Comparison of Multi-Criteria Decision-Making Methods for Online Controlled Experiments in a Launch Decision-Making Framework

Jie JW Wu, Thomas A. Mazzuchi, and Shahram Sarkani

Abstract

Context: User-intensive software systems such as Web and mobile applications are defined as systems that serve and interact with an increasingly large number of users. Inefficient launch decisions in user-intensive systems in domains such as social media, information retrieval and e-commerce can lead to dramatic loss in the goal metrics of these highly scalable applications, and therefore impact potentially billions of users.

Objective: Due to the complexity of user-intensive systems, engineers rely heavily on A/B testing (i.e., online controlled experiments) to evaluate and measure the impact of new changes. However, little attention has been paid to improve the empirical process of making launch decisions based on the A/B testing results. In this paper, we propose a framework to address this issue.

Method: We propose a Multi-Criteria Decision Making (MCDM) framework that uses A/B Testing results to provide launch decisions analysis, as a complementary tool to assist decision making. The framework includes modules for 1) configuration setup, 2) criteria weighting, 3) pairwise comparison between criteria and alternatives 4) analysis of alternatives using MCDM and produces launch decisions based on the A/B testing results.

Results: Experimental results from publicly available dataset that compares well-known and widely applied MDCM methods shows that a good combination of the Analysis of Alternative method (such as TOPSIS-Vector, MMOORA, and VIKOR) and Criteria Weighting method (such as Standard Deviation) in the framework led to more effective launch decision making.

Conclusion: We formulate the problem of launch decision making using A/B testing results and propose a MCDM based framework for it, as an imperfect first step to address this problem. The experiments suggest that MCDM methods such as TOPSIS, MMOORA and VIKOR may be effective at making launch decisions based on A/B testing results.

Index Terms—Controlled Experiment, A/B Testing, Multi-Criteria Decision Making, Design of Experiments

1 Introduction

USER-intensive systems [17] are defined as Web or mobile software applications that serve an increasingly large number of users (potentially billions of daily user traffic and user interactions), and span over different application domains such as social media, information retrieval, and e-commerce etc. A key distinguishing characteristic of user-intensive system is the live interactions with a large volume of users, who use the applications with various preferences and needs.

Due to this characteristic of high user traffic and user interactions, it is particularly important for user-intensive systems to ensure that new product deployments and releases maintain or improve the goal metrics, or stakeholder requirements such as user satisfaction, user retention, engagement metrics, or revenues etc. However, due to the fast iterations and complexity of the user-intensive applications, in practice, a product launch of a small change could potentially lead to significant losses in the key metrics of the system, such as user engagement metrics or revenues [22,48,49]. As a result, engineers rely heavily on A/B testing (i.e., online controlled experiments) to evaluate product changes of user-intensive applications. The goals of A/B testing include understanding how the product works, identifying bugs or heterogenous effects, and making launch decisions [50]. The stakeholders (or decision makers) benefit from these A/B testing practices that are essential for organizational decision-making [39]. In this research, we focus on how to improve the launch decision making process based on the A/B test of a product change.

• *Jie JW Wu, Thomas A. Mazzuchi and Shahram Sarkani are with Department of Engineering Management and Systems Engineering at the George Washington University, Washington, DC 20052. (E-mail: jiewu@gwu.edu; mazzu@gwu.edu; sarkani@gwu.edu).*

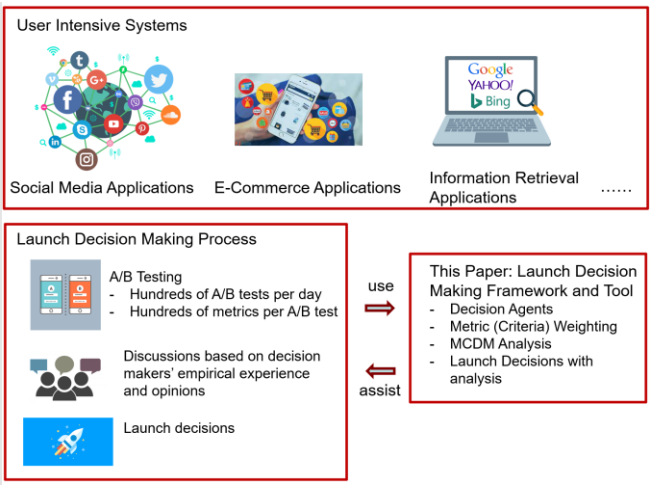


Fig. 1. The overall picture of the proposed framework/tool in the context of user-intensive systems and the launch decision process of user-intensive systems.

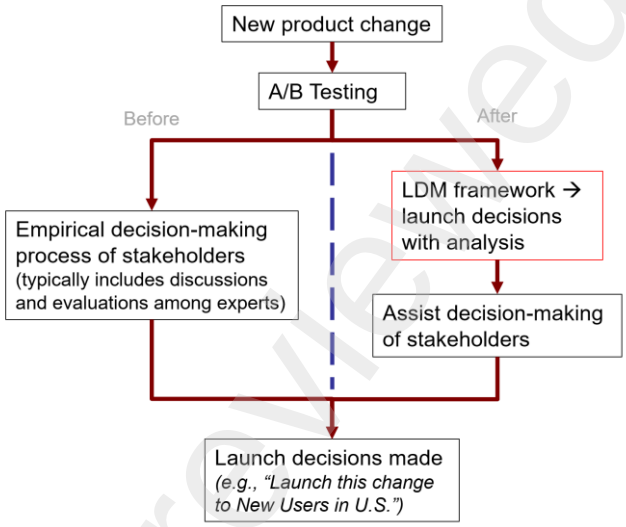


Fig. 2. The flow of product launch decision-making process using A/B testing, with "before" and "after".

A/B testing (also known as online controlled experiments, split tests, or randomized experiments) has become popular in user-intensive systems to collect the implicit user behavior and product effect of a given change. It serves different product variants to different users, then collects data related to the user behavior, and compares different variants of the product to the unmodified product. This allows the gathering of information for a small but sufficient number of users for stakeholders to make decisions on whether to launch the change to 100% of users [22].

More specifically, A/B testing is a term for an online controlled experiment with two variants, A and B, which are the control variant (unmodified product) and treatment variant (new change on the product) of a Web or mobile application. Behavioral, performance, and key metrics are collected, analyzed, and compared during the experiment period to evaluate the effect of the "A" or "B" variant on these metrics. A/B testing is frequently used in the product development at internet companies, including Facebook, LinkedIn, Pinterest (social media application), Google, Microsoft, Yahoo (information retrieval application), Amazon, eBay, Netflix, Uber, Airbnb (e-commerce application), and others. For example, the XLNT A/B testing platform in Linkedin [22] instilled A/B experimentation at Linkedin into the decision-making process and benefitted stakeholders, including roles inside or even outside R&D. Similar examples can be found at Microsoft [39], Facebook [37], Netflix [38] and Google [35]. On the frequency of usage, a report in 2017 shows that for several leading tech companies, including Microsoft, Amazon, Booking.com, Facebook and Google, conduct more than 10,000 A/B tests annually [48]. The number could be a few hundred annually at a mature company. The user behavior metrics in the social media and social network domain are hard to predict without A/B tests. This combined with the fast pace of development iterations in social media, has resulted in more than 400 concurrent A/B tests running daily in Linkedin [22] with large amounts of different metrics being logged. In the ideal case, all code changes that are pushed to production should be A/B tested first, even though the change looks transparent to the user. This is because the change may still have an unexpected impact on the user experience and user behaviors.

Web facing companies have certain forms of decision-making process for A/B testing in user-intensive systems [39]. Broadly speaking, they typically include 1) defining overall-evaluation criteria (OEC), or goal metrics, along with secondary metrics and guardrail metrics [18], 2) alerting, scorecards, and periodical diagnosis on these metrics [74], 3) multiple approvals and discussions with stakeholders or experts before shipping features via A/B testing [35]. As a quantitative measure of the experiment's objective, OEC is typically defined by a company as a single metric to guide decision-making at a high level and incorporates the tradeoff among metrics is essential for organizational decision-making [48], and it's possible to have multiple OECs in large organizations [39].

Despite the above decision-making process in companies, little attention has been paid to improve the empirical process of launch decision making based on the A/B testing results. For example, at Microsoft, it is reported in [47] that analyzing and discovering the A/B testing results insights by hand to make informed decisions can be cumbersome and challenging. At Netflix, A/B test results are used as an important and informative source for making product decision, and yet interpreting A/B tests results remains "partly art" [38]. At Google, a process is put in

place where experimenters bring their experiment results to discuss with experts and agree on whether the experiment is a positive or negative user experience for decision-makers to determine whether to launch this change [35]. Other examples in [52,53] mentioned that it's complicated to make decisions for A/B tests, due to the complexity of analyzing A/B tests and multiple roles involved such as product manager, data scientist, engineer, and stakeholder etc. In literature, there are very few approaches that try to automate the A/B testing [6][46] or develop analysis to improve effectiveness of A/B testing process [47]. Therefore, the identified problem is that the launch decision-making process of A/B tests is empirical and involves discussions and evaluations among experts [35,22].

Multi-Criteria Decision Making (MCDM) methods provide a formal approach to help decision makers improve analytic rigor, auditability, and conflict resolution of decision-makers [61,63,65]. It has been applied in a wide range of areas [7-14,75], and there are studies indicating that MCDM methods provide better rankings than intuitive approaches [73,76,77]. Therefore, the assumption of this work is that an automated decision-making framework can better assist, simplify and crosscheck the empirical decision-making process in A/B testing that typically involve multiple discussions and approvals from stakeholders and experts, especially with the increasingly large amount of A/B tests being conducted and multiple metrics being logged. To the best of our knowledge, in the literature, there is no generalized or principled decision framework that suggests launch decisions with analysis based on the A/B testing results.

To bridge the literature gap, in this paper, we propose a MCDM based framework, called LDM framework, for the *launch decision making* of A/B testing results in user-intensive systems. Our objective is to develop a complementary MCDM tool that provides the stakeholders with automatic decision analysis to assist, simplify and crosscheck the launch decision making process based on A/B testing results. To achieve this goal, we integrated, evaluated, and compared various criteria weighting methods and MCDM methods in the proposed LDM framework. Our contributions in this study are as follows:

1. We highlight the problem of launch decision making in user-intensive systems, provide its problem formulation and convert the problem to a multi-objective optimization formulation.
2. We propose a MCDM framework that uses A/B testing results to handle launch decision making and output launch candidates with confidence values. The framework includes modules for 1) framework configuration setup, 2) criteria weighting, 3) pairwise comparison between criteria and alternatives 4) analysis of alternatives using MCDM.
3. We compared and evaluated 6 Analysis of Alternative (MCDM) methods and 9 Criteria Weighting methods in the framework and demonstrated the merits of our framework on a public dataset including more than 4,800 A/B tests. Our results show that the framework can effectively suggest launch decisions using the A/B testing results as input.

Paper Organization. The remainder of this article is organized as follows. Section 2 introduces the related work. Section 3 introduces the problem statement and formulation. Section 4 introduces the proposed LDM framework. Section 5 is about experiments and Section 6 provides the conclusion and future work.

2 Related Work

The literature review section is split into two parts: (a) A/B testing in user-intensive systems, (b) multiple criteria decision making.

2.1 A/B Testing

Historically, the concept of the controlled experiment theory was firstly introduced in the 1920s by Sir Ronald A. Fisher's experiments at the Rothamsted Agricultural Experimental Station in England. The controlled experiment can be broadly divided into 1) offline experiments, which has been studied and developed very well in the field of Statistics [15], and 2) online experiments. Online controlled experiments, also known as A/B test, split test, randomized test, began to gain increasing usage in the late 1990s with the rise of the Internet. With the increasing popularity of Web or mobile user-intensive applications, A/B tests have become a gold standard in internet companies for evaluating new product changes [22, 50]. Today, well-known internet companies, such as Amazon, Bing, Facebook, Google, LinkedIn, and Yahoo! etc., run thousands of A/B tests weekly to test the effect of changes to the applications. A/B testing is now considered a crucial tool [50] for user-intensive systems, and it has also been used by startups and smaller websites more frequently [51].

