Received: 26 July 2025 | Revised: 27 August 2025 | Accepted: 2 September 2025

DOI: 10.1002/aaai.70032

SPECIAL TOPIC ARTICLE

Open-source AI at scale: Establishing an enterprise AI strategy through modular frameworks

Serdar Kadıoğlu[1,2]

[1]AI Center of Excellence, Fidelity Investments, Boston, Massachusetts, USA

[2]Department of Computer Science, Brown University, Providence, Rhode Island, USA

Correspondence
Serdar Kadıoğlu
Email: serdark@cs.brown.edu

Abstract

We present a comprehensive enterprise AI strategy developed within the AI Center of Excellence at Fidelity Investments, emphasizing the strategic integration of open-source AI frameworks into scalable, modular, and reproducible enterprise-grade solutions. Our approach is structured around five key pillars: learning from offline data, learning from online feedback, intelligent decision-making, automated assistants, and responsible AI practices. Through a suite of 12 open-source libraries, we demonstrate how modular and interoperable tools can collectively enhance scalability, fairness, and explainability in real-world AI deployments. We further illustrate the impact of this strategy through three enterprise case studies. Finally, we distill a set of best deployment practices to guide organizations in implementing modular, open-source AI strategies at scale.

INTRODUCTION

The adoption of AI at the enterprise scale faces several challenges, including scalability, interoperability, explainability, and operational complexity. These challenges are amplified in highly regulated sectors such as financial services, where trust, transparency, and resilience are non-negotiable. This raises the need for common and core capabilities that are built as *reusable* and *modular* components to address business demand across diverse applications. To enable this, open-source software plays a crucial role in accelerating innovation and deployment, providing a key advantage in the ongoing "buy versus build" debate in enterprise settings.

According to the 2025 AI Index Report, there are over 5 million open-source AI projects on GitHub, with a remarkable increase of over 40% in the past year (Maslej et al. 2025). A study conducted at Harvard Business School estimates the global value of open-source software to be $8.8 trillion (Hoffmann, Nagle, and Zhou 2024). Beyond software, research from IDC indicates that open-source models constitute more than half of the currently deployed enterprise use cases (Rosen 2025). The scale of open-source AI is undeniable, as also recognized in the AI Action Plan (The White House 2025).

To address the challenges of AI adoption, many organizations have established AI Centers of Excellence, which are specialized units that centralize expertise, governance, and best practices to accelerate AI adoption across the enterprise. An AI Center serves as both a strategic hub and an execution engine, ensuring that AI solutions are not isolated experiments but integrated, scalable capabilities aligned with business objectives. Similarly, the AI Center of Excellence at Fidelity serves as a centralized hub for advancing AI adoption across the financial services enterprise. Unlike academic research labs, the AI Center

TABLE 1 Strategic pillars of the Enterprise AI Strategy, their key requirements, and the supporting open-source software components.

Strategic pillars	Key requirements	Open-source software
Offline learning	Robust, scalable, reproducible	Selective, TextWiser, Seq2Pat
Online learning	Real-time, A/B testing, adaptive, efficient	Mab2Rec, MABWiser
Decision making	Large-scale, transparent, integrated	PathFinder, Balans
Automated assistants	Accurate extraction and translation	Ner4Opt, Text2Zinc, iCBS
Responsible AI	Explainability, fairness, bias mitigation	Jurity, BoolXAI

operates at the intersection of R&D and production, balancing innovation with compliance, security, and operational scalability. Our mandate is to deliver AI responsibly and at scale, bridging research and production through modular, reusable, and open-source-driven frameworks. For a historical perspective, Iansiti & Lakhani (2020) explores Fidelity's AI journey in their chapter on becoming an AI-driven company. According to Fidelity's 2024 Annual Report, the firm serves over 40 million individual investors and manages over $15 trillion in assets. In 2024 alone, we processed an average of 3.5 million trades per day, handled 37.5 million calls annually, supported 40 million unique customers in digital interactions, facilitated 5.5 million retail customer appointments, and conducted over 2.2 million customer service engagements through social media platforms (Fidelity Investments 2024). Fidelity operates across a wide range of financial domains, including wealth management, retirement services, brokerage, and institutional investing.[1] This operational breadth of Fidelity necessitates an AI strategy capable of supporting heterogeneous workloads while maintaining strict regulatory compliance and ethical standards.[2]

To meet this challenge, we have developed a comprehensive suite of open-source AI libraries that cover a wide range of focus areas across *learning* and *reasoning* systems, and their hybridization. These focus areas are driven by business-critical problems and include online learning, large language and vision models, decision-making under uncertainty, and responsible AI. Within each focus area, we have open-sourced modular frameworks that, when combined, enable innovative applications and support strategic, multi-sector partnerships with academia and industry. Collectively, our software components have been *downloaded more than two million times* in the broader AI community beyond Fidelity.

In this paper, we present an overview of how the integration of open-source libraries has shaped an overarching Enterprise AI Strategy that emphasizes scalability, modularity, reproducibility, fairness, and explainability to enhance applied AI research and industrial deployment. Enterprise AI strategies encompass not only software and models but also the organizational structures and human expertise required to sustain them. To be effective, these strategies must align specific business requirements with dedicated software components that are actively maintained by teams possessing deep knowledge of both the technology and its application domain. By defining abstract workloads in terms of modular software artifacts, organizations can foster scalable adoption, facilitate cross-functional collaboration, and enable continuous improvement of their AI capabilities.

Table 1 presents an overview of the strategic pillars of our enterprise AI strategy, their key requirements, and the supporting open-source software components. In the following, we begin with our overarching vision and goals, and then, through a collection of 12 open-source libraries, present our modular approach to open-source AI, categorized as:

1. **AI for learning from offline data**: robust and scalable feature extraction from structured, unstructured, and semi-structured datasets.
2. **AI for learning from online feedback**: adaptive systems that continuously improve from user interaction.
3. **AI for decision making**: integrated (meta) solvers for large-scale resource management.
4. **AI for automated assistants**: accurate translation of natural language problem descriptions into formal models for non-experts.
5. **Responsible AI**: explainability, fairness assessment in the absence of protected attributes and ground truth labels, and reproducibility.

ENTERPRISE AI STRATEGY: VISION AND GOALS

For applied AI, it is essential to integrate open-source frameworks into enterprise-grade solutions. Open-source components accelerate innovation and experimentation while lowering costs, and their modularity ensures that enterprise teams can scale and maintain solutions across diverse workloads. Without this integration, applied AI often results in isolated prototypes that cannot be reliably

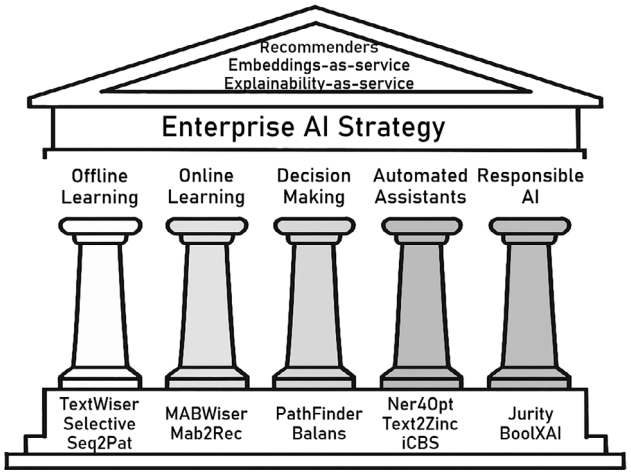

FIGURE 1 Enterprise AI strategy: Five strategic pillars (middle) mapped to modular components (bottom) and their integration into enterprise applications (top).

deployed, audited, or sustained in production. These components enable teams to experiment quickly while maintaining the reliability and compliance standards required for production-scale systems.

The core principles of Fidelity's AI strategy rely on *modularity* and *reusability* across diverse applications while enhancing adoption through easy-to-use software, robustness, explainability, and fairness in systems, as well as reproducibility, deployment readiness, and ease of maintenance.

Why modular and reusable open-source? Enterprise-scale AI is only as strong as its weakest pillar: without robust offline learning, personalization is shallow; without online adaptation, systems cannot track evolving preferences; without optimization, insights do not translate into prescriptive actions; without automated assistants, adoption is slow; and without responsible AI, trust and compliance are jeopardized. By structuring capabilities into modular components, we make them easier to track, evaluate, and improve independently of each other, yet collaboratively across teams.

In this paper, we provide an overview of our AI ecosystem within each category while illustrating how tools complement and integrate with one another to create unifying themes. Our goal is to show how these open-source tools collectively address enterprise-scale AI challenges by examining three case studies: a recommendation platform, a text embedding service, and an explainability service, as illustrated in Figure 1.

Best Deployment Practices: Drawing from our real-world experience, we distill key lessons for implementing modular, open-source AI strategies at enterprise scale. These best practices (BP), referenced throughout the paper and summarized later in Table 2, cover design principles, interoperability, and cost management to ensure scalable and responsible AI deployment. The design of our modular architecture is anchored in BP1 on *modularity from the start*.

