

Fostering Inexperienced User Participation in ML-based Systems Design: A Literature Review of Visual Language Tools

Serena Versino^{1,*}, Tommaso Turchi¹ and Alessio Malizia^{1,2}

¹University of Pisa, Largo Bruno Pontecorvo, 3, 56127 Pisa (Italy)

²Molde University College, Molde (Norway)

Abstract

The application of Artificial Intelligence (AI) technologies in various sectors is based on machine learning (ML) systems, which, despite their transformative potential, can be complex and opaque for non-technical users. This review explores the role of Visual Programming Languages (VPLs) in lowering these barriers, enhancing the accessibility of ML-based system design for domain experts. We examine the application of ML processes through VPLs, seeking tools that open AI to a broader audience while identifying current challenges and future research directions. Bridging the gap between experts and the broader society is necessary, especially in sectors where responsible and trustworthy AI systems play a pivotal role in decision-making. By democratizing AI, we aim to provide socio-technical conditions that enable users with diverse background to actively contribute to the design of ML-based systems, enhancing their understanding and trust. Therefore, this literature review addresses also how VPL-based tools incorporate features for interpretability and collaboration. Our findings reveal that tools either lack comprehensive customizability, demand computing proficiency, or lack interpretability features. These limitations can affect a synergistic communication between users and intelligent systems, uncovering a research gap in the development of VPLs suited for novices engaged in the design of ML-based systems.

Keywords

visual programming language, participation, AI democratization, machine learning

1. Introduction


Nowadays, Artificial Intelligence (AI) is transforming business, academia, and socio-cultural dynamics alike. AI applications range widely, from facilitating language translation and email spam filtering to enhancing virtual personal assistant functionalities for scheduling. Moreover, AI is instrumental in refining medical diagnoses, boosting agricultural efficiency, aiding in climate change efforts, and increasing production system efficiency via predictive maintenance [1]. Therefore, AI integration across diverse sectors has the potential to drive innovation in product development, decision-making processes, and organizational efficiencies, marking a


Proceedings of the 1st International Workshop on Designing and Building Hybrid Human-AI Systems (SYNERGY 2024), Arenzano (Genoa), Italy, June 03, 2024.

*Corresponding author.

✉ serena.versino@phd.unipi.it (S. Versino); tommaso.turchi@unipi.it (T. Turchi); alessio.malizia@unipi.it (A. Malizia)

ORCID 0000-0002-9860-9142 (S. Versino); 0000-0001-6826-9688 (T. Turchi); 0000-0002-2601-7009 (A. Malizia)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

pivotal shift in operational paradigms [2, 3]. Educational institutions are similarly adapting, revising pedagogical approaches to integrate AI, reflecting its transformative impact on teaching and learning methodologies [4].

Recent trends in AI development are propelling society towards an increasingly algorithmic era [5]. The European Commission's white paper emphasizes that this trajectory of AI will significantly influence our future, though the exact nature of AI's interaction with people and its subsequent impact remains uncertain [1]. Although AI systems are perceived as fair and precise, their performance can vary significantly across different domains. AI technology can entail a number of potential risks, such as opaque decision-making, gender-based or other kinds of discrimination. For example, recommender systems utilize algorithms to manipulate search engine outcomes based on user inquiries, thus impacting consumption decisions [6], shaping public opinion, and societal perceptions [7]. These systems filter and prioritize information based on underlying factors such as browsing history and demographic data [8].

At the core of AI technology lie sophisticated ML-based systems, trained on human data that encompasses a broad spectrum of demographics, cultures, and personal traits of those who generate them. The growing complexity of algorithms has centralized their development and management among a small group of technical experts, such as software developers, and increased society's dependence on their expertise [9]. Domain specialists are often excluded from the design process of ML-based systems, which limits their understanding of these systems and relegates them to the role of mere end-users. However, individuals with high computing proficiency often lack insight into the specific operational domains of their applications. This gap raises concerns about the societal impact, transparency, and trustworthiness of ML-based systems [10, 11]. Therefore, closing the knowledge divide between domain specialists and computing professionals is crucial for ensuring ethical and fair decision-making in these systems [11]. This objective can be achieved by facilitating broader participation in the design of ML-based systems across different levels of expertise.