Outside of online or web domain, the experimentation practice in A/B testing has gained attentions in other domains, such as embedded system domain [71] and automative software engineering domain [72]. In embedded systems, the concept of DevOps adoption, including continuous deployment and A/B testing practices is introduced [71], but certain challenges still exist for DevOps adoption, such as hardware dependency, limited user size and visibility of customer environments, scarcity of tools and lack of feature usage data. Recently, a design method and a corresponding case study are presented to conduct A/B testing in automotive embedded software to address the problem of limited user size [72]. This suggests that A/B test practices in embedded software are in an early stage, and there hasn't been any study that reports extensive adoption of A/B testing practices at organizational levels [42].

The academic literature on A/B test started in 2007. Kohavi et al. [36] initiated the academic discussion in 2007 on A/B test by describing the experience of controlled experiments at Microsoft with guidelines. Later, other

well-known tech companies, Facebook [37] and Netflix [38], started to use data-driven decision making [39], and described their experiences about experimentation to the research community. After more than ten years of research, there has been a couple of works on different problems in this topic, such as 1) the experimentation process definition [41], 2) building infrastructure for large-scale controlled experimentation [22], 3) metrics selection and development [40], 4) summary of knowledge and challenges in continuous experiments [42] and 5) analysis of the heterogeneous treatment effect (HTE) in A/B tests [55].

The benefits of A/B testing are categorized in the following 3 aspects [42]: 1) at the portfolio level, the business and engineering impact of changes can be measured for company-wide product portfolio development [49]. 2) at the product level, with unnecessary features removed, the products get incremental quality improvements and reduced complexity [86]. 3) at the team level, the findings and learnings from the A/B tests support the R&D teams to prioritize their development activities, and team goals can be expressed in terms of metric changes, with measurable progress being tracked. Specifically, the roles in R&D team that benefit from the A/B testing practice include

1. Business analyst, product designer, engineering manager (in ideation phase)
2. Software developer, quality assurance (in implementation phase)
3. Release engineer, operations engineer (in execution phase)
4. Data scientist, user researcher (in experiment design and analysis phase)

Besides the above roles within R&D team, roles outside R&D team may also benefit from A/B testing, with an example that the finance managers also use and bake A/B test results into business forecasting [22].

The challenges of A/B testing when being adopted and used are categorized in the following aspects [42]: 1) Cultural, organizational and managerial challenges in changing the organizational culture to embrace data-driven decision making and experimental skills [22]. 2) Business challenges in defining metrics to measure business value [18]. 3) Technical challenges to ensure efficient experimentation. 4) Statistical challenges such as exogenous effects and endogenous effects. 5) Ethical challenges when user data is involved. 6) Domain specific challenges such as adoption of A/B tests in embedded systems [71,72], social media, e-commerce and cyber-physical systems.

With A/B tests, one can conduct online experiments on real users to answer the question "If a specific change is introduced, will it improve the goal metrics?". Given a certain metric in A/B test, we often use hypothesis-testing procedures to produce a p-value, which is used to decide whether the treatment variant has an effect or not compared with the control variant. Thus, the decision making is easy in toy cases where only one metric or very few metrics are considered. However, the A/B tests being conducted in industry in practice typically involve multiple metrics. For example, the A/B test platform at Microsoft runs analysis to check hundreds to thousands of metrics [48], and over these years, Bing created more than 6,000 metrics, which are grouped by different sub-areas for experimenters to use such as Web search, image search, Ads, video search [48][49]. For the decision-making framework in the field of software engineering, there have been approaches [26] that make software product decisions based on decision making methodologies. Also, there have been methods [45] to prioritize user-session based test suites for Web applications to improve reliability, but there is no framework that explicitly uses A/B testing results to make decisions.

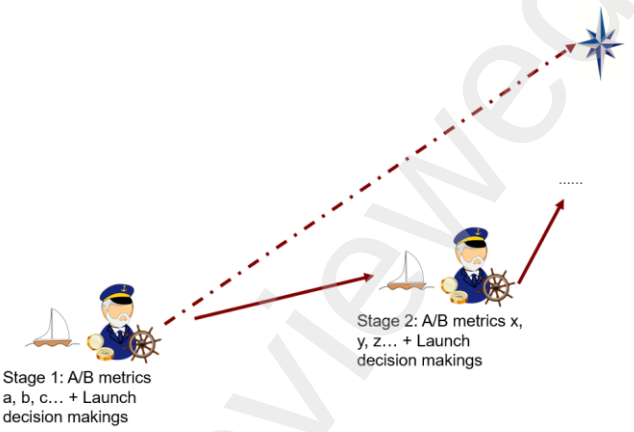


Fig. 3. The choice of online metrics, as well as the launch decision making process in A/B test evolves as the service (the ship) grows over time toward the North Star. When the service is at an earlier stage, metric a, b, c… are used as the key metrics for launch decision making. Another set metric x, y, x… are used in decision making when the service is in a more matured stage, closer to the North Star.

To the best of our knowledge, in the literature, there is no generalized or principled decision framework that suggests launch decisions with analysis based on the A/B testing results.

2.2 Multi-Criteria Decision Making

A typical engineering project usually involves multiple stakeholders and different criteria, so a decision analysis and a decision-making framework [43,44,80-85] are required in the engineering process [1,2]. When decision makers need to handle multiple criteria, which may be conflicting with each other, the decision-making process becomes more complicated. Historically, the concept of MCDM was firstly introduced in the 1700s by Benjamin Franklin [23]. The decision-making theories were first formally documented in the 1960s. With the development of MCDM there have been various methods and solutions to address the problem of decision-making.

MCDM can be divided into two parts: 1) multiple objectives decision making (MODM) and 2) multiple attributes decision making (MADM). MODM are problems that have an infinite number of possible alternatives, and their decision variables are limited by objectives and constraints. On the other hand, MADM problems typically have a small number of alternatives which are explicitly represented in terms of attributes. Examples of these techniques include the analytic hierarchy process (AHP) technique [7, 8], multi-attribute utility theory [9], Elimination and Choice

Expressing the Reality (ELECTRE) Methods [10-12], Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) [13], and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [14]. The MCDM problem can be broken down into 3 important subproblems: 1) stakeholder weighting, 2) criteria weighting, and 3) the evaluation of alternatives given the criteria weights. In this work, we tested, used, and evaluated different criteria weighting methods and analysis of alternative methods in the proposed framework.

Our framework is like the previous frameworks in the following way: our framework has 3 major components: Criteria Weighting, Pairwise Comparison, Analysis of Alternative, and they follow and are built on the components of the existing mainstream MCDM frameworks [7-14]. In our framework, subjective and objective methods such AHP, SMART (see the full list in Table 2) are used in Criteria Weighting component. Methods such as TOPSIS, VIKOR (see the full list in Table 3) are used in Analysis of Alternative component. This is accumulative knowledge of our framework. On the other hand, the new knowledge provided by our framework mainly comes from the particular focus on the launch decision making problem in the A/B testing domain. Specifically, our framework differs from the previous frameworks in several aspects: 1) Our framework has a configuration setup that selects A/B test metrics (criteria) and determine launch decision candidates (alternatives). 2) Besides traditional Criteria Weighting methods that need expert knowledge, the framework also uses objective Criteria Weighting methods that do not need input from experts. 3) The pairwise comparison comes directly from the A/B test result, instead of human expert. 4) Besides the limited decision candidates, the extended framework also provides solutions for infinite decision candidates. In our framework, these new elements due to the focus on the launch decision making of A/B testing are combined with the existing elements from the previous MCDM methods.

3 Problem Statement and Formulation

Assuming the A/B metrics are selected and defined by stakeholders already, how do we leverage the A/B testing result to generate launch decisions reliably? We first start by describing the key characteristics of A/B testing metrics, then we state the problem statement with an example, formulate, and cast it as a mathematical optimization problem.

3.1 A/B Testing Metrics Characteristics

In A/B testing, the online metrics are used typically as pointers to guide the decision making towards the North Star (the success of the business). Table 1 shows a typical result of an example A/B testing. There are different categories of A/B test metrics [18][74] such as:

1) *Business report driven metrics:* metrics defined based on the long-term goal and success of the business, such as Monthly Active Users (MAU), Revenue per User, etc.

2) *Simple heuristic-based metrics:* metrics defined based on the actual interaction between users and the user-intensive applications, such as Click-Through Rate (CTR).

3) *User behavior driven metrics:* metrics defined based on user behavior models to measure user experiences.

4) *Guardrail metrics:* metrics that help to guard against situations when the goal metrics (or OEC) may provide wrong signals.

5) *Debugging metrics:* metrics used to help understand the changes in important metrics, especially the goal metrics (or OEC), and how to interpret the changes.

Besides different types of metrics, Deng and Shi [18]

TABLE 1

Metric Category	**Metric Name**	**Absolute Change for *all users* in Treatment (over Baseline)**	**% Change for *all users* in Treatment (over Baseline)**	**% Change for *new users* in Treatment (over Baseline)**	**% Change for *users in US* in Treatment (over Baseline)**
Engagement Metrics	App Open	88.409→ 88.621	**+0.24% +/- 0.31% (p=0.015)**	**+ 0.40% +/- 0.51% (p=0.005)**	**+1.50% +/- 0.25% (p=0.007)**
Engagement Metrics	Time Spent	5,101.722→ 5,108.36	+0.13% +/- 0.30%	+0.25% +/- 0.50%	**+0.67% +/- 0.30% (p=0.003)**
Performance Metrics	Network Success	452.978→ 450.713	**-0.50% +/- 2.16% (p=0.010)**	-0.05% +/- 3.30%	-0.08% +/- 2.35%
Performance Metrics	Start Latency P50	1,169.787→ 1,169.294	-0.04% +/- 0.26%	+0.05% +/- 0.33%	+0.03% +/- 0.35%
...	...	...	...	...	…

A simplified example of the AB result in a typical AB test. In this example, a new Feed ranking algorithm is tested in the treatment group. The positive "App Open" metrics with small p-value and flat results in other metrics indicate this may be a potential launch candidate to ship. However, the negative "Network Success" metrics indicate there may be a regression in the new ranking algorithm that causes higher network errors. Besides the results for all users, we can also see the results for different user cohorts such as new user cohort, user cohort in a specific country (e.g., US) etc.

proposed the quality definition of the online metrics, especially goal metrics (or OEC), including:

1) *Directionality*: a high-quality metric should demonstrate positive direction when the A/B test positively impacts user experience and show the opposite direction when the test negatively impacts user experience.

2) *Sensitivity*: a high-quality metric should be sensitive to the user experience improvements.

Due to various types and varying quality measurement of the online metrics, the metrics are evolving over time and need to be updated to better serve the business goal of user-intensive systems, as shown in Figure 3. So, it is unrealistic to have one fixed MCDM method to be applied well to a given scenario for all time. To alleviate this problem, our proposed framework is designed as general as possible and allows us to quickly switch to different criteria weighting methods or MCDM methods for a given scenario.

3.2 Problem Statement

Suppose we conduct an A/B testing for a new product change in a treatment. Given the A/B test result, the problem for the stakeholders and decision makers is to decide whether to launch this change to production or not. More formally, we define an A/B test key metrics $S_1, S_2, \dots, S_n$ and get the A/B Test results for the treatment:

$$\{(m_1', CI_1, p_1), (m_2', CI_2, p_2) \dots, (m_n', CI_n, p_n)\}$$

where m_i' is the raw % change of treatment group over control group on key metric S_i, p_i and CI_i are the p-value and confidence interval size of m_i'. Next, we convert the results to

$$\{m_1, m_2, \dots, m_n\}$$

where m_i is normalized, and finished the multiple testing adjustment [50], so that m_i is set to 0 if it is not statistically significant after the multiple testing adjustment. Note that in this work, to simplify the comparison among different types of metrics, we assume the % of metric change m_i is normalized or scaled so that different metrics can be compared properly.

Given the A/B test results, the first question we want to address is a binary decision to decide:

1. whether we want to launch this new change or not.

Furthermore, if the answer is yes, the next question is to:

2. decide which user cohort we want to launch this to.