Before introducing our strategic pillars, we begin with a concrete, enterprise case study on recommender systems, powered by the integration of multiple components from Figure 1.

Case study: Recommender systems

The applications of personalization and recommender systems are widespread in the industry; see, for example, (Amatriain and Basilico 2016) for a comprehensive overview. Despite being so pervasive, building recommender *systems* still requires complex machinery of data preparation, transformation, modeling, and production deployment. Each of these components requires significant domain expertise and serious engineering effort to deploy a robust system that can operate at scale.

Most prominent players of the technology industry contributed dedicated software such as Torch Recommenders from Meta (Naumov et al. 2019), TensorFlow Recommenders from Google (Pasumarthi et al. 2019), Recommenders from Microsoft (Argyriou, González-Fierro, and Zhang 2020), and Merlin Recommenders from NVIDIA (Oldridge et al. 2020). Despite these efforts, two issues remain for the enterprise: one challenge for technical users of these systems and another challenge for companies that want to deploy them.

First, these are monolithic frameworks specialized in one and only one task, that is, recommenders. The effective use of these systems still requires expert skills. This poses a significant learning curve for generalist data scientists. Even with the best effort, the skill is not immediately transferable to other tasks, such as propensity modeling, natural language processing, text featurization, feature selection, and pattern mining, among others. Such in-depth knowledge might be desirable for recommendation experts, but it also poses a challenge for the broader talent that must deploy personalization applications.

Second, these frameworks necessitate significant engineering capabilities to integrate the training and testing pipelines to ensure real-time performance at scale. Additionally, they have high hardware demands due to the data-intensive nature of neural networks, which require millions of user and item interactions. This engineering demand poses a challenge for industry players that are not primarily software-focused, in contrast to the high-tech companies that developed these systems specifically for their own needs.

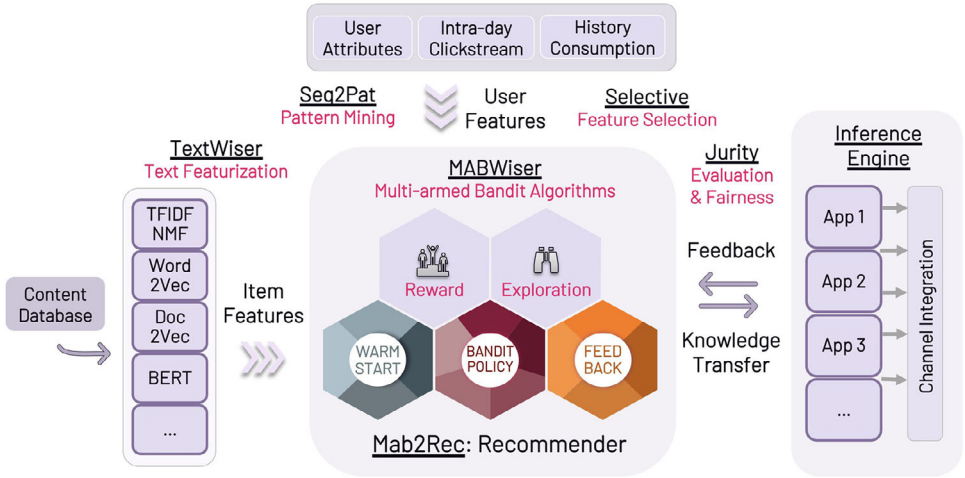

FIGURE 2 Modular components integrated into the MAB2REC for building recommender systems.

Personalization is not a privilege reserved for technology companies; most other businesses also have opportunities to personalize touchpoints for their end-users. In fact, from the end-user's perspective, today, we would like any interaction to be personalized seamlessly for our needs, regardless of the provider. To tackle these two real-world challenges, we deployed a *modular framework*, named MAB2REC, for constructing recommender systems from higher-order abstractions[3]. We conceptualize recommender systems as having plug-and-play components that can operate effectively independently.

As shown in Figure 2, the system consists of several integrated components. The MAB2REC framework unifies several of the industry-strength components from Table 1. These include: content features obtained through natural language processing using TEXTWISER; user features developed via sequential pattern mining with SEQ2PAT and SELECTIVE; learning from user interactions using multi-armed bandit algorithms through MABWISER; and finally, performance and fairness evaluations conducted with JURITY. More details can be found in All Things Open blog.[4]

When these higher-order components are integrated via MAB2REC, we create a robust toolchain that facilitates the development and deployment of industrial recommender systems. At the same time, each component functions independently, is open-source, and can be reused outside of recommendation applications.

A practical application of this framework can be found in (Verma et al. 2023), which details the use of MAB2REC for training and deploying article recommendations for 463 subscribers who opted for 81 articles with 37,423 user–item interactions. Contextual multi-armed bandits are effective in managing the exploration-exploitation trade-off in a rapidly changing article recommendation environment.

Thanks to this enterprise recommendation platform, we identified a subtle, non-deterministic behavior in the well-known LinTS bandit policy, which had remained unnoticed in the academic community for over 10 years. We provided a simple solution to ensure reproducibility and formally proved its correctness in (Kilitçioğlu and Kadıoğlu 2022).

The MAB2REC framework has been featured in our industry collaboration with NVIDIA[5], highlighted in Comet ML[6], and referenced in ACM RecSys[7]. Beyond Fidelity, our recommendation framework has found applications across diverse domains, including:

1. Gelom for designing automated machine learning pipelines (Nikitin et al. 2021).
2. HawkAI for an automated trading recommendation system (Tosa 2025).
3. TopTune (Wei et al. 2025) and UniTune (Zhang et al. 2023) for recommendations in tuning the performance of database management systems.
4. Stream for context-aware recommendations in video streaming (Zhao and Pan 2023).
5. AEM for A/B testing in Adobe experience manager (Adobe 2025)
6. ALNS for adaptive large neighborhood search for solving optimization problems (Wouda and Lan 2023).

AI FOR LEARNING FROM OFFLINE DATA

In the context of our enterprise AI strategy, learning from offline data supports workloads that require extensive historical context before deployment, as utilized heavily in our case study for recommenders. In general, *contextual information* is a key component of modern machine learning systems. However, not all information available

offline is immediately relevant or usable as context in a model. Without robust text embeddings, feature selection, and sequential pattern mining tools, these workloads become brittle, expensive to maintain, and inconsistent across teams.

As shown in Figure 1, our approach addresses this through three modular, open-source libraries; TEXTWISER, SELECTIVE, and SEQ2PAT, each specializing in a different data modality from unstructured text data, structured tabular data, and semi-structured sequential data.

Unstructured Text Data: We next turn to unstructured text data to complement structured data. Various text embeddings have become prevalent for consuming raw text. However, there is no silver bullet for determining which featurization technique or combination will provide the best performance for downstream applications. The choice of embeddings ranges widely. When faced with several options, benchmarking becomes essential to identify the most effective technique. The task becomes more challenging when we consider combinations of featurization methods or their transformation using dimensionality reduction or decomposition techniques.

Thankfully, a rich set of tools exists that we can utilize, including Spacy (Honnibal and Montani 2017), Flair (Akbik, Blythe, and Vollgraf 2018), and HuggingFace Transformers (Wolf et al. 2019). Still, the availability of various techniques in standalone, isolated libraries does not provide the needed uniformity. This is especially problematic in the industry, as it requires experimenting with many different tools, building a custom layer on top to find the best approach, and redoing the work when the scenario changes in a new use case. Even worse, this is an effort that *every* data scientist in the organization has to repeat *for each* application.

To address this challenge, we introduced a novel context-free grammar of embeddings (Kilitçioğlu and Kadıoğlu 2021). This grammar allows us to systematically represent the language of all valid featurization techniques. The main idea is to treat each instantiation of embedding and transformation combinations as a sentence subject to language membership against the grammar of embeddings. At its core, the grammar utilizes two production rules, concatenate and transform, that define arbitrarily complex yet valid text featurization pipelines.

TEXTWISER implements this grammar for rapid experimentation with more than 25 text embeddings using over 100 pre-trained models. It serves as a building block in applications, including an embedding service that we review next.[8]

Case study: Embeddings-as-a-Service

We now share an enterprise application powered by TEXTWISER. The motivation behind this service is as follows: even with a unified text featurization toolchain, similar efforts are *re-invented* in machine learning projects across the organization. A considerable amount of data scientists' time and shared computing resources (especially GPUs) are spent utilizing text data, often yielding different results. More concretely, according to the NVIDIA benchmark, a V100 GPU can process inference on a BERT base model at a rate of 766 sequences per second (NVIDIA 2020). Sequences with over 100M tokens are considered typical in text datasets, which amounts to 36 h of GPU time spent generating embeddings for a single featurization task (NVIDIA 2020). This effort is compounded across various business units in the organization over time. Moreover, the proliferation of pre-processing pipelines, embedding models, and pooling methods results in different representations across various use cases, with no quantifiable benefits. This complexity increases further when fine-tuning and data-specific models are introduced.