The democratization of AI encourages participation from a broad user base by fostering socio-technical ecosystems that equip diverse societal segments with the tools to navigate the challenges brought by AI advancements. Therefore, AI democratization seeks to harmonize the technical knowledge of computing professionals with the nuanced understanding of domain-specific practitioners, ensuring that AI systems are ethically aligned and contextually relevant [12]. End-User Development (EUD) has emerged as a pivotal strategy for this cultural transformation. It enables users to transition from passive roles, such as consumers of artifacts and systems, to active roles, like designers [12, 13]. By facilitating knowledge reformulation, enabling creative expression, and fostering content generation, EUD allows diverse audiences to design and create their own tools and artifacts.

This cultural transformation has given rise to cultures of participation, where multidisciplinary teams collaborate within socio-technical settings to achieve common goals [14, 15]. These teams span the spectrum of computer users: from those who program such as computing professionals to those who use applications for productivity such as domain specialists. While the objective is to empower domain specialists to develop and modify systems, it does not shift the burden of designing high-quality systems to them. Instead, EUD and Human-Centered AI (HCAI) offer the necessary support for end-users, who are most familiar with their requirements, to adapt and improve their systems. HCAI research, for example, explores innovative methods

to engage novice users through visual user interfaces [16]. In educational contexts, tools like Visual Programming Languages (VPLs) and no-code platforms such as Scratch [17] prioritize user-friendly experiences by simplifying complex computational operations. By engaging users in the design of ML-based systems through such participatory approaches can support the broad appropriation and integration of trustworthy AI technologies across various domains [18, 19].

This work is based on the main research question: ‘Can VPL-based frameworks foster the participation of both novice and expert practitioners in the design of ML-based systems?’ This study contributes to the research in Hybrid Human-AI Systems exploring the application of ML techniques through VPLs, aiming to reveal how VPLs can democratize AI and promote a synergistic communication between novice users and ML-based systems. Research for this review was conducted through a search of publications within the ACM and IEEE digital libraries.

This study is organized as follows: Section 1 introduces the topic. Section 2 provides background and highlights contributions from EUD in VPLs. Section 3 discusses related works. Section 4 outlines the methodology for the literature review, detailing the data collection, search processes, exclusion criteria, and paper selection. Section 5 delves into the literature analysis. Finally, Section 6 focuses on the discussion and conclusions.

2. Background

This section explores the historical progression and current state of EUD and VPLs, along with advancements in user interface technologies. It also discusses the integration of Explainable AI (XAI) techniques into the design of ML-based systems to enhance domain specialists’ trust and understanding of these systems.

2.1. End-User Development (EUD)

Since the 1960s, the development of various programming languages has been driven by the goal of enhancing coding accessibility, catering to educational purposes and user empowerment [20]. Initially, software development was predominantly the domain of computing specialists, which left end-users with little to no influence over the design and functionality of software [21]. The advent of EUD in the late 1980s, coupled with advancements in personal computing, marked a paradigm shift in this dynamic. EUD revolutionized the way users interact with software by enabling them to configure systems and develop applications, thereby democratizing software design and modification beyond what was previously possible within the domain of professional software engineering [13]. This transformation covered the entire software development life-cycle [22]. Central to this transformation was the adoption of participatory design principles, engaging end-users directly in the system design process. Such participation transformed users from passive participants into active contributors, who could influence software design without needing extensive coding skills [13]. Concurrently, advances in AI technology began to emerge as powerful tools for solving real-world problems. These advancements brought a renewed focus to computing, ranging from knowledge representation and utilization to system assembly, and encompassing activities such as perception, reasoning, and decision-making [23]. Despite its advantages, the application of EUD often focused on short-term problem-solving, occasionally

sidestepping the traditional, more complex methodologies necessary for developing sustainable, long-term AI applications. This tendency persisted until recent years, when a growing body of research began to support efforts to bridge the knowledge and involvement gap between professional software designers and end-users.