Besides the answer to question 1, we believe it is also helpful to get answers to question 2, because a product change or feature usually is not one-size fits all for all users: it could be only suitable for certain user cohorts. In addition, since there may be multiple launch decisions available with different levels of confidence from the decision algorithm, it would be helpful to get a score assigned for each decision for stakeholders to review.

3.3 Motivating Example

Let us illustrate the problem with a practical example. Suppose the product change that we conduct the A/B test for is a new Feed ranking algorithm used in the user-intensive application. Table 1 illustrates a typical example of A/B test results. In Table 1, we can see from the column "% *Change for* ***all users*** *in Treatment (over Baseline)*" that: the positive "App Open" metrics with small p-value and flat results in other metrics indicate this may be a potential launch candidate to ship. However, the negative "Network Success" metrics indicate either there is regression or an unexpected bug from the new ranking algorithm in the treatment that causes higher network errors. If the network error metric is a critical top-level metric, this could be a launch blocker of this A/B test. At this point, discussions regarding launch decision-making would typically happen that involve roles including product, engineering and data science (see more empirical examples of this type of discussion at Google, Microsoft and Linkedin [22,35,39]). In this example, we need a principled way to decide whether we want to launch this new Feed ranking change or not, based on the A/B test result.

Besides the option to launch it to all users in the treatment, we could have other options to launch it to a specific user cohort that shows optimal A/B results. Thus, the goal is to select a subset user group, x, from the entire user base Λ ($x \subseteq \Lambda$) that maximizes the gains of A/B metrics comparing the treatment group with the control group. In this example, this would be x = {new users (users who register within 30 days)}. We can get $+0.4\%$ improvement on App Open rate and flat results on Network Success from the column "% *Change for* ***new users*** *in Treatment (over Baseline)*". Suppose Network Success is a critical metric, if we find that for any other user group x', the corresponding Network Success is negative, and App Open rate < $+0.4\%$. It indicates that the new Feed ranking has a relatively large impact only on new users, and there's no regression on Network Success given the small volume of new users. Thus, we may decide to launch the new Feed ranking algorithm to the new user group x as the final launch decision.

3.4 Mathematical Formulation

We convert the problem statement to the mathematical formulation as follows. The goal is to maximize the function f, the positive benefit of launching Treatment t ∈ T ,the set of possible treatments, to user cohort x:

$$Max\ f(x,t) = Max\{m_1(x,t), m_2(x,t), \dots, m_n(x,t)\}$$

where m_i represents % of statistically significant change on Treatment t ∈ T over Control on A/B metric S_i for user cohort x; The sign may need to be flipped so that positive m_i means it has positive benefits for goals (or success of the experiment); x represents a certain user cohort, it could be empty ($x = \{\emptyset\}$), all users ($x = \{all\ users\} = \Lambda$), or any subset of all users ($x \subset \Lambda$). If $argmax_x f(x,t) = \{\emptyset\}$ for a given Treatment t, it means we should not launch this change,

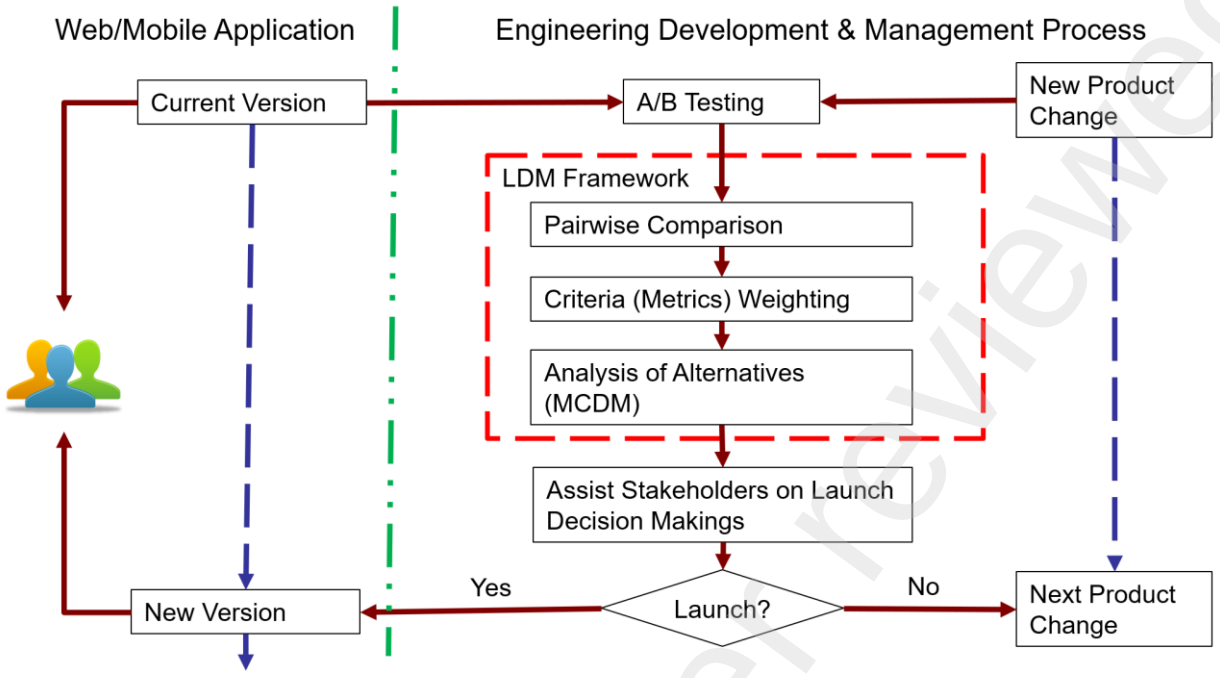


Fig. 4. Visual illustration of the proposed LDM framework in the context of engineering development process of the user-intensive applications. The evolution of Web/mobile user-intensive applications can be viewed as iterations of 1) new product change created; 2) A/B testing of the new product change; 3) launch decision making of the product change. The LDM framework produces launch decisions based on the A/B testing result to assist the decision making.

since no user cohort gets positive improvements on key metrics for Treatment t.

Given the math formulation as a multi-objective optimization problem, there are a few methods to address this. For example, a common technique to convert the multiple objectives into a single sum of weighted objectives, called weighted Goal Programming (by setting weights to each objective):

$$\max f(x,t) = \sum_{i}^{n} w_i \, m_i(x,t)$$

where $w_i \geq 0$ is the weight representing the importance of the A/B metric m_i. In this paper, we use MCDM to address this multi-objective optimization problem. The *criteria weighting* (determine the weights w_i) and *analysis of alternatives* (determine the final objective value f given the weights w_i and metrics change m_i) are two key components in MCDM methods, so they are critical to solving this multi-objective optimization. We describe them in the LDM framework in the next section.

4 The LDM Framework

The proposed LDM framework is used to produce a list of launch decisions for a productive change based on the A/B test results of this change. At a high level, the framework is composed of 1) the framework configuration, 2) the criteria weighting, 3) the pairwise comparison, and 4) the analysis of alternatives for either finite or infinite launch decision candidates. The output of the framework is whether to ship the A/B treatment variant or not, and additionally, the optimal user cohort to launch this change to. Figure 4 provides a visual illustration of the proposed framework in the engineering development process of the user-intensive applications. Figure 5 shows the stepwise representation of the framework. Figure 6 illustrates the pseudocode for this framework. We describe each element in detail in this section.

It should be noted that the framework is carefully designed so that the basis and motivations do not loss generality and are based on already established A/B testing fundamentals [50] and published empirical examples of A/B testing usage in industry [22,35,39,48]. Based on the observation from both a) the fundamentals of A/B testing [50] and b) published empirical reports of A/B testing platforms in different companies [22,35,39,48], we identify the problem that the launch decision-making process of A/B tests is empirical and involves discussions and evaluations among experts [35,22]. For this problem that serves as the basis, the motivation is that MCDM methods provide a formal approach to help decision makers improve analytic rigor, auditability, and conflict resolution [61,63,65]. The above basis and motivation guide the development of this framework.

4.1 Framework Configuration Setup

We need to set up the following items for us to use the framework:

4.1.1 Select A/B metrics (criteria) and A/B test configuration.

- Step 1. Pairwise Comparison : For each alternative, obtain the Pairwise Comparison matrix between criteria and the alternative $\{m_1(x,t), m_2(x,t), \dots, m_n(x,t)\}$, using the A/B test result.
- Step 2. Criteria weighting: obtain the weights (importance) of A/B metrics from pairwise comparison matrix in Step 1 (objective criteria weighting method). Alternatively, the weights can also be obtained from human expert judgements (subjective criteria weighting method).
- Step 3. Analysis of Alternatives (MCDM): calculate the score of each alternative, using the selected MCDM methodology

Fig. 5. Stepwise representation of the LDM framework to produce the launch decisions.

```
s = setting.getABMetrics()
t = setting.getABTreatment()
c = setting.getDeploymentDecisionCandidates()
w = calculateCriteriaWeighting()

// compute pairwise comparison matrix p
For i = 1 to length(s)
    For j = 1 to length(c)
      p_ij = m_i(c_j, t) = getPctChangeFromABResult(s_i, c_j)

getPctChangeFromABResult Results = runMCDMMethod(p, w, setting)
```

Fig. 6. Pseudocode for the LDM framework.

For any A/B test performed in a data-driven software company [48,49], typically we have hundreds of metrics in the A/B test result (e.g., Microsoft Bing created more than 6,000 metrics for online A/B tests [48]). However, only a few of these metrics in the particular domain of this new change are relevant and enough to explain the experiment hypothesis [54]. Thus, it is important for stakeholders and decision-makers to do a thorough review and carefully select a set of key A/B metrics from the specific domain as well as from the experiment hypothesis [54] for the launch decision making process. Therefore, the selected metrics should include both the top-level key metrics in this domain and the metrics we intend to move according to the experiment hypothesis. Figure 7 shows a simplified practical example of selecting key metrics for the domain of friending in the A/B test platform.

4.1.2 Determine Launch Decision Candidates (Alternatives).

Besides metrics, we should also determine the launch decision candidates (alternatives) C which are alternatives to be analyzed in the MCDM method. We categorize them in two types:

1. Limited Launch Decision Candidates:

This represents a finite list of alternatives selected as launch decision making candidates. In this case, the framework should output a ranked list of alternatives with corresponding scores as confidence. For example, if it is a binary launch decision problem (simplest case, such as considering only the option of shipping to all users in the motivating example in section 3.3), the launch decision candidate is either "ship" or "no ship", then $C = \{\emptyset, \Lambda\}$. Another example of more than 2 alternatives: besides "no ship" and "ship to all users", we may also include other user cohorts such as new users' cohort, users in U.S. Thus, the candidates $C = \{\emptyset, \Lambda, \{new\ users\}, \{users\ in\ US\}\}$.

Objective	Improve friending and user engagement
Test hypothesis	Add "people you may know" section with friend suggestions in the upper left corner of the page will allow users add more friends, therefore drive more user visits and user engagement.
Success metrics	Total friend adds Daily active user volume
Other key business metrics	Suggestion to communication rate New friendships with communications Suggestions viewed Chat sent …

Fig. 7. A simplified example of selecting key AB metrics for launch decision making in the friending domain in the A/B test platform [5]. For a typical product domain of the user-intensive application such as friending in this example, a small set of key metrics are selected by stakeholders carefully to measure the success of the features in this domain.

2. Infinite Launch Decision Candidates:

As illustrated in the motivating example, if we do not have a finite list of launch decision candidates, but instead are given a set of users in which each user has a few properties (such as age, country, user tenure, user's friend count, user's activeness, etc.), the potential launch candidates are infinite. This represents that the potential alternative could be infinite, so the framework should do optimization and thus produce optimal user cohorts with scores as launch candidates. We will discuss the solution of this scenario in section 4.4.

4.1.3 Update Framework Configurations from Feedback.

Finally, we should get feedback from stakeholders and periodically update the framework configurations to ensure the quality of the launch decision making framework. We should interact with stakeholders to 1) update the framework to reflect the new changes in the A/B metrics, 2) periodically collect feedback on all aspects of the application that can impact the goal metrics. It is also essential to include different types of stakeholders such as

1) technical domain expert who has deep knowledge and experience in the technical domain to provide engineering context.