Our Embeddings-as-a-Service abstracts these steps away from data scientists by hosting pre-computed and readily available embeddings for various datasets and language models. It serves dense feature vectors built via TEXTWISER to users on demand with a REST API and removes repetitive steps from the AI modeling pipeline. This deployment began with a focused use case, aligning with BP2's principle of starting small and service optimization demonstrates BP3 on *cost management*. Other benefits include:

- Reduced entry barrier to consuming text data.
- Ease of sharing language models across teams and business units.
- Transparency and archivability of data processing steps.
- Immediate baseline performance from readily available embeddings.

An additional benefit is increased security and privacy. When embeddings are already available for consumption, fewer access requests are needed for computing resources and the raw text of content stored across multiple systems. Similar efforts exist in the public domain, such as BERT-as-a-Service (Xiao 2018), which, unlike TEXTWISER, is specific to only one embedding.

Structured Tabular Data: In industry, it is common to have large amounts of unprocessed tabular data. This is especially true for user context, as most companies possess a wide range of user attributes. However,

determining the most relevant features for a given application remains a non-trivial task. For this purpose, we open-source SELECTIVE[9]. This white-box feature selection library supports both unsupervised and supervised selection methods, as well as automatic task detection for classification and regression tasks. Users do not need to determine which selection method is appropriate manually.

SELECTIVE (Kadıoğlu, Kleynhans, and Wang 2021) offers a range of filtering and embedded selection methods, varying in complexity from simple variance-, statistics-, correlation-, and divergence-based methods to embedded penalized linear regression models and non-linear tree-based methods. Users can exploit text annotations associated with each feature using a novel multi-objective optimization formulation (Kleynhans, Wang, and Kadıoğlu 2021a), similar in spirit to recent foundational tabular models, such as TabICL (Qu et al. 2025) and TabPFN (Hollmann et al. 2025). The library allows benchmarking multiple selection methods using cross-validation for robustness and offers built-in parallelization for scalability. It is effective when integrated into active learning for recommenders (Kadıoğlu, Kleynhans, and Wang 2024).

For predictive modeling, SELECTIVE helps determine the subset of user features that are most relevant for a given outcome. Note that this is not a one-off exercise independent of the modeling approach. Instead, it needs to be regularly re-evaluated as more interaction data and new features become available. By offering a standardized and efficient toolkit for feature selection, we enable practitioners to identify robust candidate features consistently with minimal effort. The use of standardized APIs and interfaces supports BP4 on *prioritizing interoperability*.

Semi-Structured Sequential Data: The other data type we consider is semi-structured sequential data, particularly sequential clickstream, to better understand users' digital behavior. Clickstream has unique properties, such as its real-time and streaming nature. On the one hand, it provides *unstructured text* such as web pages. On the other hand, it yields sequential information where visits can be viewed as *structured events* representing user journeys. Each sequence is an ordered set of *items* associated with a set of *attributes* that capture item properties, for example, price and timestamp.

As an alternative to neural architectures, we consider an approach based on sequential pattern mining (SPM). A *pattern* is a subsequence that occurs in at least one sequence, maintaining the original item ordering. The idea of SPM is to search for patterns with high *frequency* of occurrence. However, finding the entirety of frequent patterns is highly costly as the set is typically too large and may not provide significant insights. It is thus important to search for patterns that are not only frequent but also capture specific properties. This motivated Constraint-based SPM (CSPM) (Pei, Han, and Wang 2007) to incorporate constraint reasoning into data mining to find smaller subsets of interesting patterns.

In (Kadıoğlu et al. 2023), we applied Constraint-based SPM (CSPM) using declarative constraint models based on multi-valued decision diagrams (Hosseininasab, van Hoeve, and Ciré 2019). The method is embodied in SEQ2PAT[10] to support pattern mining applications, developed through an industry–academia collaboration with CMUs (Wang et al. 2022).[11]

In the enterprise setting, SEQ2PAT enables knowledge discovery in large sequence databases, subject to desired properties, for example, identifying frequent patterns from web sessions where users spend at least a minimum amount of time on particular items within a specific price range. SEQ2PAT also serves as an integration technology for automated feature generation from sequential data. To that end, we designed an algorithm for embedding CSPM in Dichotomic Pattern Mining (DPM) (Wang and Kadıoğlu 2022) that leverages the dichotomy between positive and negative outcomes in user cohorts. With DPM, we identify frequent patterns that uniquely distinguish between positive and negative outcomes, for example, buy versus no buy. Our experiments on real-world e-commerce shopping intent prediction and intrusion detection show that models built on top of our SEQ2PAT patterns are competitive with the state-of-the-art LSTMs (Requena et al. 2020), and best results are achieved when frequent and deep patterns are combined (Ghosh et al. 2022).

AI FOR LEARNING FROM ONLINE FEEDBACK

Many customer-facing services require systems that can adapt in real-time as user preferences evolve. This is the role of learning from online feedback in our enterprise strategy, where models continuously update based on interaction data. Without online learning capabilities, personalization systems risk becoming static, failing to adjust to changing markets or user behaviors. The primary tool in our strategy from Figure 1 for this workload is MABWISER, which implements multi-armed bandit algorithms with reproducibility guarantees even when run in parallel similar to (Cire, Kadıoğlu, and Sellmann 2014).

Multi-Armed Bandits: In the online setting for dealing with sequential decision-making problems, it is well-known that focusing purely on exploitation fails to capture the dynamic nature of evolving preferences and a principled approach is needed to explore different policies for continuous learning. To balance this

exploration-exploitation trade-off, Multi-Armed Bandits (MAB) is a well-known family of algorithms that focus on sequential decision-making. MAB algorithms define each "arm" as a decision that an agent can make, generating either a deterministic or stochastic "reward." At each time step, the agent faces a decision on whether to utilize an arm that has a high expected reward ("exploit") or to try out new arms to learn something new ("explore"). The agent's goal is to maximize the cumulative reward, which requires balancing exploration with exploitation. Contextual MAB utilize a state ("context") that captures side information that might affect the reward for a given arm. Formally, the reward for an arm becomes a function of the selected arm and the state (Strong, Kleynhans, and Kadıoğlu 2019).

We contribute MABWISER[12] to open-source which offers contextual, context-free, non-parametric, and parametric MAB algorithms (Strong, Kleynhans, and Kadıoğlu 2021) as deployed in our recommender framework.

The available bandit policies accommodate both batch and online learning. The built-in parallelization strategy allows speedups in both training and testing for scalability. Moreover, we demonstrate how to ensure *reproducibility* between parallel runs, a highly desirable property in industrial settings, regardless of the number of cores available or the order in which tasks are allocated to different jobs. For hyperparameter tuning and rapid experimentation, it includes a simulation capability. In our enterprise recommender platform MAB2REC, MABWISER serves as the decision engine balancing exploration and exploitation. We also integrate exploration strategies into Bayesian deep learning (Wang and Kadıoğlu 2019, 2023). Beyond recommenders, we apply online feedback within a read-write-learn framework using self-supervision to enhance handwriting recognition in document automation services (Boteanu, Cheng, and Kadıoğlu 2023) and reactive restart strategies (Kadıoğlu, Sellmann, and Wagner 2017).

AI FOR DECISION MAKING

Enterprise-scale decision-making workloads such as resource allocation, routing, and scheduling require more than predictive models. They demand optimization engines capable of producing high-quality, explainable solutions under uncertainty. Without optimization capabilities, operational AI systems may identify predictions on what is next but fail to prescribe how to act on those predictions.

As illustrated in Figure 1, we address this gap with PATHFINDER for stochastic scheduling and BALANS, a meta-solver integrating adaptive search, multi-armed bandits, and mixed-integer programming.

Assignment, Routing, and Scheduling: Our first application considers the service industry that deals with the provisioning of services to people, for example, home health care, banking services, and repair services. These problems require teams to travel to deliver services to geographically distributed users. Service providers often quote an appointment time (planned service start time) to each customer in advance to avoid delivery failure. Therefore, when providers plan for service, they need to solve an a problem that integrates:

- The number of service teams to hire (i.e., the sizing problem).
- The allocation of service teams to the customers (i.e., the assignment problem).
- The visit of service teams to customers (i.e., the vehicle routing problem).
- The assignment of appointment times for the customers (i.e., the scheduling problem).

For the enterprise setting, *stochasticity* is a crucial property of the *integrated* service assignment, vehicle routing, and appointment scheduling problem. Our main contribution is addressing this problem under three key sources of stochasticity: service duration, travel time, and customer cancellation, as detailed in (Samuel, Areyan Viqueira, and Kadıoğlu 2021). The corresponding PATHFINDER library[13] brings our methodology to practice as a finalist of the International Modelling and Optimization Competition.[14] Addressing stochasticity also forms the basis of our collaboration with the Rhode Island Public Transit Authority to provide reliable and cost-effective service across the community.[15] Finally, we addressed similar resource allocation problems for the Oracle Cloud, including core group allocation and provisioning (Kadıoğlu 2019; Kadıoğlu, Colena, and Sebbah 2016) and heterogeneous assignments for in-memory data grids (Kadıoğlu et al. 2015; Sebbah et al. 2016).

