N., Çıtak, F., & Enginoğlu, S. (2011). FP-soft set theory and its applications. *Annals of Fuzzy Mathematics and Informatics*, *2*(2), 219-226.

Cakir, F.S., & Adiguzel, Z. (2020). Analysis of leader effectiveness in organization and knowledge sharing behavior on employees and organization. *SAGE Open*, *10*(1), 1.

Carvalhosa, S., & Lucas, A. (2026). A stable ranking framework using historical data envelopment analysis frontiers and Mahalanobis distance. *Decision Analytics Journal*, *19*, 100700.

Chang, D.Y. (1996). Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research*, *95*(3), 649-655.

Charnes, A., Cooper, W.W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, *2*(6), 429-444.

Chen, C.T. (2000). Extensions of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems*, *114*(1), 1-9.

Cosa, M., & Torelli, R. (2024). Digital transformation and flexible performance management: a systematic literature review of the evolution of performance measurement systems. *Global Journal of Flexible Systems Management*, *25*(3), 445-466.

Crant, J.M., & Bateman, T.S. (1993). Assignment of credit and blame for performance outcomes. *Academy of Management Journal*, *36*(1), 7-27.

Dahooie, J.H., Zavadskas, E.K., Abolhasani, M., Vanaki, A., & Turskis, Z. (2018). A novel approach for evaluation of projects using an interval-valued fuzzy additive ratio assessment (ARAS) method: a case study of oil and gas well drilling projects. *Symmetry*, *10*(2), 45.

de Araújo, M.L., Caldas, L.S., Barreto, B.S., Menezes, P.P.M., Silvério, J.C.D.S., Rodrigues, L.C., Serrano, A.L.M., Neumann, C., & Mendes, N. (2024). How to evaluate the effectiveness of performance management systems? an overview of the literature and a proposed integrative model. *Administrative Sciences*, *14*(6), 117.

de Melo Santos, C.J., Barbosa, A.S., & Sant'Anna, Â.M.O. (2025). Performance measurement systems in primary health care: a systematic literature review. *BMC Health Services Research*, *25*(1), 353.

Delfgaauw, J., Dur, R.A., & Souverijn, M. (2017). Team incentives, task assignment, and performance: a field experiment. *Tinbergen Institute Discussion Papers*, 17-090/VII, 1-31.

Deli, I., & Çağman, N. (2015). Intuitionistic fuzzy parameterized soft set theory and its decision making. *Applied Soft Computing*, *28*, 109-113.

Deli, I., & Karataş, S. (2016). Interval valued intuitionistic fuzzy parameterized soft set theory and its decision making. *Journal of Intelligent & Fuzzy Systems*, *30*(4), 2073-2082.

Do, Q.H., Vinh, N.Q., & Thinh, T.N. (2025). A hybrid fuzzy AHP-TOPSIS technique for open-source software selection. *Journal of Information Systems Engineering and Management*, *10*(25s), 158-171.

Enginoğlu, S., & Arslan, B. (2019). On intuitionistic fuzzy parameterized intuitionistic fuzzy soft sets and their application in decision-making. In *2019 International Conferences on Science and Technology-Natural Science and Technology* (pp. 78-94). ICONST. Prizren, Kosovo.

Enginoğlu, S., & Memiş, S. (2020). A new approach to the criteria-weighted fuzzy soft max-min decision-making method and its application to a performance-based value assignment problem. *Journal of New Results in Science*, *9*(1), 19-36.

Gau, W.L., & Buehrer, D.J. (1993). Vague sets. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(2), 610-614.

Goguen, J.A. (1967). L-fuzzy sets. *Journal of Mathematical Analysis and Applications*, *18*(1), 145-174.

Gradišnik, M., Beranič, T., & Karakatič, S. (2020). Impact of historical software metric changes in predicting future maintainability trends in open-source software development. *Applied Sciences*, *10*(13), 4624.

Grattan-Guinness, I. (1976). Fuzzy membership mapped onto intervals and many-valued quantities. *Mathematical Logic Quarterly*, *22*(1), 149-160.

Gulwani, S., Radiček, I., & Zuleger, F. (2014). Feedback generation for performance problems in introductory programming assignments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 41-51). ACM. Hong Kong, China.

Huang, B., Zhuang, Y.L., & Li, H.X. (2013). Information granulation and uncertainty measures in interval-valued intuitionistic fuzzy information systems. *European Journal of Operational Research*, *231*(1), 162-170.

Hwang, C.L., & Yoon, K. (1981). *Multiple attribute decision making: methods and applications: a state-of-the-art survey*. Springer-Verlag, Berlin, Germany.

Jiang, Y., Tang, Y., Chen, Q., Liu, H., & Tang, J. (2010). Interval-valued intuitionistic fuzzy soft sets and their properties. *Computers & Mathematics with Applications*, *60*(3), 906-918.