2) data scientist and project manager who can provide in-depth information about the key A/B test metrics in the specific product domain.

3) leadership of the company to have more high-level information regarding different product areas and top priorities.

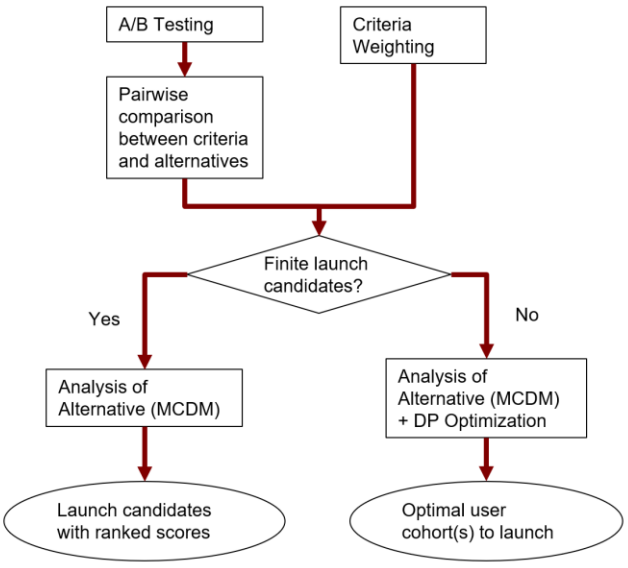


Fig. 8. the framework for finite (or binary) launch decision candidates and infinite launch decision candidates.

4.2 Criteria Weighting

The criteria weighting determines the weights or importance of the A/B test metrics, which is critical for the launch decision-making problem. Determining the weights is a key part of MCDM [4]. The weights can be calculated either objectively from information without decision makers' judgment [34], or subjectively from judgments of different stakeholders or decision makers [4]. Thus, we categorize the criteria weighting methods into these two types, as described in Table 3:

1) objective methods that calculate the weights from objective information (e.g., the pairwise comparison matrix) without human judgments.

2) subjective methods that use human judgments and combine weights of stakeholders, such as pairwise comparison (APH), SMART, etc.

We use objective methods in the framework, mainly because it is a standard way to calculate weights without the input from human/expert judgment, thus there is no need to handle scenarios where there is any change or change in quality for the A/B metrics, as discussed earlier. Also, since we only select relevant metrics as discussed in section 4.1.1, the weights are not affected by irrelevant metrics. However, with human expert input, subjective methods tend to perform better than objective methods, so we also tried the subjective methods in the experiments to compare and show evaluation for both subjective and objective methods, which are described in the experiment section.

4.3 Pairwise Comparison between Criteria and Alternatives

Given the criteria j (the j th A/B metric) and the alternative x (from the launch decision candidate list C), we can get the pairwise comparison result $m_j(x,t)$ from the A/B testing results on treatment t. For example, Table 1 shows the A/B results on several key metrics for all users. From this we can get the pairwise comparison value between all users and the A/B metric "App Open" is +0.24%. Different from traditional MCDM methods where human experts are usually involved in this step, we do not need human judgment for pairwise comparison in our framework, since we are able to get the pairwise comparison matrix from A/B testing results, which is more objective, and data-driven.

4.4 Analysis of Alternatives (MCDM) Given Criteria Weights

Given the input results from the criteria weighting and pairwise comparison between criteria and alternatives, we run the Analysis of Alternatives using any one of the MCDM methods listed in Table 2 and get the score for each alternative. We compare the performance of various Criteria Weighting methods and MCDM methods in the experimental section. According to the two types of launch decision candidates discussed in section 4.1.2, here we describe 1) framework for limited decision candidates and 2) the extended framework for infinite decision candidates:

4.4.1 Framework for Limited Launch Decision Candidates (Launch to All or Not)

The LDM framework is displayed in the flow of Figure 8 if we set the launch decision candidates as finite ($X = \{\emptyset, z_1 \dots z_e, \Lambda\}$) or binary ($X = \{\emptyset, \Lambda\}$). In this case, the framework produces a ranked list of launch decisions with scores. The launch decision with the highest score is chosen by default. The other launch decisions with lower scores may also be helpful to assist stakeholders in their decision making.

4.4.2 Extended Framework for Infinite Decision Candidates (Launch to Optimal User Cohort or Not)

As described in the motivating example, instead of a pre-defined limited list of user cohorts, we may choose to launch the new Feed ranking algorithm to an optimal user cohort, such as new user cohort, or users in US etc. In other words, if we do not have a finite list of launch decision candidates, but instead are given a set of users in which each user has a few properties (such as age, country, registration date etc.), the potential launch candidates are infinite. In this case, we propose to leverage the observation that the final output user cohort could be coarse-grained. Thus, our solution is to quantize the k properties of users and group users to a finite combination of user cohorts as launch decision candidates. In the earlier motivating example, suppose we are given users with their registration date and age, but without predefined launch decision candidates. In this case, we divide users into these cohorts by quantizing and splitting these 2 properties:

1) new users registered within 30 days, and younger than age 20,

2) new users registered within 30 days, and older than age 20,

3) old users registered over 30 days, and younger than age 20,

4) old users registered over 30 days, and older than age 20.

If we observe user cohort 1) has positive A/B testing result, while the other cohorts have neutral or negative A/B result, we would launch the feature in this A/B test to user cohort 1).

Recall that the objective we want to optimize is:

$$Max\ f(x,t) = Max\{m_1(x,t), m_2(x,t), \dots, m_n(x,t)\}.$$

We can quantize the otherwise arbitrary user cohort x to $x^* = \{u | P_j(u) \in [s_j, e_j], for\ \forall j \in [1,k]\} \subseteq \Lambda$, in which each user u has property 1...k, repr esented by $P_{1\dots k}(u)$. The property j of each user is bounded by $[s_j, e_j]$. If we choose a few limited s_j, e_j for each property j carefully, we could get a finite list of candidates $X^* = \{x^*\}$ to run the framework and produce the optimal user cohort(s) from X^* as output. This not only avoids the over-fitting user cohort output, but also makes this optimization solvable.

However, the candidates X^* could be a huge list, so that running the framework is computationally slow. In this case, we propose to use Dynamic Programming (DP) technique to compute pairwise comparison matrix that is both globally optimal and more efficient computation-wise than the brute-force solution. Specifically, we define:

$$m_i(x,t) = \frac{T_i(x,t) - C_i(x)}{C_i(x)} = \frac{T_i(x,t)}{C_i(x)} - 1$$

where $T_i(x,t)$ and $C_i(x)$ are the absolute value for A/B metric i in treatment variant and control variant (baseline) respectively. Since $T_i(x,t)$ and $C_i(x)$ can also be computed recursively using dynamic programming, thus $m_i(x,t)$ and the final objective $f(x,t)$ can also be computed together with $T_i(x,t)$ and $C_i(x)$. By leveraging dynamic programming, we can obtain the result of f(x, t) given any quantized x^* more efficiently. Figure 8 shows the framework that includes the DP optimization and the output of optimal user cohort.

5 Experiment Setup

In this section, we describe the experiment design and setup for evaluating the proposed launch decision making framework. The experiments were conducted on the Upworthy headline A/B tests dataset [27]. We first describe the Upworthy dataset, then we discuss about the criteria weighting methods, and the MCDM methods evaluated in the experiment, finally we introduce the Design of Experiments (DOE) of this experiment, and then the corresponding experimental results.

5.1 Outlined Research Process

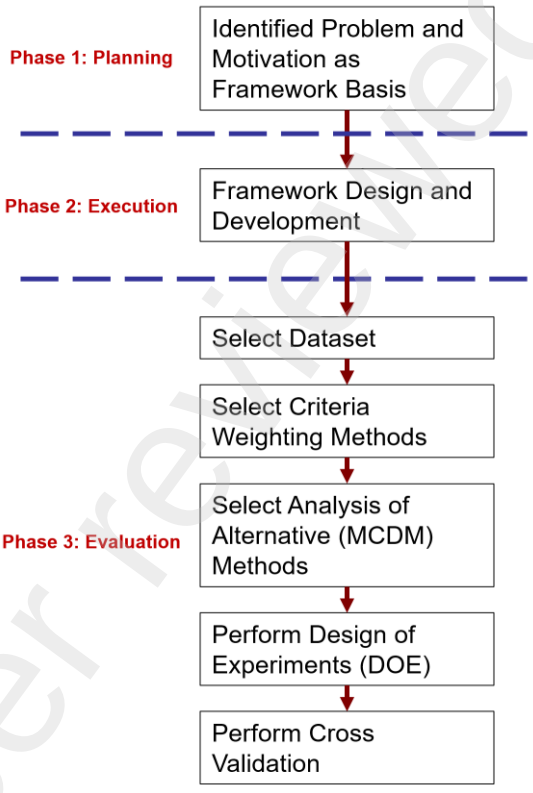


Fig. 9. Outlined research process of LDM framework in experiments.

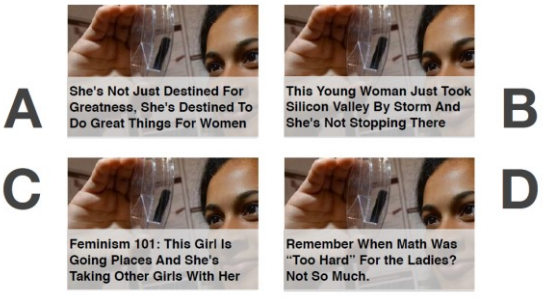


Fig. 10. An example of the A/B test that contains 4 packages (treatments): A, B, C and D in Upworthy dataset (Figure from the dataset website in [27])

Figure 9 shows the research process outlined in the experiments. The following sequence of activities are involved in this research to help the reader understand how this study is planned, executed and evaluated:

Identify Problem and Motivation as Framework Basis. In the planning phase of the research (phase 1), the problem or issue identified based on the literature gap and the motivation to address the problem need to be studied and discussed. Both the identified problem and the motivation serve as the basis and foundation of the framework. This forms the assumption of this work to be validated in the evaluation result of the framework.

Framework Design and Development. In the execution phase of the research (phase 2), the framework is designed and developed for the launch decision making problem in A/B testing. The designed framework is implemented and evaluated in experiments to measure its effectiveness. See

the detailed description in section 4.

Select Dataset. For the evaluation phase (phase 3), choosing the dataset that includes A/B test statistics is an important first step to evaluate the launch decision making in A/B tests. To ensure the evaluation results are statistically significant, a dataset with a high number of A/B test statistics is highly desired.

Select Criteria Weighting Methods. Criteria Weighting methods need to be selected to comparatively evaluate the impact of metric weights in the result of the framework.

Select Analysis of Alternative (MCDM) Methods. The Analysis of Alternative method, the major component of the framework, needs to be selected to comparatively evaluate their impact in the result of the framework.

Perform Design of Experiments (DOE). DOE is a systematic, efficient method to study the relationship between multiple factors and their responses. DOE is used in evaluation to determine the effect of different variants of Criteria Weighting methods and Analysis of Alternative methods.

Perform Cross Validation. Cross-validation is a resampling method to evaluate the results using different portions of the data. In the experiments, we performed cross validation to evaluate the generalizability and robustness of the framework.

5.2 Datasets

We used the Upworthy headline A/B tests dataset for this experiment. It has 4,873 A/B tests and 22,666 packages (treatments in the A/B tests) of headlines conducted by Upworthy from January 2013 to April 2015. In this dataset, each A/B test can include any number of packages (treatments), but most tests have four packages as shown in Figure 10. Each package in an A/B test mainly includes the following data:

1) created_at: time the package was created

TABLE 2
LIST OF CRITERIA WEIGHTING METHODS USED IN DESIGN OF EXPERIMENTS.