ML-Guided Solvers: Solution approaches for optimization problems rely on solvers such as SCIP or GUROBI Incorporating learning-based methods into solvers has shown great potential to improve performance (Bengio, Lodi, and Prouvost 2021; Kadıoğlu et al. 2011; Kadıoğlu, Malitsky, and Sellmann 2012; Kadıoğlu, Sellmann, and Wagner 2017; Liberto et al. 2016). Our next application considers how to improve these mixed-integer programming solvers via online learning in a solver–agnostic manner without relying on supervised labels.

For real-world deployments, the significant drawback of existing learning-guided methods is their heavy dependency on offline training, which is computationally costly, requires carefully curated training datasets with desired properties and distributions, and has limited

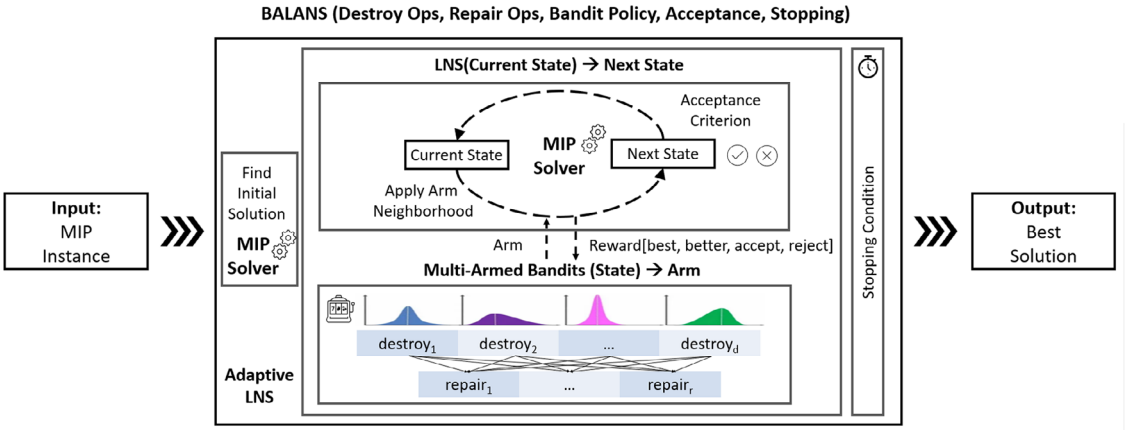

FIGURE 3 The modular architecture of the BALANS meta-solver with its configurable components integrating online learning, meta-heuristics, multi-armed bandits, and mixed-integer programming.

generalization capabilities. Moreover, training might even depend on using exact solvers in the first place to create the supervised datasets, which defeats the purpose of improving solving for hard instances. Adapting offline learning-based methods to new distributions and domains remains a challenge, and hence, online learning approaches are a much-needed alternative in practice.

In (Cai, Kadıoğlu, and Dilkina 2025a), we enhance solvers using online adaptive methods that do not require *any offline training*. We take the idea of machine learning-guided large-neighborhood search (LNS) a step further and propose a meta-solver, BALANS, based on Adaptive LNS operating on top of a solver composed of a diverse set of neighborhoods driven by a multi-armed bandit online learning policy.[16]

As shown in Figure 3, BALANS serves as an integration technology that combines ALNS library for adaptive large-neighborhood search (Wouda and Lan 2023), our MABWISER library for multi-armed bandits, and SCIP (Bolusani et al. 2024) and GUROBI (Gurobi 2024) as solvers. To summarize the methodology from Figure 3: once an initial solution is found, the main ALNS loop starts, as an interplay between LNS and MAB. MAB selects a neighborhoods and LNS applies the operation. Based on the solution quality, and acceptance criterion, LNS provides the reward feedback to update the estimate of the arm based on the learning policy. As in other meta-heuristics (Kadıoğlu and Sellmann 2009), this process repeats until the stopping criteria, and the best solution found is returned.

The result is a highly configurable, modular, and extensible integration technology at the intersection of adaptive search, meta-heuristics, multi-armed bandits, and mixed-integer programming. Future advances in these distinct fields, realized independently within each software, propagate to our meta-solver with compounding effects (Cai, Kadıoğlu, and Dilkina 2025b) and even parallelization (Yilmaz et al. 2025). To bridge predictive methods with prescriptive techniques further, we share our practical advice in (Kadıoğlu 2025).

AI FOR AUTOMATED ASSISTANTS

Even with powerful solvers and learning systems, many enterprise AI opportunities remain untapped because modeling expertise is scarce. As part of our enterprise AI strategy, automated assistants bridge this gap by turning natural language problem descriptions into formal models. Without automation in model creation, non-experts must rely on scarce optimization specialists, slowing down adoption. As highlighted in Figure 1, we address this with NER4OPT, a specialized named entity recognition system for optimization (Kadıoğlu et al. 2024), and TEXT2ZINC, a dataset and evaluation pipeline of co-pilots to generate MINIZINC models from natural language to alleviate users from the cognitive task of modeling (Singirikonda, Kadıoğlu, and Uppuluri 2025). Finally, with iCBS (Rosenberg et al. 2024), we prune large-language and vision models to support inference and interactivity.

Although optimization technology has achieved significant advancements from improvements in solver efficiency to the development of high-level modeling languages, the fundamental decision-making framework has remained unchanged for decades, adhering to the de facto *model-and-run strategy*. Within this status quo, users are required to manually convert problem descriptions into optimization models, which are subsequently processed by solvers to obtain solutions. Translating business problems into formal constraint models remains a challenge,

particularly for non-specialists. This gap often slows down adoption in enterprise settings.

Our contributions focus on modeling assistants that bridge the gap between understanding textual descriptions and translating them into precise mathematical formulations, thereby enhancing the accessibility of optimization technology. As outlined in our proposal for Holy Grail 2.0 (Tsouros et al. 2023), our ultimate vision is a paradigm shift that integrates automated modeling assistants capable of translating problem descriptions expressed in natural language into formal requirements, referred to as LLM Modeling Assistants, similar to LLM Code Co-Pilots.

Entity Extraction for Optimization: We first study named entity recognition to capture components of optimization models such as the objective, variables, and constraints from free-form natural language text, and coin this problem and its solution as NER4OPT.[17]

We demonstrate how to solve NER4OPT using classical NLP techniques that leverage morphological and grammatical properties, as well as modern methods that utilize pre-trained LLMs and fine-tune the transformer architecture with optimization-specific corpora. For best performance, we present their hybridization, combined with feature engineering and data augmentation, to leverage the language of optimization problems (Dakle et al. 2023). Our solution to NER4OPT is open-source and has already been utilized in experimental co-pilots, such as ChatOpt (Michailidis, Tsouros, and Guns 2024).

Modeling Co-Pilots: We next introduce TEXT2ZINC, a unified, cross-domain dataset that combines both satisfaction and optimization problems expressed in natural language[18]. This is the *first dataset* in this domain that enables modeling satisfaction *and* optimization problems. In addition, our modeling approach remains *agnostic to the underlying solving technology*, addressing a significant gap in existing work that only focuses on a single solver. In our approach, the generated models can be compiled into various backends, thanks to MINIZINC integration, including constraint programming solvers such as Gecode and Chuffed, SAT/LCG solvers like Google OR-Tools, and mathematical programming solvers such as SCIP and Gurobi.

The TEXT2ZINC dataset encompasses a diverse range of problem types across multiple domains, including applications in finance, healthcare, energy, scheduling, transportation, logistics, manufacturing, and production. We demonstrate how to unify various sources, problem types, and resources using a generic schema designed for LLMs. We further enhance problems through reformulation, metadata enrichment, and curation to ensure consistency and quality. This comprehensive collection provides a robust foundation for evaluating language-to-model translation.

Using TEXT2ZINC, we build several modeling co-pilots from zero-shot learning to multi-round chain-of-thought prompting, integration with knowledge graphs, and grammar-constrained generation (Singirikonda, Kadıoğlu, and Uppuluri 2025). An active Hugging Face Leaderboard accompanies this to enable the community with ongoing benchmarks in pursuit of automated assistants.[19]

Pruning Large-Language and Vision Models: Lastly, it is worth noting that the computational effort required to run language models remains significant, resulting in a high economic cost and increased power usage for training, storage, and inference. This can be prohibitive for automated modeling assistants, which motivated our work on pruning. Pruning neural networks, which involves removing some of their weights, maintains high performance while reducing model complexity.

Partnering with Amazon, in (Rosenberg et al. 2024), we propose a neural network pruning technique that solves an optimization problem over a subset of the weights in an iterative manner using block coordinate descent. We refer to this as iterative Combinatorial Brain Surgeon (iCBS)[20] and show that it applies successfully to large-language and vision models such as Mistral (Jiang et al. 2023) and DeiT (Touvron et al. 2021), as highlighted in Amazon blog.[21]

We show that iCBS achieves higher performance metrics at the same density compared to existing pruning methods such as Wanda (Sun et al. 2023), while optimizing over only a fraction of the weights. Moreover, our approach allows for a quality-time tradeoff that is not available when using a one-shot pruning technique alone. The block-wise formulation enables the use of hardware accelerators, potentially offsetting the increased computational costs compared to one-shot pruning methods. In particular, the optimization problem solved for each block is quantum-amenable in that it could, in principle, be solved by a quantum computer.