Klassen, A., Miller, A., Anderson, N., Shen, J., Schiariti, V., & O'Donnell, M. (2010). Performance measurement and improvement frameworks in health, education and social services systems: a systematic review. *International Journal for Quality in Health Care*, *22*(1), 44-69.

Love, K., Turner, S., Runger, G., Adams, C., & Riley, W. (2024). Patient assignment and quality performance: a misaligned system. *American Journal of Managed Care*, *30*(10), 482-487.

Maji, P.K., Roy, A.R., & Biswas, R. (2002). An application of soft sets in a decision making problem. *Computers & Mathematics with Applications*, *44*(8-9), 1077-1083.

Memiş, S. (2023). Picture fuzzy parameterized picture fuzzy soft sets and their application in a performance-based value assignment problem to salt-and-pepper noise removal filters. *International Journal of Fuzzy Systems*, *25*(7), 2860-2875.

Molnar, A.J., & Motogna, S. (2020). Longitudinal evaluation of open-source software maintainability. In *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering* (pp. 120-131). SCITEPRESS. Prague, Czech Republic.

Molodtsov, D. (1999). Soft set theory-first results. *Computers & Mathematics with Applications*, *37*(4), 19-31.

Moran, R., Keramati, M., Dayan, P., & Dolan, R.J. (2019). Retrospective model-based inference guides model-free credit assignment. *Nature Communications*, *10*(1), 750.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, *11*(5), 341-356.

Renshaw, P.S.J., Parry, E., & Dickmann, M. (2021). The organizational value of international assignments-the relational underpinning. *The International Journal of Human Resource Management*, *32*(1), 106-146.

Saaty, T.L. (1980). *The analytic hierarchy process*. McGraw-Hill, New York, USA.

Sahlin, J., & Angelis, J. (2019). Performance management systems: reviewing the rise of dynamics and digitalization. *Cogent Business & Management*, *6*(1), 1642293.

Saxena, P., Kumar, V., & Ram, M. (2022). A novel CRITIC-TOPSIS approach for optimal selection of software reliability growth model (SRGM). *Quality and Reliability Engineering International*, *38*(5), 2501-2520.

Schleicher, D.J., Baumann, H.M., Sullivan, D.W., Levy, P.E., Hargrove, D.C., & Barros-Rivera, B.A. (2018). Putting the system into performance management systems: a review and agenda for performance management research. *Journal of Management*, *44*(6), 2209-2245.

Sebastian, S., & Ramakrishnan, T.V. (2011). Multi-fuzzy sets: an extension of fuzzy sets. *Fuzzy Information and Engineering*, *3*(1), 35-43.

Shahar, N., Moran, R., Hauser, T.U., Kievit, R.A., McNamee, D., Moutoussis, M., Consortium, N.S.P.N. & Dolan, R.J. (2019). Credit assignment to state-independent task representations and its relationship with model-based decision making. In *Proceedings of the National Academy of Sciences* (vol. *116*(32), pp. 15871-15876). NAS, USA.

Singh, V., Kumar, V., & Singh, V.B. (2023). A hybrid novel fuzzy AHP-TOPSIS technique for selecting parameter-influencing testing in software development. *Decision Analytics Journal*, *6*, 100159.

Smarandache, F. (2018). Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic Sets and Systems*, *22*, 168-170.

Spitzer, M.W.H., Moeller, K., & Musslick, S. (2023). Assignment strategies modulate students' academic performance in an online learning environment during the first and second COVID-19 related school closures. *PLOS One*, *18*(5), e0284868.

Sulukan, E., Cagman, N., & Aydın, T. (2019). Fuzzy parameterized intuitionistic fuzzy soft sets and their application to a performance-based value assignment problem. *Journal of New Theory*, *29*, 79-88.

Torra, V. (2010). Hesitant fuzzy sets. *International Journal of Intelligent Systems*, *25*(6), 529-539.

Wu, J. Z., & Tiao, P.J. (2018). A validation scheme for intelligent and effective multiple criteria decision-making. *Applied Soft Computing*, *68*, 866-872.

Xu, Y., & Yeh, C.H. (2014). A performance-based approach to project assignment and performance evaluation. *International Journal of Project Management*, *32*(2), 218-228.

Xu, Z., & Yager, R.R. (2006). Some geometric aggregation operators based on intuitionistic fuzzy sets. *International Journal of General Systems*, *35*(4), 417-433.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353.

Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning-II. *Information Sciences*, *8*(4), 301-357.

Zhang, Y., Lu, X., Li, J., & Lin, C. (2026). Dynamic selection of cooperative partners of agricultural science and technology industry-university-research system based on field theory. *Kybernetes*, *55*(6), 2256-2283.

Zhu, P., & Wen, Q. (2010). Probabilistic soft sets. In *2010 IEEE International Conference on Granular Computing* (pp. 635-638). IEEE. California, USA.