Abbreviation	Method	Category	Description	Reference
Mean Weight (MW)	Mean Weight method	Objective	Equal importance based on the assumption that all criteria are of equal importance.	[66]
Standard Deviation	Standard Deviation Method	Objective	Method that determines the weights of the criteria in terms of their standard deviations.	[66]
Statistical Deviation	Statistical Deviation method	Objective	Another objective weighting approach based on statistical variance of information.	[67]
Entropy	Entropy method	Objective	Method that assesses the weights using the entropy of the pre-defined decision matrix.	[68]
SMART	Simple Multi-attribute Rating Technique	Subjective	A process of rating of alternatives and weighting criteria.	[69]
Ranking Sum (RS)	Rank sum weights method	Subjective	Weights are computed from the individual ranks normalized by dividing the sum of the ranks.	[70]
Ranking Reciprocal (RR)	Reciprocal (or inverse) weights method	Subjective	Similar to the rank sum method except that the value is raised to an exponential of a parameter which is estimated by a decision maker as a result of the most important criterion.	[70]
Ranking Exponent (RE)	Rank exponent weight method	Subjective	Similar to the rank sum method except that it uses the normalized reciprocal of the criterion rank.	[70]
Pairwise Comparison (AHP)	Analytic Hierarchy Process	Subjective	One of the most commonly applied methods based on pairwise comparisons.	[7, 8]
SMART	Simple Multi-attribute Rating Technique	Subjective	A process of rating of alternatives and weighting criteria.	[69]
Ranking Sum (RS)	Rank sum weights method	Subjective	Weights are computed from the individual ranks normalized by dividing the sum of the ranks.	[70]
Ranking Reciprocal (RR)	Reciprocal (or inverse) weights method	Subjective	Similar to the rank sum method except that the value is raised to an exponential of a parameter which is estimated by a decision maker as a result of the most important criterion.	[70]

TABLE 3
LIST OF ANALYSIS OF ALTERNATIVE METHODS (MCDM) USED IN THE DESIGN OF EXPERIMENTS.

Abbreviation	Method	Category	Description	Reference
WSM	Weight sum method	Scoring method	The simplest available method, applicable to single-dimensional problems.	[33]
WPM	Weight product method	Scoring method	An alternative to the WSM, with the main difference being a product instead of a sum in the method.	[33]
TOPSIS-Linear	Technique for order preference by ideal solution	Distance to ideal point method	Technique based on the concept that the best alternative is the one which is closest to its ideal solution and farthest from the negative ideal solution, with the linear transformation of maximum as normalization procedure.	[32]
TOPSIS-Vector	Technique for order preference by ideal solution	Distance to ideal point method	Technique based on the concept that the best alternative is the one which is closest to its ideal solution and farthest from the negative ideal solution, with the vectorial normalization procedure.	[31]
MMOORA	Multi-objective optimization by ratio analysis	Distance to ideal point method	Technique based on the process of simultaneously optimizing two or more conflicting attributes (objectives) subject to certain constraints.	[29]
VIKOR	VlseKriterijumska Optimizacija I Kompromisno Resenje	Distance to ideal point method	Method that uses aggregating functions and focuses on determining compromising solutions for a prioritization problem with conflicting criteria.	[30]

2) test_week: week the package was created
3) clickability_test_id: test the package was in headline
4) eyecatcher_id: image ID (images unavailable)
5) impressions: # who viewed the package
6) clicks: # who clicked the package

In this experiment, the following metrics were used as the key evaluation metrics of the A/B tests: 1) #impression, 2) #clicks, 3) #click-through-rate, which was obtained by dividing #clicks over #impressions.

5.3 Evaluated Criteria Weighting Methods

Nine different Criteria Weighting methods were picked for the analysis of alternatives for experiments. Table 2 summarizes the Criteria Weighting methods used in the DOE. The first 4 are objective methods ("Mean Weight", "Standard Deviation","Statistical Deviation","Entropy"), and are included in the LDM framework. Additionally, we also tested and included 5 subjective methods ("AHP", "SMART", "Ranking-Sum", "Ranking-Reciprocal", "Ranking-Exponent") in the experiments to showcase the result for subjective methods, two experts in the field of software engineering and social media provided human judgments for the 5 subjective methods. We implemented these 9 methods following [34] to understand which ones are most suitable to be used in the LDM framework.

5.4 Evaluated Analysis of Alternative (MCDM) Methods Given the Criteria Weights

From a large number of possibilities, we have reviewed several methods and finally selected the following six MCDM methods for evaluation of the LDM framework during the experiments: 1) WSM [33], 2) WPM [33], 3) TOPSIS-Linear [32], 4) TOPSIS-Vector [31], 5) MMOORA [29] and 6) VIKOR [30]. These six methods have been selected, as they are the most widely applied in multi-criteria analysis problems given the criteria weights (either subjective weights or objective weights) as input for different applications [63][64][65]. Once given the criteria weights as input, these methods can be performed without interactive participation of decision-makers. Table 3 summarizes the Analysis of Alternative methods

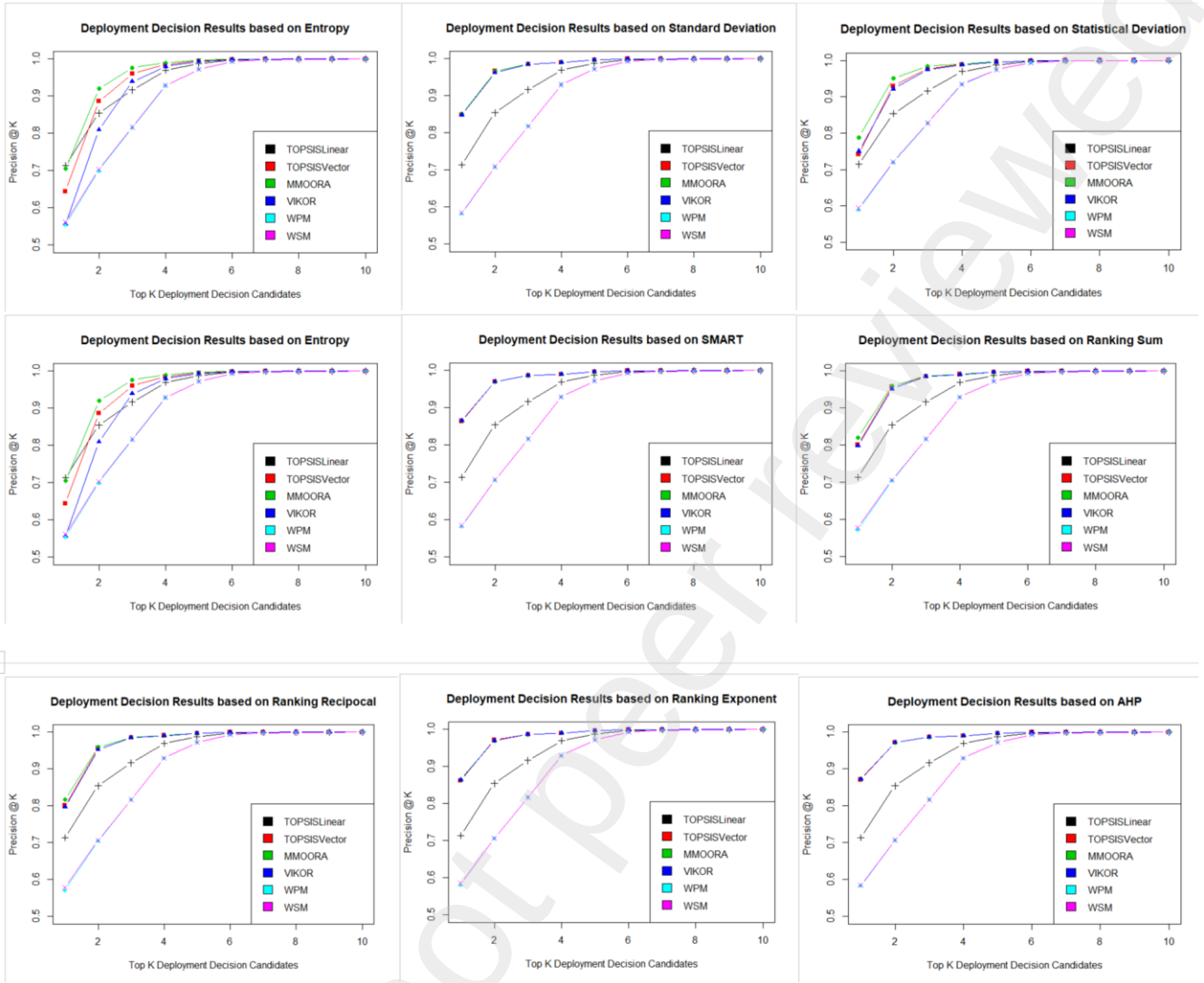


Fig. 11. Launch decision results of varying Analysis of Alternative method (MCDM) while the Criteria Weighting method was held constant.

used in the DOE. In the next paragraphs, a brief review is given with indicative applications of each of them in the literature.

WSM and WPM. WSM is a commonly used MCDM method, a straightforward decision-making method applied to one-dimensional problems and is usually considered as the first option for application. The WPM is similar to WSM but uses multiplication instead of addition [59]. Despite the disadvantages of WSM and WPM, i.e., sensitivity to units' ranges and exaggeration of specific scores, there are numerous applications in the literature that employ either of them primarily due to their straightforward implementation. Both WSM and WPM are easy to use and well understandable, well-proven technique, applicable when exact and total information is collected, providing good performance when compared with more sophisticated methods [58].

TOPSIS-Vector and TOPSIS-Linear. The TOPSIS method is a popular approach to MCDM and has been widely used in the literature. TOPSIS was first developed by Yoon [60] for solving a MCDM problem. Hwang and Yoon [31] expanded it further on the use of TOPSIS when assessing applications of MCDM. TOPSIS assumes that each attribute has a tendency of monotonically increasing or decreasing utility [61]. Therefore, it is easy to locate the ideal and negative-ideal solutions. "A relative advantage of TOPSIS is the ability to identify the best alternative quickly" [62]. Vector normalization is required in TOPSIS [60] (referred to as TOPSIS-Vector) solve multidimensional problems which can be considered as weakness of the method. So, we added and tested another variant called TOPSIS-Linear [32] in this experiment. TOPSIS-Linear is essentially the TOPSIS method with the linear transformation of maximum as normalization procedure.

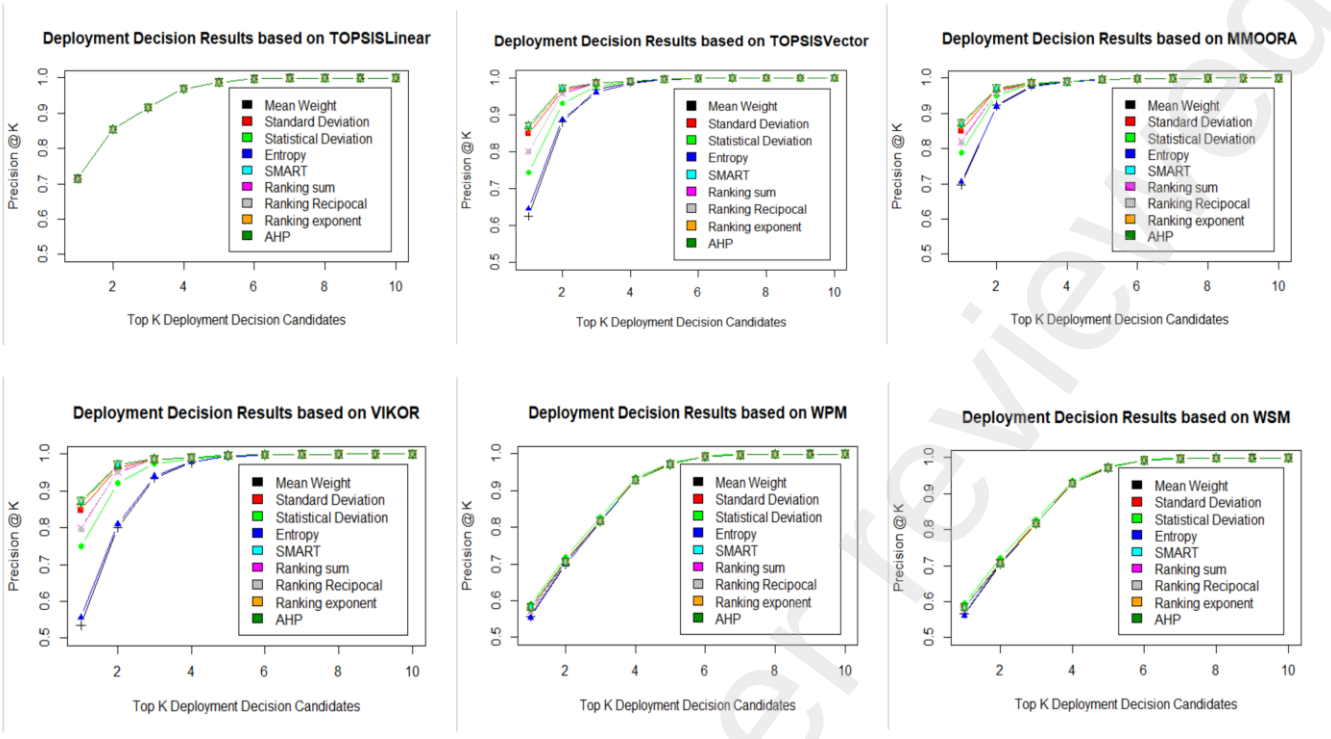


Fig. 12. Launch decision results of varying Criteria Weighting method, while the Analysis of Alternative method (MCDM) was held constant.