RESPONSIBLE AI

Trust and compliance are foundational to our enterprise AI strategy. Ensuring AI systems are developed responsibly and in ways that protect trust is vital, and we envision AI as a technology that can have a lasting positive impact for everyone. Responsible AI workloads ensure that systems are fair, interpretable, and reproducible even when sensitive demographic attributes or ground truth labels are unavailable. Without responsible AI capabilities, models risk violating regulations or eroding stakeholder trust. As illustrated in Figure 1, we address this with JURITY fairness metrics with surrogate groups, and BOOLXAI with an

interpretable Boolean classifier. These align with BP5 on *monitoring and evaluation*.

Fairness Evaluation: In the literature, various metrics that measure fairness in different ways have been proposed (Caton and Haas 2020). However, all standard fairness metrics require that group membership be known. Unfortunately, in many practical scenarios, this is not feasible. Fairness concerns often involve discrimination based on information that many people consider private, such as race, religion, and gender, which are legally protected in domains including housing (HUD.gov 2020), credit systems (FED 2020), and human resources (EEOC 2020). For the enterprise, this information is often hard to obtain, limited, and possibly illegal to collect, which invalidates most common fairness metrics for industry applications (Andrus et al. 2021).

As part of our enterprise strategy, we advocate that the lack of membership data should not exempt machine learning models from fairness evaluation. Although immediate information may not be available, there is often associated data that can be gathered. We show how to utilize surrogate membership information for fairness evaluation *without attempting to predict class membership for individuals* (Kadıoğlu and Thielbar 2024).

Our work extends the scope of fairness metrics to scenarios where fairness testing was impossible before, as showcased in Open at Intel[22]. Specifically, we relax the requirement of exact deterministic knowledge of protected membership for individuals to group-level information, which we refer to as *surrogates*. Surrogate groups provide estimates of membership in the protected class, and we demonstrate that, in the absence of individual-level information, surrogates provide an effective method for inferring fairness metrics. More broadly, our approach offers a generalization of the existing literature from the deterministic setting to its probabilistic counterpart (Thielbar et al. 2023). This also extends the previous literature on surrogate ground truth generation (Kadıoğlu and Michalský 2024; Michalský and Kadıoğlu 2021) for the analogous scenario where ground truth labels are unavailable, yet another dependency of existing fairness metrics. In combination, our research addresses the two severe limitations of the existing literature: the lack of ground truth and the lack of protected membership attributes. These methods are embodied in our open-source software Jurity[23]. The library further helped mitigate bias in recommender systems as detailed in (Du Cheng and Kadıoğlu 2022).

Explainable AI (XAI): The ever-increasing complexity of machine learning models due to their sophisticated inner workings and the sheer scale of their parameter space is not only a concern for inference. This also makes it difficult to understand and interpret their predictions.

Explainability is highly desirable in many applications and is even mandatory in several domains such as finance and healthcare due to industry regulations (Freitas 2014; Stöger, Schneeberger, and Holzinger 2021). For Responsible AI, explainable models help identify superfluous patterns and avoid unwanted bias. For Enterprise AI Strategy, explainable models are easier to deploy, debug, reproduce, maintain, and improve over time.

Through a unique collaboration between academia (Brown & CalTech), financial technology (AI Center at Fidelity & Fidelity Center of Applied Technology), and the high-tech industry (Amazon Quantum Solutions & AWS Center of Quantum Computing), we develop BoolXAI (Kadıoğlu et al. 2025), an interpretable classification approach based on expressive Boolean formulas[24]. Our partnership exemplifies BP6 on *multi-sector collaboration* as highlighted in the Amazon blog.[25]

In its simplest form, given a supervised dataset, the main idea behind BoolXAI is to find the smallest logical rule that satisfies the truth labels of the maximum number of samples. Under the hood, we formulate this as an optimization problem to design inherently interpretable classification models with tunable complexity. The formulation allows expressive Boolean operators that capture the classical conjunction (And) and disjunction (Or) operators, as well as AtMost(), AtLeast(), and Choose() to generate logical rules that best explain the given dataset. To solve this optimization problem efficiently, we search the feasible space of all formulas using local optimization.

More broadly, our approach is connected to integer linear programming (ILP) and Quadratic Unconstrained Binary Optimization (QUBO). As such, BoolXAI can be seen as an integration technology that brings together Boolean Satisfiability (SAT), Stochastic Local Search, ILP, and QUBO to provide a scalable and deployment-ready approach as detailed in (Rosenberg et al. 2023). We deploy BoolXAI as part of an explainability service, which we review next in a case study.

Case study: Explainability-as-a-Service

Our final enterprise application is geared toward non-technical business partners, exemplifying BP7 on *investing in explainability*. Even with our best efforts on software usability, BoolXAI, as a library, remains a tool for practitioners to utilize. Yet, explainability is in high demand by business stakeholders. To make XAI accessible to broader audiences, we developed Explainability-as-a-Service, powered by BoolXAI, as detailed in (Kadıoğlu et al. 2025). Through BoolXAI, we delivered explainability directly to business stakeholders without requiring programming. This reduced time-to-insight and

increased trust in AI outputs, as measured in stakeholder surveys.

This is an interactive web service that does not require programming. Users can upload their input data (or its path on Amazon S3) and examine the BoolXAI rules and visualizations. As explained earlier, users can seed their assumptions or extract base rules from previous runs to incrementally re-optimize the remaining formula. One of the strong assumptions of BoolXAI is that the input features are inherently interpretable. The service relaxes this assumption by offering a set of attributes commonly used in segmentation and analytics to its users out of the box.

Our key findings and lessons learned from this application are:

Complexity: As anticipated, among the most critical parameters for the interpretability were depth and complexity. Users typically kept the complexity of explanations below 15, with a maximum depth of less than 5.

Robustness: Users experienced confusion when BoolXAI identified various rules that significantly differed from each other yet performed similarly in terms of metrics, such as balanced accuracy. Empirically, users found that the best rule stabilizes when they first identify frequent features in the top-10 results and then run BoolXAI using only those features. We recommend this method in practice.

Visualization: Users frequently relied on BoolXAI's built-in visualization features to interpret the resulting rules and to share findings with non-technical users.

Runtime: Running BoolXAI on a dataset of 500,000 rows and 100 columns takes approximately 60 s on a modern laptop with an Intel i7 2.20 GHz processor using a single core. By default, this process runs 2000 samples for 500 iterations with 10 restarts. This translates into ∼0.01 sec per rule optimization iteration, enabling interactivity. Most users did not take advantage of parallelization, likely due to the quick iteration time, and the majority of their time was spent interpreting the generated rules.

Predefined User Rules: Surprisingly, users were rarely interested in running BoolXAI on the dataset as-is. Defining predetermined rules based on their existing business understanding proved to be much more valuable. Given this feedback, we developed a BoolXAI feature that allows users to optimize only part of a predefined rule. Users can select which features to include in the base rule, while BoolXAI optimizes the remaining parts of the formula, keeping the base rule fixed. This process resembles bootstrapping but is guided by the user, allowing for quick hypothesis testing.

Incremental Rule Generation: A similar usage pattern emerged for building formulas incrementally. By default, BoolXAI identifies the single best feature. Users utilized this best rule as the starting point and then optimized the rest of the formula. This approach enabled them to identify additional features to include in subsequent iterations. Users appreciated the controlled process of iterative formula generation, where they could introduce features incrementally, manage the rule's complexity, and decide when to stop. The fast runtime of BoolXAI facilitated these iterative scenarios.

Deployment Decisions: For technical users with access to sophisticated machine learning methods operating on high-dimensional data, an intriguing phenomenon has emerged. BoolXAI provided data scientists with a lower bound on baseline performance. This lower bound then served as a reference to validate the performance achieved by complex methods. The delta gap provides a systematic approach to reevaluating the need for additional complexity when making deployment decisions.

BEST DEPLOYMENT PRACTICES

Finally, let us conclude with best deployment practices based on our industry experience. Organizations looking to implement modular open-source AI strategies can benefit from the following practical recommendations:

[BP1] Modularity from the Start: Design with plug-and-play in mind. Use containerized microservices (e.g., via Docker and Kubernetes) to isolate each module, such as embedding, decision-making, and explainability, allowing them to evolve independently. For example, we deployed TEXTWISER as an independent service with a clear API, making it reusable across different applications without requiring the rewriting of logic or duplication of code.

[BP2] Small but Scalable: Build for scalability but start small. Begin with a focused use case (e.g., feature selection, pattern mining, or recommendation), validate the benefits of a modular architecture, and incrementally expand upon it. Use early wins to justify broader adoption and investment in modular infrastructure.