MMOORA. As a relatively new MCDM method with increasing applications, MMOORA is the process of simultaneously optimizing two or more conflicting attributes (objectives) subject to certain constraints. MMOORA, first introduced by Brauers [29], is such a multi-objective optimization technique that can be successfully applied to solve various types of complex decision-making problems in the manufacturing environment. MMOORA method is mathematically very simple, systematic comprehendible, and computationally efficient.

VIKOR. The VIKOR method is developed to solve MCDM problems with conflicting and non-commensurable (attributes with different units) criteria, assuming that compromise can be acceptable for conflict resolution, and when the decision-maker wants a solution that is the closest to the ideal solution, the alternatives can be evaluated according to all established criteria [30].

5.5 Design of Experiments

We performed Design of Experiment (DOE) to understand the decision-making quality of the LDM framework with respect to different criteria weighting methods, and analysis of alternative (MCDM) methods. Given the A/B test results, we computed the accuracy of the launch decision results predicted by the proposed LDM framework. We designed experiments with varying criteria weighting method and analysis of alternative method used in the LDM framework. The first 9 experiments were designed such that the criteria weighting methods were held constant, while the Analysis of Alternative method varied differently. The next 6 experiments were designed such that the Analysis of Alternative methods were fixed, while the Criteria Weighting methods varies. As mentioned before, six different Analysis of Alternative methods were chosen and used in LDM framework for evaluation during the experiments. Nine different Criteria Weighting methods were picked for the analysis of alternatives for experiments.

Given an A/B test result in the Upworthy dataset, the LDM framework first calculates the weights produced by the Criteria Weighting method, then we compute the pairwise comparison results between criteria and alternatives based on the A/B test result. Finally, we run the Analysis of Alternative method based on the weights and the pairwise comparison results. The output of the LDM framework is a ranked list of packages, representing the suggested launch decisions. The packages in an A/B test represent alternatives. For all packages in each A/B test, we select one package with smallest X1 value (an attribute for packages in the dataset) as "control" group, and the other packages as "treatment" groups. If the output is the control group, it means the LDM framework suggested "no ship" as the launch decision. We calculate the accuracy of LDM framework by comparing the output of launch decision-making with the actual winning treatment for each A/B test, provided by the dataset. For each experiment, the following plots were generated to illustrate the performance of launch decision results in Figure 11 and Figure 12. Each curve in the plot represents the Precision @ K, meaning whether the winning treatment is among the top K output of the LDM framework. Note that we did not test the extended framework to compute optimal user cohort since the user-level data is unavailable for this dataset.

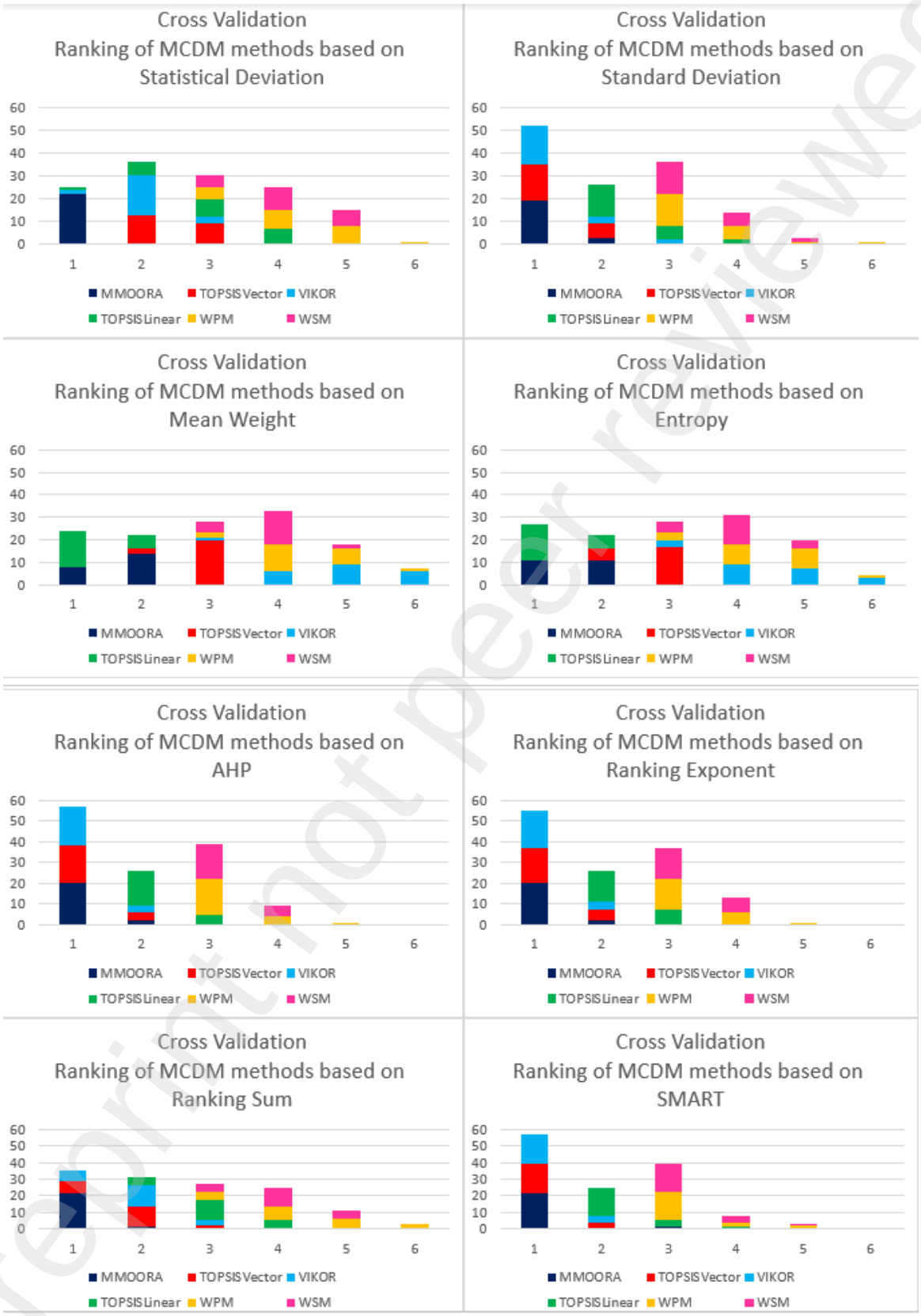


Fig. 13. The 22-fold cross validation results of 6 MCDM methods on the 22k packages in the Upworthy dataset. The dataset is split into 22 folds sequentially, 1k packages per fold. For each fold, 6 MCDM methods are ranked based on the accuracy of the predicted launch decisions compared with the groundtruth. In the figure, x-axis represents the ranking positions of the 6 MCDM methods, and y-axis represents the number of ranked results for each MCDM method. From the results, MMOORA, TOPSIS-Vector and VIKOR has better accuracy for this task than WPM, WSM and TOPSIS-Linear.

TABLE 4
RESULTS OF CRITERIA WEIGHTING METHODS USED IN THE DESIGN OF EXPERIMENTS.

List of Criteria Weighting Methods	Classification of Weighting Methods	Weight 1	Weight 2	Weight 3
Mean Weight	Objective	0.333	0.333	0.333
Standard Deviation	Objective	0.467	0.063	0.469
Statistical Deviation	Objective	0.49	0.02	0.49
Entropy	Objective	0.352	0.299	0.348
SMART	Subjective	0.333	0.061	0.606
Ranking Sum	Subjective	0.333	0.167	0.5
Ranking Reciprocal	Subjective	0.27	0.18	0.55
Ranking Exponent	Subjective	0.286	0.071	0.643
AHP (Pairwise Comparison)	Subjective	0.221	0.05	0.729

6 DISCUSSIONS

In this section, we first analyze and interpret the results. Then we describe the cross-validation result that measures the generalizability and robustness of the evaluation results. Finally, we discuss the threats of validity.

6.1 Interpreting results

Figure 11 shows noticeable changes in precision due to different Analysis of Alternative methods. No matter which Criteria Weighing method we pick, the results for WSM, WPM and TOPSIS-Linear are of the lowest precision overall. The MMOORA, TOPSIS-Vector and VIKOR achieved the top 3 in precision, with relatively small difference. This result illustrates that the Analysis of Alternative method is critical for the performance of the LDM framework. Using the DOE to change the Analysis of Alternative demonstrates how sensitive and critical the Analysis of Alternative method is within the LDM framework.

Figure 12 shows the difference of precision from modifying the Criteria Weighting method. From the plots, the precisions stay mostly the same for TOPSIS-Linear, WPM and WSM with different Criteria Weighting methods. No matter which Criteria Weighting method we use, the precision of WPM and WSM stays below 60% when K=1. However, the adjustments resulted in significant precision changes in TOPSIS-Vector, MMOORA, and VIKOR. For these 3 MCDM methods, Standard Deviation is the top performing objective Criteria Weighting method, with the precision of ~ 85%. AHP is the top performing subjective method, with precision of ~87%. On the other hand, Entropy and Mean Weight achieved the lowest precision (ranked 8th and 9th respectively). Table 4 shows the weight results of the Criteria Weighting methods in the experiment. Note that the weights of Standard Deviation, Statistical Deviation and Entropy may change due to different pairwise comparison matrix as input.

6.2 Cross Validation Result Analysis

We performed cross validation to evaluate the generalizability and robustness of the framework. We split the 22k packages in the Upworthy dataset into 22 folds (each fold has 1k packages), then rank the 6 MCDM methods for each fixed Criteria Weighting method, based on accuracy of each fold. The results were shown in Figure 13. We also summarized the accuracy (mean and standard deviation) of different combination of Criteria Weighting and MCDM methods from these 22 folds in Table 5.

From the results, MMOORA performs the best and is constantly ranked top 3 positions. Overall, MMOORA, VIKOR and TOPSIS-Vector appear to be the top 3 MCDM methods. For objective criteria weighting methods, MMOORA ranks the best. For entropy and mean weight methods which have very low accuracy, TOPSIS-Linear achieves the best results. For the subjective criteria weighting methods, MMOORA, VIKOR and TOPSIS-Vector have much better accuracy than WPM, WSM and TOPSIS-Linear.

6.3 Threats of Validity

Internal Validity. This threat relates to our assumption that the packages in each A/B test are labeled correctly in the Upworthy dataset [27,28,78]. However, mislabeling could happen. Still, we believe that MCDM method such as MMOORA, VIKOR and TOPSIS-Vector help the launch decision making process, as demonstrated in the experiments of Upworthy dataset.

Another threat to internal validity considers the internal parameters (e.g., MCDM algorithms [57]) that could potentially affect the results. In the experiments, internal validity threats might occur since only one set of configuration settings for the MCDM algorithm is conducted. To mitigate this threat, we used the default parameter settings for all the MCDM methods in our experiment.

TABLE 5
ACCURACY OF DIFFERENT CRITERIA WEIGHTING METHODS AND MCDM METHODS IN THE 22-FOLD CROSS VALIDATION. TOP 3 RESULTS FOR EACH ROW ARE MARKED AS BOLD.