[BP3] Cost Management: Plan for performance optimization and monitor the compute and memory cost of individual modules in production. Modular designs help attribute performance trade-offs. Optimize compute-intensive modules (such as embedding generators) using batch processing, caching, or lightweight approximations (e.g., approximate search over complete embedding comparison). For example, we transitioned embedding generation to asynchronous batch TEXTWISER queries and made it available as an enterprise service.

[BP4] Prioritize Interoperability: Design frameworks that emphasize clear, standardized APIs and data

TABLE 2 Best deployment practices for modular enterprise AI, distilled from our experience in real-world implementations.

Recommendation	Rationale
Modularity from the start	Isolate change; enable independent evolution and reliability
Small but scalable	Validate on focused use cases before general availability
Cost management	Attribute costs to modules; batch, cache, approximate as possible
Prioritize interoperability	Standardize APIs & data contracts to reduce integration friction
Monitor and evaluate	Ensure fairness, performance, and reproducibility over time
Cross-functional collaboration	Align AI, Eng, UX, Business, and Compliance on boundaries
Invest in explainability	Boost trust, auditability, and faster decision cycles
Open-source culture	Community feedback and contributions increase quality & speed

formats to minimize integration friction. For example, we follow scikit-learn style methods for ease of use and utilize the bridge design pattern to decouple the interface from the details of algorithm implementation in our libraries.

[BP5] Monitor and Evaluate: Establish continuous monitoring and evaluation processes using quantitative and qualitative metrics to measure the impact of AI-driven solutions systematically.

[BP6] Cross-Functional Collaboration: The success of AI deployment relies on close collaboration between data scientists, software engineers, experience designers, business stakeholders, compliance teams, and multi-sector partnerships. Modular boundaries divide responsibilities, and regular reviews ensure alignment.

[BP7] Invest in Explainability: Choose modules with native explainability (e.g., BoolXAI) or integrate post-hoc explanation tools to ensure model decisions are transparent to users, auditors, and stakeholders to increase trust and confidence.

[BP8] Open-Source Culture & Education: Foster a culture of active contribution to open-source projects and maintain regular engagement with external communities to promote continuous learning and alignment with evolving best practices. We hosted researchers and students to collaborate on new open-source features. We also organized educational competitions to provide associates with hands-on experience using these tools (Kadıoğlu and Kleynhans 2024). In addition, thoughtful leadership in educational outreach is essential. Sharing accurate, self-contained AI content accessible to general audiences helps demystify complex concepts. To support this, we produced a video introducing the evolution of AI paradigms from classical approaches to modern and generative AI[26], which was recognized as the winner of the AAAI Educational AI Videos Competition.[27]

Table 2 consolidates our best deployment practices referenced across the various sections. We believe adopting these recommendations can streamline AI deployment and amplify organizational benefits.

BALANCING OPEN-SOURCE AND INTELLECTUAL PROPERTY

Finally, we outline considerations for balancing open-source contributions while protecting intellectual property. Organizations may approach this through a combination of legal, technical, and governance practices. One common approach is to establish clear boundaries between open-source components and proprietary assets. For example, some organizations choose to patent core innovations before releasing supporting components as open-source, helping ensure that strategic IP remains protected. The critical insight to realize is that *modular architectures enable this separation* by allowing organizations to open-source generic frameworks while retaining proprietary data pipelines, integration layers, and domain-specific logic. In our case, patented work spans multiple areas for domain-specific applications of resource allocation (Kadıoğlu 2018; Kadıoğlu and Colena 2017; Kadıoğlu et al. 2018), efficient testing (Kadıoğlu and Sebbah 2019; Strong et al. 2023), online orchestration (Jain et al. 2021; Pramod et al. 2022), surrogate methods for responsibility (Michalský and Kadıoğlu 2022), and constrained machine learning (Kilitçioğlu and Kadıoğlu 2024). Additionally, internal review processes and contributor agreements can help maintain compliance and clarify ownership. This dual strategy is intended to support innovation and ecosystem growth while managing IP considerations.

CONCLUSIONS AND FUTURE OUTLOOK

We presented a comprehensive Enterprise AI strategy grounded in modular and open-source frameworks. These tools, publicly available and actively used by over a hundred data scientists across our organization internally and many other practitioners externally, address key challenges such as scalability, interoperability, and responsible deployment of AI.

By promoting an open-source culture, this work has fostered several strategic partnerships across industry, academia, and public sector partners. These collaborations have enabled joint research, thereby accelerating innovation and broadening the adoption of advanced AI technologies. The successful deployment of modular AI components in real-world applications has contributed to improvements in operational efficiency in our deployments and enterprise-wide adoption.

In addition to our research contributions and open-source releases, we distilled a set of best deployment practices based on our real-world experience. These practices, ranging from modular design and cost management to explainability and cross-functional collaboration, serve as actionable guidance for organizations seeking to implement scalable and responsible AI systems.

Furthermore, our approach reinforces a long-term commitment to ethical and responsible AI. As the field continues to evolve, we anticipate that open-source tools and modular architectures will play a central role in enabling organizations to remain adaptable, transparent, and competitive.

ACKNOWLEDGMENTS

We extend our deepest gratitude to our co-authors, collaborators, and partners whose insights and contributions have been essential to this work. We are particularly indebted to our colleagues at Fidelity for cultivating a collaborative environment, providing critical domain expertise, and supplying the infrastructure that support our modular, open-source frameworks. We also thank our academic and industry partners for their support in advancing modular, open-source, and responsible AI. We also acknowledge the broader open-source community for its ongoing dedication to innovation and collective progress.

CONFLICT OF INTEREST STATEMENT

The author served as the lead guest editor for this Special Issue. The editorial and peer review process for this article was conducted independently, without the author's involvement. The author declares no conflict of interest related to the review or decision-making process for this article.

ENDNOTES

[1] Fidelity https://www.fidelity.com/about-fidelity
[2] AI Innovation at Fidelity https://tech-on-deck.castos.com/episodes/ai-innovation-at-fidelity
[3] Mab2Rec http://github.com/fidelity/mab2rec
[4] All Things Open blog https://tinyurl.com/4pkey59w
[5] NVIDIA & Fidelity https://www.youtube.com/watch?v=PWcNlRI00jo&t=4974s
[6] Comet ML RecSys Resources https://tinyurl.com/ykww64r7
[7] ACM RecSys https://github.com/ACMRecSys/recsys-evaluation-frameworks
[8] TextWiser http://github.com/fidelity/textwiser
[9] Selective http://github.com/fidelity/selective
[10] Seq2Pat http://github.com/fidelity/seq2pat
[11] CMU & Fidelity https://tinyurl.com/mr3fmutb
[12] MabWiser http://github.com/fidelity/mabwiser
[13] PathFinder http://github.com/skadio/pathfinder
[14] AIMMS-MOPTA Challenge https://tinyurl.com/ye27san7
[15] Brown Data Science Initiatives https://tinyurl.com/d8tf8rss
[16] Balans http://github.com/skadio/balans
[17] Ner4Opt https://github.com/skadio/ner4opt
[18] Text2Zinc https://huggingface.co/datasets/skadio/text2zinc
[19] Text2Zinc Leaderboard https://huggingface.co/spaces/skadio/text2zinc-leaderboard
[20] iCBS https://github.com/amazon-science/icbs
[21] Amazon & Fidelity https://tinyurl.com/mafef5yd
[22] Intel & Fidelity https://tinyurl.com/4r7ckkeb
[23] Jurity https://github.com/fidelity/jurity
[24] BoolXAI https://github.com/fidelity/boolxai
[25] Amazon & Fidelity https://tinyurl.com/4jswxj8d
[26] The Evolution of AI Paradigms https://www.youtube.com/watch?v=8SMmjBQ40YE
[27] AAAI Educational AI Video Competition https://tinyurl.com/4uyz6jva

REFERENCES

Adobe. 2025. "GitHub - Adobe Experience Manager." https://github.com/adobe-rnd/aem-experimentation-gh-actions. [Accessed 25-07-2025].

Akbik, A., D. Blythe, and R. Vollgraf. 2018. "Contextual String Embeddings for Sequence Labeling." In COLING 2018, 27th International Conference on Computational Linguistics, 1638–49.

Amatriain, X., and J. Basilico. 2016. "Past, Present, and Future of Recommender Systems: An Industry Perspective." In *ACM RecSys*, 211–14.

Andrus, M., E. Spitzer, J. Brown, and A. Xiang. 2021. "What We Can't Measure, We Can't Understand: Challenges to Demographic Data Procurement in the Pursuit of Fairness." In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 249–60.

Argyriou, A., M. González-Fierro, and L. Zhang. 2020. "Microsoft Recommenders: Best Practices for Production-Ready Recommendation Systems." In Proceedings of the Web Conference, 50–51.

Bengio, Y., A. Lodi, and A. Prouvost. 2021. "Machine Learning for Combinatorial Optimization: A Methodological Tour D'horizon." *European Journal of Operational Research* 290(2): 405–21.

Bolusani, S., M. Besançon, K. Bestuzheva, A. Chmiela, J. Dionísio, T. Donkiewicz, et al. 2024. "The SCIP Optimization Suite 9.0." *arXiv preprint arXiv:2402.17702*.