Criteria Weighting Method	TOPSIS Linear Mean (SD)	TOPSIS Vector Mean (SD)	MMOORA Mean (SD)	VIKOR Mean (SD)	WPM Mean (SD)	WSM Mean (SD)
Mean Weight	**59.70% (0.037)**	**62.18% (0.033)**	**69.48% (0.024)**	53.24% (0.029)	55.70% (0.029)	56.67% (0.030)
Standard Deviation	72.32% (0.039)	**85.14% (0.029)**	**85.29% (0.034)**	**85.03% (0.032)**	58.22% (0.030)	58.55% (0.030)
Statistical Deviation	68.76% (0.033)	**74.17% (0.025)**	**78.86% (0.023)**	**74.95% (0.020)**	59.08% (0.032)	59.48% (0.032)
Entropy	**58.87% (0.036)**	**64.23% (0.035)**	**70.47% (0.025)**	55.48% (0.030)	55.42% (0.029)	56.22% (0.029)
SMART	72.38% (0.032)	**86.55% (0.022)**	**86.71% (0.024)**	**86.62% (0.020)**	58.18% (0.031)	58.50% (0.032)
Ranking Sum	67.13% (0.027)	**80.06% (0.019)**	**82.01% (0.022)**	**79.77% (0.026)**	57.13% (0.030)	57.84% (0.030)
Ranking Reciprocal	67.12% (0.026)	**79.98% (0.022)**	**81.66% (0.020)**	**79.58% (0.027)**	57.08% (0.030)	57.85% (0.030)
Ranking Exponent	72.08% (0.032)	**86.34% (0.022)**	**86.63% (0.023)**	**86.47% (0.021)**	58.11% (0.030)	58.52% (0.031)
AHP	73.04% (0.030)	**87.17% (0.020)**	**87.32% (0.021)**	**87.34% (0.020)**	58.32% (0.031)	58.54% (0.031)

External Validity. This relates to the generality of the effectiveness of MCDM method on other A/B testing evaluation. In this work, we have reduced this threat by performing a 22-fold cross validation on the Upworthy dataset [27] and demonstrated the effectiveness of MCDM methods such as MMOORA, VIKOR and TOPSIS-Vector. Therefore, these MCDM methods can potentially be adapted to work for other A/B testing as well. However, since we have not tested this, we cannot make a sound claim regarding the efficacy of these MCDM methods on another dataset. In our future work, we plan to create new datasets with larger A/B testing results for more comprehensive evaluation of MCDM methods.

Another threat to external validity considers the implementation of the MCDM methods used in the evaluation. In the comparative analysis, for MCDM methods, we directly use the library from MCDM Package [57] in R. For criteria weighting methods, we reimplement our own versions following [34]. From the evaluation results and previous test results, we believe that our implementation reflects the original methods.

Construct Validity. In the design of experiment, we adopted Precision@K to assess the performance of the MCDM methods. Precision@K and Recall@K are commonly used in other studies that investigate the decision-making results [79], and we also use this in our evaluations.

Conclusion Validity. This relates to the potential risk that evaluation results in the experiments being not statistically significant. To mitigate this risk, in the 22-fold cross validation, we computed the mean and statistical deviation of the accuracy from the 22 folds.

7 CONCLUSIONS AND FUTURE WORK

With the rapid changes in the internet industry and user-intensive system, a generalized framework is required to assist the decision makers in making launch decisions using data-driven A/B testing results. We presented the problem of launch decision making and proposed a MCDM based framework for it using A/B testing results. As an extension of the framework, DP is used to produce launch decisions with optimal user cohort more efficiently. Experiments and comparison of various MCDM methods on the selected A/B test dataset showed the effectiveness of the proposed LDM framework. The DOE also showed the sensitivity and importance of the Analysis of Alternative and Criteria Weighting used in the LDM framework. The experiments suggest that a good combination of the Analysis of Alternative method (such as TOPSIS-Vector, MMOORA, and VIKOR) and Criteria Weighting method (such as Standard Deviation) in the LDM framework led to more efficient launch decision making.

From a practical standpoint, in the motivating example, the LDM framework would provide recommended launch solutions with confidence scores. This aids and crosscheck the stakeholders' decision-making process. As the launch decision making gets more automated and more accurate, the stakeholders could start to gradually rely more on this automated decision-making results instead of analyzing and deciding by themselves.

Future work can focus on using other decision-making

techniques such as 1) more advanced MCDM/MODM methodologies, 2) machine learning method, 3) sensitivity analysis 4) treating different types of metrics (such as OEC, guardrail metrics, debugging metrics etc.) differently to further improve the decision-making quality, sensitivity, robustness of the LDM framework. Another potential extension to this research is to improve the evaluation process of the LDM framework for the user-intensive systems, such as 1) introduce A/B testing dataset with user level attributes, so that the case of infinite launch decision candidates can be evaluated as well, 2) conduct industrial validation of the LDM framework through a practical case study. Another potential application of this work is the automatic recommendation of A/B test launch decisions [56] when the experiment maturity is high enough.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their constructive feedback and suggestions that greatly improved our paper and research.

REFERENCES

[1] INCOSE. INCOSE System Engineering Handbook. INCOSE, 4 edition, 2015.

[2] A.Sage and J. Armstrong, Introduction to Systems Engineering. Hoboken, NJ, USA: Wiley, 2000, ser. Wiley Series in Systems Engineering.

[3] Louviere, J. J. "Why stated preference discrete choice modeling is NOT conjoint analysis (and what SPDCM is?)." Memetrics white paper 1 (2000): 1-11.

[4] Shukla, Vikas, Guillaume Auriol, and Keith W. Hipel. "Multicriteria decision-making methodology for systems engineering." IEEE Sytems Journal 10.1 (2014): 4-14.

[5] https://conversion.com/blog/how-to-measure-a-b-tests-for-maximum-impact-and-insight/

[6] Tamburrelli, Giordano, and Alessandro Margara. "Towards automated A/B testing." International Symposium on Search Based Software Engineering. Springer, Cham, 2014.

[7] T. Saaty, The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation. New York, NY, USA: McGraw-Hill, 1980.

[8] T. Saaty, "How to make a decision: The analytic hierarchy process," Eur. J. Oper. Res., vol. 48, no. 1, pp. 9–26, Sep. 1990.

[9] R. Keeney and H. Raiffa, DecisionsWith Multiple Objectives: Preferences and Value Trade-Offs. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[10] B. Roy, "Classement et choix en présence de points de vue multiples (la méthode electre)," Riro, vol. 2, no. 8, pp. 57–75, 1968.

[11] B. Roy, "Electre iii: Un algorithme de classement fondé sur une représentation floue des préférences en présence de critères multiples," Cahiers du CERO, vol. 20, no. 1, pp. 3–24, 1978.

[12] B. Roy and P. Bertier, La méthode electre ii(une application au médiaplanning) 1973.

[13] J.-P. Brans and B. Mareschal, "Promethee methods," Multiple Criteria Decision Anal., State Art Surveys, pp. 163–186 2005.

[14] C.-L. Hwang et al., Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey, vol. 13. New York, NY, USA: Springer-Verlag, 1981.

[15] Box GEP, Hunter JS, Hunter WG (2005) Statistics for experimenters: design, innovation, and discovery. Wiley, Hoboken

[16] L. A. Ocampo and E. E. Clark, "A comprehensive evaluation of sustainable manufacturing programs using analytic network process (Anp)," Multiple Criteria Decision Making, vol. 9, pp. 101–122, 2014.

[17] Ghezzi, Carlo, et al. "Mining behavior models from user-intensive web applications." Proceedings of the 36th International Conference on Software Engineering. 2014.

[18] Deng, Alex, and Xiaolin Shi. "Data-driven metric development for online controlled experiments: Seven lessons learned." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.

[19] INCOSE, "Systems engineering vision 2025," Int. Council Syst. Eng., vo. 327, no. 5970, pp. 1183–1183, 2014. [Online]. Available: http://doi.org/10.1126/science.327.5970.1183-d

[20] Madni, Azad M., and Scott Jackson. "Towards a conceptual framework for resilience engineering." IEEE Systems Journal 3.2 (2009): 181-191.

[21] Wollmann, Dewey, and Maria Teresinha Arns Steiner. "The strategic decision-making as a complex adaptive system: a conceptual scientific model." Complexity 2017 (2017).

[22] Xu, Ya, et al. "From infrastructure to culture: A/b testing challenges in large scale social networks." Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015.

[23] Franklin, Benjamin. "Letter to joseph priestley." Reprinted in the Benjamin Franklin Sampler (1956).

[24] De Montis, Andrea, et al. "Assessing the quality of different MCDA methods." Alternatives for environmental valuation 4 (2004): 99-133.

[25] Kodikara, Prashanthi N., B. J. C. Perera, and M. D. U. P. Kularathna. "Stakeholder preference elicitation and modelling in multi-criteria decision analysis–A case study on urban water supply." European Journal of Operational Research 206.1 (2010): 209-220.

[26] Büyüközkan, Gülçin, and Da Ruan. "Evaluation of software development projects using a fuzzy multi-criteria decision approach." Mathematics and Computers in Simulation 77.5-6 (2008): 464-475.

[27] Matias, J. N., & Munger, K. (2019). The Upworthy Research Archive: A Time Series of 32,488 Experiments in US Advocacy.

[28] Matias, J.N. Aubin Le Quere, M. (2020) Asking Questions of the Upworthy Research Archive, a slide deck from Matias's field experiments class. This deck includes advice on meta-analyzing the archive.

[29] Brauers, W. K. M.; Zavadskas, E. K. Project management by MULTIMOORA as an instrument for transition economies. Technological and Economic Development of Economy, 16(1), 5-24, 2010.

[30] Opricovic, S.; Tzeng, G.H. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. European Journal of Operational Research, 156(2), 445-455, 2004.

[31] Hwang, C.L.; Yoon, K. Multiple Attribute Decision Making. In: Lecture Notes in Economics and Mathematical Systems 186. Springer-Verlag, Berlin, 1981.

[32] Garcia Cascales, M.S.; Lamata, M.T. On rank reversal and TOPSIS method. Mathematical and Computer Modelling, 56(5-6), 123-132, 2012.

[33] Zavadskas, E. K.; Turskis, Z.; Antucheviciene, J.; Zakarevicius, A. Optimization of Weighted Aggregated Sum Product Assessment. Electronics and Electrical Engineering, 122(6), 3-6, 2012.

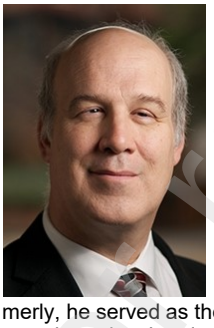
[34] Odu, G. O. "Weighting methods for multi-criteria decision making technique." Journal of Applied Sciences and Environmental Management 23.8 (2019): 1449-1457.