Boteanu, A., D. Cheng, and S. Kadıoğlu. 2023. "Read-Write-Learn: Self-Learning for Handwriting Recognition." In Proceedings of the ACM Symposium on Document Engineering 2023, 1–4.

Cai, J., S. Kadıoğlu, and B. Dilkina. 2025a. "Balans: Multi-Armed Bandits-based Adaptive Large Neighborhood Search for Mixed-Integer Programming Problem." In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-25*. International Joint Conferences on Artificial Intelligence Organization.

Cai, J., S. Kadıoğlu, and B. Dilkina. 2025b. "Balans: Multi-Armed Bandits-based Adaptive Large Neighborhood Search for Mixed-Integer Programming Problem." arXiv:2412.14382.

Caton, S., and C. Haas. 2020. "Fairness in Machine Learning: A Survey." *arXiv preprint arXiv:2010.04053*.

Cheng, D., D. Kilitçioğlu, and S. Kadıoğlu. 2022. "Bias mitigation in recommender systems to improve diversity." *CIKM (CEUR)*. https://ceur-ws.org, 3318.

Cire, A., S. Kadıoğlu, and M. Sellmann. 2014. "Parallel restarted search." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28.

Dakle, P. P., S. Kadıoğlu, K. Uppuluri, et al. 2023. "Ner4opt: Named entity recognition for optimization modelling from natural language." In International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, 299–319. Springer.

EEOC, U. 2020. "Prohibited Employment Policies/Practices." https://www.eeoc.gov/prohibited-employment-policiespractices. Accessed: 2020-09-04.

FED, U. 2020. "Federal Fair Lending Regulations and Statutes." https://www.federalreserve.gov/boarddocs/supmanual/cch/fair_lend_over.pdf. Accessed: 2020-09-04.

Fidelity Investments. 2024. "2024 Fidelity Investments Annual Report." *Technical report, Fidelity Investments*, Accessed: 2025-08-21.

Freitas, A. A. 2014. "Comprehensible classification models: a position paper." *SIGKDD Explorations Newsletter* 15(1): 1–10.

Ghosh, S., S. Yadav, X. Wang, B. Chakrabarty, and S. Kadıoğlu. 2022. "Dichotomic Pattern Mining Integrated with Constraint Reasoning for Digital Behaviour Analyses." *Frontiers in AI* 12(5): 868085.

Gurobi. 2024. "Gurobi Optimizer Reference Manual." Accessed: 2024-08-10.

Hoffmann, M., F. Nagle, and Y. Zhou. 2024. "The Value of Open Source Software." *Harvard Business School Working Paper*, 24-038.

Hollmann, N., S. Müller, L. Purucker, et al. 2025. "Accurate Predictions on Small Data with a Tabular Foundation Model." *Nature* 637(8044): 319–26.

Honnibal, M., and I. Montani. 2017. "spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing.".

Hosseininasab, A., W. van Hoeve, and A. A. Ciré. 2019. "Constraint-Based SPM with Decision Diagrams." In AAAI.

HUD.gov. 2020. "FAIR HOUSING RIGHTS AND OBLIGATIONS." https://www.hud.gov/program_offices/fair_housing_equal_opp/fair_housing_rights_and_obligations. Accessed: 2020-09-04.

Iansiti, M., and K. R. Lakhani. 2020. "Competing in the Age of AI: Strategy and Leadership When Algorithms and Networks Run the World." Boston: Harvard Business Review Press.

Jain, A., D. Gupta, S. Shekhar, B. Kleynhans, S. Kadıoğlu, and A. Arias-Vargas. 2021. "Automated predictive product recommendations using reinforcement learning." US Patent 10,936,961.

Jiang, A. Q., A. Sablayrolles, A. Mensch, et al. 2023. "Mistral 7B." *arXiv preprint arXiv:2310.06825*.

Kadıoğlu, S. 2018. "Systems and methods for providing dynamic and real time simulations of matching resources to requests." US Patent 10,146,665.

Kadıoğlu, S. 2019. "Core Group Placement: Allocation and Provisioning of Heterogeneous Resources." *EURO Journal on Computational Optimization* 7(3): 243–64.

Kadıoğlu, S. 2025. "Advancing Decision Science: Lessons from the Machine Learning Community."

Kadıoğlu, S., and M. Colena. 2017. "System and method of assigning requests to resources using constraint programming." US Patent 9,588,819.

Kadıoğlu, S., M. Colena, S. Huberman, and C. Bagley. 2015. "Optimizing the cloud service experience using constraint programming." In International Conference on Principles and Practice of Constraint Programming, 627–37. Springer.

Kadıoğlu, S., M. Colena, and S. Sebbah. 2016. "Heterogeneous resource allocation in Cloud Management." In 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), 35–38. IEEE.

Kadıoğlu, S., M. Colena, S. Sebbah, and M. M. Beg. 2018. "Assigning applications to virtual machines using constraint programming." US Patent 10,007,538.

Kadıoğlu, S., and B. Kleynhans. 2024. "The Design and Organization of Educational Competitions with Anonymous and Real-Time Leaderboards in Academic and Industrial Settings." arXiv:2402.07936.

Kadıoğlu, S., B. Kleynhans, and X. Wang. 2021. "Optimized Item Selection to Boost Exploration for Recommender Systems." In International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, 427–45. Springer.

Kadıoğlu, S., B. Kleynhans, and X. Wang. 2024. "Integrating optimized item selection with active learning for continuous exploration in recommender systems." *Annals of Mathematics and Artificial Intelligence* 92(6): 1585–607.

Kadıoğlu, S., Y. Malitsky, A. Sabharwal, H. Samulowitz, and M. Sellmann. 2011. "Algorithm Selection and Scheduling." In *Principles and Practice of Constraint Programming - CP 2011-17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, edited by J. H. Lee, *Lecture Notes in Computer Science*, vol. 6876, 454–69. Springer.

Kadıoğlu, S., Y. Malitsky, and M. Sellmann. 2012. "Non-Model-Based Search Guidance for Set Partitioning Problems." In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, edited by Hoffmann, J., and Selman, B., AAAI Press.

Kadıoğlu, S., and F. Michalský. 2024. "Fairness Evaluation for Uplift Modeling in the Absence of Ground Truth." arXiv:2403.12069.

Kadıoğlu, S., P. Pravin Dakle, K. Uppuluri, R. Politi, P. Raghavan, S. Rallabandi, et al. 2024. "Ner4Opt: Named Entity Recognition for Optimization Modelling from Natural Language." *Constraints*, 29: 261–99.

Kadıoğlu, S., and S. Sebbah. 2019. "Selecting a set of test configurations associated with a particular coverage strength using a constraint solver." US Patent 10,248,550.

Kadıoğlu, S., and M. Sellmann. 2009. "Dialectic Search." In International Conference on Principles and Practice of Constraint Programming, 486–500. Springer.

Kadıoğlu, S., M. Sellmann, and M. Wagner. 2017. "Learning a Reactive Restart Strategy to Improve Stochastic Search." In International Conference on Learning and Intelligent Optimization, 109–23. Springer.

Kadıoğlu, S., X. Wang, A. Hosseininasab, and W.-J van Hoeve. 2023. "Seq2Pat: Sequence-to-Pattern Generation to Bridge Mining with Machine Learning." *AI Magazine* 44(1): 54–66.

Kadıoğlu, S., E. Y. Zhu, G. Rosenberg, et al. 2025. "BoolXAI: Explainable AI Using Expressive Boolean Formulas." *Proceedings of the AAAI Conference on Artificial Intelligence* 39(28): 28900–28906.

Kadıoğlu, S., and M. Thielbar. 2024. "Surrogate Modeling to Address the Absence of Protected Membership Attributes in Fairness Evaluation." *ACM Transactions on Evolutionary Learning* 5(3): 1–25.

Kilitçioğlu, D., and S. Kadıoğlu. 2021. "Representing the Unification of Text Featurization using a Context-Free Grammar." *Proceedings of the AAAI Conference on Artificial Intelligence* 35(17): 15439-45.

Kilitçioğlu, D., and S. Kadıoğlu. 2022. "Non-Deterministic Behavior of TS with Linear Payoffs and How to Avoid It." *TMLR*, 2022.

Kilitçioğlu, D., and S. Kadıoğlu. 2024. "Automatic Data-Driven Optimization of a Target Outcome Using Machine Learning." US Patent 12,169,870.

Kleynhans, B., X. Wang, and S. Kadıoğlu. 2021. "Active Learning Meets Optimized Item Selection." arXiv:2112.03105.

Liberto, G. M. D., S. Kadıoğlu, K. Leo, and Y. Malitsky. 2016. "DASH: Dynamic Approach for Switching Heuristics." *European Journal of Operational Research* 248(3): 943–53.

Maslej, N., L. Fattorini, R. Perrault, et al. 2025. "Artificial Intelligence Index Report 2025." *arXiv preprint arXiv:2504.07139*, https://arxiv.org/abs/2504.07139.

Michailidis, K., D. Tsouros, and T. Guns. 2024. "Constraint Modelling with LLMs Using In-Context Learning." In *30th International Conference on Principles and Practice of Constraint Programming (CP 2024)*, edited by Shaw, P., vol. 307, 20:1–20:27. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-336-2.

Michalský, F., and S. Kadıoğlu. 2021. "Surrogate Ground Truth Generation to Enhance Binary Fairness valuation in Uplift Modeling." In IEEE ICMLA, 1654–59.

Michalský, F., and S. Kadıoğlu. 2022. "Surrogate Ground Truth Generation in Artificial Intelligence based Marketing Campaigns." US Patent App. 17/122,362.

Naumov, M., D. Mudigere, H.-J. M. Shi, et al. 2019. "Deep Learning Recommendation Model for Personalization and Recommendation Systems." *arXiv preprint arXiv:1906.00091*, https://arxiv.org/abs/1906.00091.

Nikitin, N. O., P. Vychuzhanin, M. Sarafanov, and I. S. Polonskaia. 2021. "Automated Evolutionary Approach for the Design of Composite Machine Learning Pipelines." *Future Generation Computer System* 127: 109–25.

NVIDIA. 2020. "NVIDIA Data Center Deep Learning Product Performance." https://developer.nvidia.com/deep-learning-performance-training-inference.

Oldridge, E., J. Perez, B. Frederickson, et al. 2020. "Merlin: A GPU Accelerated Recommendation Framework." *IRS*.

Pasumarthi, R. K., S. Bruch, X. Wang, et al. 2019. "TF-Ranking: Scalable Tensorflow Library for Learning-to-Rank." In KDD.

Pei, J., J. Han, and W. Wang. 2007. "Constraint-Based Sequential Pattern Mining: The Pattern-Growth Methods." *Journal of Intelligent Information Systems* 28(2): 133–60.

Pramod, R., A. Pradhan, S. Shekhar, S. Kadıoğlu, and A. Arias-Vargas. 2022. "Digital content classification and recommendation based upon artificial intelligence reinforcement learning." US Patent 11,361,239.

Qu, J., D. Holzmüller, G. Varoquaux, and M. L. Morvan. 2025. "TabICL: A Tabular Foundation Model for In-Context Learning on Large Data." *arXiv preprint arXiv:2502.05564*, https://arxiv.org/abs/2502.05564

Requena, B., G. Cassani, J. Tagliabue, C. Greco, and L. Lacasa. 2020. "Shopper Intent Prediction from Clickstream E-Commerce Data with Minimal Browsing Information." *Scientific Reports* 10: 16983.

Rosen, M. 2025. "Open GenAI Models, 2024: Benefits, Experimentation, and Deployment." IDC: The Premier Global Market Intelligence Company. https://my.idc.com/getdoc.jsp?containerId=US52477724

Rosenberg, G., J. K. Brubaker, M. J. A. Schuetz, et al. 2023. "Explainable Artificial Intelligence Using Expressive Boolean Formulas." *Machine Learning and Knowledge Extraction* 5(4): 1760–95.

Rosenberg, G., J. K. Brubaker, M. J. A. Schuetz, et al. 2024. "Scalable Iterative Pruning of Large Language and Vision Models using Block Coordinate Descent." arXiv:2411.17796.

Samuel, S. G., E. Areyan Viqueira, and S. Kadıoğlu. 2021. "Integrated Vehicle Routing and Monte Carlo Scheduling Approach for the Home Service Assignment, Routing, and Scheduling Problem." *CoRR*, abs/2106.16176.

Sebbah, S., C. Bagley, M. Colena, and S. Kadıoğlu. 2016. "Availability Optimization in Cloud-Based In-Memory Data Grids." In International Conference on Principles and Practice of Constraint Programming, 666–79. Springer.

Singirikonda, A., S. Kadıoğlu, and K. Uppuluri. 2025. "Text2Zinc: A Cross-Domain Dataset for Modeling Optimization and Satisfaction Problems in MiniZinc." arXiv:2503.10642.

Stöger, K., D. Schneeberger, and A. Holzinger. 2021. "Medical Artificial Intelligence: The European Legal Perspective." *Communications of the ACM* 64(11): 34–36.

Strong, E., S. Kadıoğlu, M. Jain, F. Michalský, A. Arias-Vargas, and S. Narayanan. 2023. "Determining future user actions using time-based featurization of clickstream data." US Patent 11,799,734.

Strong, E., B. Kleynhans, and S. Kadıoğlu. 2019. "MABWiser: A Parallelizable Contextual MAB Library for Python." In IEEE ICTAI.

Strong, E., B. Kleynhans, and S. Kadıoğlu. 2021. "MABWiser: Parallelizable Contextual Multi-armed Bandits." *International Journal on Artificial Intelligence Tools* 30(4): 2150021.

Sun, M., Z. Liu, A. Bair, and J. Z. Kolter. 2023. "A Simple and Effective Pruning Approach for Large Language Models." *arXiv preprint arXiv:2306.11695*.

The White House. 2025. "Winning the Race: America's AI Action Plan." Accessed on July 25, 2025.

Thielbar, M., S. Kadıoğlu, C. Zhang, R. Pack, and L. Dannull. 2023. "Surrogate Membership for Inferred Metrics in Fairness Evaluation." In *Learning and Intelligent Optimization - 17th International Conference, LION 17, Nice, France, June 4-8, 2023*, edited by Sellmann, M., and Tierney, K., vol. 14286, 424–42. Springer.

Tosa. 2025. "GitHub - Tosaaaki/piphawk-ai — github.com." https://github.com/Tosaaaki/piphawk-ai. [Accessed 25-07-2025].

Touvron, H., M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. 2021. "Training Data-Efficient Image Transformers & Distillation Through Attention." In International Conference on Machine Learning, 10347–57. PMLR.

Tsouros, D., H. Verhaeghe, S. Kadıoğlu, and T. Guns. 2023. "Holy Grail 2.0: From Natural Language to Constraint Models." arXiv:2308.01589.

Verma, G., S. Sengupta, S. Simanta, et al. 2023. "Empowering RecSys Using Automatically Generated KG and RL." *arXiv:2307.04996*.

Wang, X., A. Hosseininasab, P. Colunga, et al. 2022. "Seq2Pat: Sequence-to-Pattern Generation for Constraint-Based Sequential Pattern Mining." *Proceedings of the AAAI Conference on Artificial Intelligence* 36(11): 12665-71.

Wang, X., and S. Kadıoğlu. 2019. "Bayesian Deep Learning Based Exploration-Exploitation for Personalized Recommendations." In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 1715–19. IEEE.

Wang, X., and S. Kadıoğlu. 2022. "Dichotomic Pattern Mining with Applications to Intent Prediction from Semi-Structured Clickstream Datasets." In *arXiv preprint arXiv:2201.09178*.

Wang, X., and S. Kadıoğlu. 2023. "Modeling Uncertainty to Improve Personalized Recommendations via Bayesian Deep Learning." *International Journal of Data Science and Analytics* 16(2): 191–201.

Wei, R., Y. Liu, Y. Hou, H. Cui, Y. Zhang, and K. Zhou. 2025. "TopTune: Tailored Optimization for Categorical and Continuous Knobs Towards Accelerated and Improved Database Performance Tuning." In 2025 IEEE 41st International Conference on Data Engineering (ICDE), 613–26. IEEE Computer Society.

Wolf, T., L. Debut, V. Sanh, et al. 2019. "HuggingFace's Transformers: State-of-the-art Natural Language Processing." *ArXiv*, abs/1910.03771.

Wouda, N. A., and L. Lan. 2023. "ALNS: A Python Implementation of the Adaptive Large Neighbourhood Search Metaheuristic." *Journal of Open Source Software* 8(81): 5028.

Xiao, H. 2018. "bert-as-service." https://github.com/hanxiao/bert-as-service.

Yilmaz, A., J. Cai, S. Kadıoğlu, and B. Dilkina. 2025. "ParBalans: Parallel Multi-Armed Bandits-based Adaptive Large Neighborhood Search." arXiv:2508.06736.

Zhang, X., Z. Chang, H. Wu, et al. 2023. "A Unified and Efficient Coordinating Framework for Autonomous DBMS Tuning." *Proceedings of the ACM on Management of Data* 1(2): 186.

Zhao, J., and J. Pan. 2023. "QoE-driven Joint Decision-Making for Multipath Adaptive Video Streaming." In GLOBECOM 2023-2023 IEEE Global Communications Conference, 128–33.

How to cite this article: Kadıoğlu, S. 2025. "Open-source AI at scale: Establishing an enterprise AI strategy through modular frameworks." *AI Magazine* 46: e70032. https://doi.org/10.1002/aaai.70032

AUTHOR BIOGRAPHY

Serdar Kadıoğlu is the Group VP of Artificial Intelligence in the AI Center of Excellence at Fidelity Investments and an adjunct associate professor in the Department of Computer Science at Brown University. He serves as the co-chair of the Innovative Applications of AI (IAAI) and the co-organizer of the Open-Source AI for Mainstream Use at AAAI.