[35] Tang, Diane, et al. Overlapping Experiment Infrastructure:

More, Better, Faster Experimentation. Proceedings 16th Conference on Knowledge Discovery and Data Mining. 2010.
[36] Kohavi, Ron, Randal M. Henne, and Dan Sommerfield. "Practical guide to controlled experiments on the web: listen to your customers not to the hippo." Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007.
[37] Feitelson, Dror G., Eitan Frachtenberg, and Kent L. Beck. "Development and deployment at facebook." IEEE Internet Computing 17.4 (2013): 8-17.
[38] Gomez-Uribe, Carlos A., and Neil Hunt. "The netflix recommender system: Algorithms, business value, and innovation." ACM Transactions on Management Information Systems (TMIS) 6.4 (2015): 1-19.
[39] Kohavi, Ron, et al. "Online controlled experiments at large scale." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.
[40] Machmouchi, Widad, and Georg Buscher. "Principles for the design of online A/B metrics." Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016.
[41] Fagerholm, Fabian, et al. "The RIGHT model for continuous experimentation." Journal of Systems and Software 123 (2017): 292-305.
[42] Auer, Florian, et al. "Controlled experimentation in continuous experimentation: Knowledge and challenges." Information and Software Technology 134 (2021): 106551.
[43] Gharakheili, Masoud Asghari, Mahmud Fotuhi-Firuzabad, and Payman Dehghanian. "A new multiattribute decision making support tool for identifying critical components in power transmission systems." IEEE Systems Journal 12.1 (2015): 316-327.
[44] Esmaeilzadeh, Ehsan, Michael Grenn, and Blake Roberts. "An SoS Framework for Improved Collaborative Decision Making." IEEE Systems Journal 13.4 (2019): 4122-4133.
[45] Sampath, Sreedevi, et al. "Applying concept analysis to user-session-based testing of web applications." IEEE Transactions on Software Engineering 33.10 (2007): 643-658.
[46] Schermann, Gerald, et al. "Bifrost: Supporting continuous deployment with automated enactment of multi-phase live testing strategies." Proceedings of the 17th International Middleware Conference. 2016.
[47] Fabijan, A., Dmitriev, P., Olsson, H. H., & Bosch, J. (2018, August). Effective online controlled experiment analysis at large scale. In 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 64-67). IEEE.
[48] R. Kohavi and S. Thomke, "The Surprising Power of Online Experiments," Harvard Business Review, no. October, 2017.
[49] A. Fabijan, P. Dmitriev, H. H. Olsson, and J. Bosch, "The Benefits of Controlled Experimentation at Scale," in Proceedings of the 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2017, pp. 18–26.
[50] Kohavi, Ron, Diane Tang, and Ya Xu. Trustworthy online controlled experiments: A practical guide to a/b testing. Cambridge University Press, 2020.
[51] Koning, Rembrand, Sharique Hasan, and Aaron Chatterji. Experimentation and startup performance: Evidence from A/B testing. No. w26278. National Bureau of Economic Research, 2019.
[52] Somik Raha: Decision Analytic A/B testing for Product Leaders; found online: https://towardsdatascience.com/decision-analytic-a-b-testing-for-product-leaders-417b3a33178f; Last accessed August 25, 2021
[53] Simba Dube: How to Analyze A/B Test Results and Statistical Significance in A/B Testing; found online: https://www.invespcro.com/blog/how-to-analyze-a-b-test-results/; Last accessed August 25, 2021
[54] Fabijan, Aleksander, et al. "Three key checklists and remedies for trustworthy analysis of online controlled experiments at scale." 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE, 2019.
[55] Deng, A., Zhang, P., Chen, S., Kim, D.W. and Lu, J., 2016. Concise summarization of heterogeneous treatment effect using total variation regularized regression. arXiv preprint arXiv:1610.03917.
[56] Fabijan, Aleksander, et al. "The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale." 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). IEEE, 2017.
[57] MCDM Package. Available online: https://cran.r-project.org/web/packages/MCDM/MCDM.pdf; Last accessed on June 13, 2019.
[58] Chang, Yu-Hern, and Chung-Hsing Yeh. "Evaluating airline competitiveness using multiattribute decision making." Omega 29.5 (2001): 405-415.
[59] Miller, David William. "Executive decisions and operations research." (1963).
[60] Yoon, Kwangsun. Systems selection by multiple attribute decision making. Kansas State University, 1980.
[61] Triantaphyllou, E., Shu, B., Sanchez, S. N., & Ray, T. (1998). Multi-criteria decision making: an operations research approach. Encyclopedia of electrical and electronics engineering, 15(1998), 175-186.
[62] Lotfi, F. Hosseinzadeh, R. Fallahnejad, and N. Navidi. "Ranking efficient units in DEA by using TOPSIS method." Applied Mathematical Sciences 5.17 (2011): 805-815.
[63] Triantaphyllou, Evangelos. "Multi-criteria decision making methods." Multi-criteria decision making methods: A comparative study. Springer, Boston, MA, 2000. 5-21.
[64] Tzeng, Gwo-Hshiung, and Jih-Jeng Huang. Multiple attribute decision making: methods and applications. CRC press, 2011.
[65] Ishizaka, Alessio, and Philippe Nemery. Multi-criteria decision analysis: methods and software. John Wiley & Sons, 2013.
[66] Jahanshahloo, Gholam Reza, F. Hosseinzadeh Lotfi, and Mohammad Izadikhah. "An algorithmic method to extend TOPSIS for decision-making problems with interval data." Applied mathematics and computation 175.2 (2006): 1375-1384.
[67] Zardari, Noorul Hassan, et al. Weighting methods and their effects on multi-criteria decision making model outcomes in water resources management. Springer, 2015.
[68] Zhu, Yuxin, Dazuo Tian, and Feng Yan. "Effectiveness of entropy weight method in decision-making." Mathematical Problems in Engineering 2020 (2020).
[69] Patel, Meera Rameshkumar, Manisha Pranav Vashi, and Bhasker Vijaykumar Bhatt. "SMART-Multi-criteria decision-making technique for use in planning activities." Proceedings of New Horizons in Civil Engineering (NHCE-2017), Surat India (2017).
[70] Roszkowska, Ewa. "Rank ordering criteria weighting methods–a comparative overview." (2013).
[71] Lwakatare, Lucy Ellen, Teemu Karvonen, Tanja Sauvola, Pasi Kuvaja, Helena Holmström Olsson, Jan Bosch, and Markku Oivo. "Towards DevOps in the embedded systems domain: Why is it so hard?." In 2016 49th hawaii international conference on system sciences (hicss), pp. 5437-5446. IEEE, 2016.
[72] Liu, Yuchu, David Issa Mattos, Jan Bosch, Helena Holmström Olsson, and Jonn Lantz. "Size matters? Or not: A/B testing with limited sample in automotive embedded software." In 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 300-307. IEEE, 2021.

[73] Bernroider, Edward WN, and Patrick Schmöllerl. "A technological, organisational, and environmental analysis of decision making methodologies and satisfaction in the context of IT induced business transformations." European Journal of Operational Research 224, no. 1 (2013): 141-153.
[74] Issa Mattos, David, Pavel Dmitriev, Aleksander Fabijan, Jan Bosch, and Helena Holmström Olsson. "An activity and metric model for online controlled experiments." In International Conference on Product-Focused Software Process Improvement, pp. 182-198. Springer, Cham, 2018.
[75] Zavadskas, Edmundas Kazimieras, Zenonas Turskis, and Simona Kildienė. "State of art surveys of overviews on MCDM/MADM methods." Technological and economic development of economy 20, no. 1 (2014): 165-179.
[76] Ishizaka, Alessio, and Sajid Siraj. "Are multi-criteria decision-making tools useful? An experimental comparative study of three methods." European Journal of Operational Research 264, no. 2 (2018): 462-471.
[77] Asadabadi, Mehdi Rajabi, Elizabeth Chang, and Morteza Saberi. "Are MCDM methods useful? A critical review of analytic hierarchy process (AHP) and analytic network process (ANP)." Cogent Engineering 6, no. 1 (2019): 1623153.
[78] Matias, J. Nathan, Kevin Munger, Marianne Aubin Le Quere, and Charles Ebersole. "The Upworthy Research Archive, a time series of 32,487 experiments in US media." Scientific Data 8, no. 1 (2021): 1-6.
[79] Menon, Aditya K., Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. "Multilabel reductions: what is my loss optimising?." Advances in Neural Information Processing Systems 32 (2019).
[80] Lamprakos, Christos P., Charalampos Marantos, Miltiadis Siavvas, Lazaros Papadopoulos, Angeliki-Agathi Tsintzira, Apostolos Ampatzoglou, Alexander Chatzigeorgiou, Dionysios Kehagias, and Dimitrios Soudris. "Translating quality-driven code change selection to an instance of multiple-criteria decision making." Information and Software Technology 145 (2022): 106851.
[81] Jadhav, Anil S., and Rajendra M. Sonar. "Evaluating and selecting software packages: A review." Information and software technology 51, no. 3 (2009): 555-563.
[82] Chiam, Yin Kia, Mark Staples, Xin Ye, and Liming Zhu. "Applying a selection method to choose Quality Attribute Techniques." Information and Software Technology 55, no. 8 (2013): 1419-1436.
[83] Farshidi, Siamak, Slinger Jansen, and Mahdi Deldar. "A decision model for programming language ecosystem selection: Seven industry case studies." Information and Software Technology 139 (2021): 106640.
[84] Ghapanchi, Amir Hossein, Ahmad Reza Ghapanchi, Amir Talaei-Khoei, and Babak Abedin. "A systematic review on information technology personnel's turnover." Lecture Notes on Software Engineering 1, no. 1 (2013): 98-101.
[85] Kochovski, Petar, Pavel D. Drobintsev, and Vlado Stankovski. "Formal quality of service assurances, ranking and verification of cloud deployment options with a probabilistic model checking method." Information and Software Technology 109 (2019): 14-25.
[86] Kohavi, Ronny, Thomas Crook, Roger Longbotham, Brian Frasca, Randy Henne, Juan Lavista Ferres, and Tamir Melamed. "Online experimentation at Microsoft." Data Mining Case Studies 11, no. 2009 (2009): 39.

Jie JW Wu received the B.S. and M.S. degree in Computer Science from Shanghai Jiao Tong University, China, in 2012 and 2015, respectively. Since 2015, he has been in the software industry in the U.S., and he is currently a software engineer at Snap Inc. He is currently pursuing the Ph.D. degree in system engineering with George Washington University, Washington, DC, USA. He is mainly interested in automated techniques and system thinking in software engineering.



Thomas A. Mazzuchi received the B.A. degree in mathematics from Gettysburg College, Gettysburg, PA, USA, in 1978, and the M.S. and D.Sc. degrees in operations research, from The George Washington University (GW), Washington, D.C., USA, in 1979 and 1982, respectively.

He is a Professor of engineering management and systems engineering, and Chair of the Department of Engineering Management and Systems Engineering, in the School of Engineering and Applied Science, GW. Formerly, he served as the Chair of the Department of Operations Research, and as Interim Dean of the School of Engineering and Applied Science. He has been engaged in consulting and research in the areas of reliability and risk analysis, and systems engineering techniques, for over 30 years. He served for 2.5 years as a Research Mathematician at the International Operations and Process Research Laboratory of the Royal Dutch Shell Company. While at Shell, he was engaged in reliability and risk analysis of large processing systems, maintenance optimization of off-shore platforms, and quality control procedures at large-scale chemical plants. In his academic career, he has held research contracts in development of testing procedures for both the U.S. Air Force and the U.S. Army; in spares provisioning modeling with the U.S. Postal Service; in mission assurance with NASA; and in maritime safety and risk assessment with the Port Authority of New Orleans, the Washington Office of Marine Safety, the Washington State Department of Transportation, and the San Francisco Bay Area Transit Authority.

Dr. Mazzuchi is an Elected Member of the International Statistics Institute.

Shahram Sarkani received the B.S. and M.S. degrees in civil engineering from Louisiana State University, Baton Rouge, LA, USA, in 1980 and 1981, respectively, and the Ph.D. degree in civil engineering from Rice University, Houston, TX, USA, in 1987.

He is a Professor of engineering management and systems engineering with The George Washington University (GW), Washington, D.C., USA. His current administrative appointments are Inaugural Director, School of Engineering and Applied Science off-campus and Professional Programs since 2016, the school unit to establish cross-disciplinary and departmental programs for offer off-campus and/or by synchronous distance learning; and Faculty Adviser and Academic Director, EMSE off-campus Programs since 2001, the department unit that designs and administers five separate graduate degree programs in six areas of study that enroll over 800 students across the USA and abroad. He joined GW in 1986, where previous administrative appointments include Chair of the Civil, Mechanical, and Environmental Engineering Department (1994–1997); and Interim Associate Dean for Research, School of Engineering and Applied Science (1997–2001). In over 500 technical publications and presentations, his research in systems engineering, systems analysis, and applied enterprise systems engineering has application to risk analysis, structural safety, and reliability. He has conducted sponsored research for such organizations as NASA,

NIST, NSF, U.S. AID, and the U.S. Departments of Interior, Navy, and Transportation.

Dr. Sarkani received the Walter L. Huber Civil Engineering Research Prize by the American Society of Civil Engineers in 1999. He was inducted into the Civil and Environmental Engineering Hall of Distinction, Louisiana State University, in 2010. He is a Registered Professional Engineer in Virginia.