2.2. Visual Programming Languages (VPLs)

To overcome the technical barriers that novices face with coding, educational approaches have incorporated visual components that intuitively represent programming concepts, like pressing buttons or spatial movement. For instance, VPLs utilize visual representations of programming logic, facilitating an intuitive approach to software development [24]. At the core of programming languages are syntax and semantics, respectively the structure of the language and the meaning conveyed. In the review by Kuhail et al. [25], the merge of two well-established taxonomies, that is Myers [26] and Burnett and Baker [27], provides four distinct categories of VPLs. They include block-based, form-based, diagram-based and icon-based languages. Block-based languages simplify programming by allowing users to construct programs using drag-and-drop code blocks, thus reducing syntax errors and focusing on conceptual understanding (e.g., tools like Scratch [17] and TAPAS [28]). Icon-based languages use graphical icons, easing the integration of diverse content sources and supporting novices in creating Personal Information Spaces [29]. Form-based languages enable the configuration of forms and computational cells through both textual and visual elements, facilitating the definition of data interdependencies [30]. Diagram-based or flow-based languages employ a data flow paradigm represented as directed graphs [31, 32], making complex data processing understandable through visual nodes and arcs, such as Grasshopper [33] used in architecture domain.

2.3. Graphical, Tangible and Natural Interfaces

VPLs integrate visual elements into syntax that can enable inexperienced users to design and improve software via graphical interfaces [34]. Graphical User Interfaces (GUIs) and Tangible User Interfaces (TUIs) represent significant advancements in facilitating the comprehension of intricate concepts through interactive engagement and manipulation. GUIs, traditionally based on mouse and keyboard inputs, constrain user interactions to predefined mechanisms. TUIs leverage direct manipulation of physical objects such as blocks or cards to enhance the understanding of complex concepts, thereby accelerating improvements in software usability [35]. Further evolution has led to the development of Natural User Interfaces (NUIs), which exploits innate human capabilities such as touch, vision, and speech, offering an intuitive and natural means of digital interaction [36]. NUIs can utilize diverse mediums for digital interaction. Through cameras and sensors, they enable touch interfaces that allow direct manipulation of digital content via touchscreens. For instance, voice recognition devices allow users to interact using natural language commands, while gesture recognition devices interpret body movements, and facial expression recognition devices enable interfaces to respond to users' emotions. NUIs also extend into augmented and virtual reality, enabling interactions with digital content overlaid on the real world or in virtual environments.

2.4. Explainable AI (XAI)

The challenges encountered by novices entering AI technology extend beyond computing barriers. In recent years, the inherently complex nature of ML-based systems has raised ethical concerns regarding the fairness of their decision-making processes and their explainability. For instance, cases including the investigation into Goldman Sachs for gender-based credit discrimination¹, observed biases in Amazon’s automated hiring processes², and ethnic disparities in the COMPAS algorithm³, uncovered the need for improved transparency in such processes. These instances showed that the successful adoption of ML-based systems in their domain-specific applications relies on decision makers’ comprehension and trust. Similar to human interactions, trust in ML-based systems should be established on a foundation of mutual understanding and shared values. Indeed, our confidence in these systems increases when we understand their underlying processes, enabling us to intervene and ensure that decision-making aligns with ethical standards [12]. At the current state, decision makers, who are domain specialists, adopt AI technology as end-users, meaning they are not necessarily ML experts. However, they require a clear understanding of ML-based systems to make informed decisions about their deployment.

To tackle these challenges, researchers have developed frameworks such as Shneiderman’s model for HCAI [16]. This framework emphasizes methodologies that ensure human control, interpretability, and transparency, while enhancing the automation of ML-based systems [37]. In the realm of XAI research, this is facilitated, for example, through the application of SHapley Additive exPlanations (SHAP). SHAP is a model-agnostic approach that employs game theory to assign importance values to features for individual predictions. This technique generates data perturbations to measure the impact on model output, aiding in detecting potential biases [38]. Despite significant advancements, the development of XAI techniques remains in its early stages, particularly in the realm of data visualizations [39]. The complexity of these techniques often challenges novices, providing only a partial glimpse into the underlying ML processes, which still appear as black boxes to domain specialists. Current research in XAI and HCAI aims to refine interpretability methods by incorporating more effective techniques [40] and to develop strategies that directly involve domain specialists in the design of ML-based systems [16].

3. Related works

The body of literature shows an enduring interest in VPLs and user interfaces within the field of Human-Computer Interaction (HCI). Daniel D. Hils [41] anticipated that flow-based languages could widen the appeal of visual programming by applying it to new domains, introducing visual programming to domain specialists. Boshernitsan and Downes [42] observed a shift towards graphical displays in VPLs but cautioned against abandoning text-based languages due to challenges in readability and navigation. Later, Rouly et al. [43] emphasized the importance of

¹MIT Technology Review: Gender Bias in Goldman Sachs’ Apple Card Algorithm

²Reuters: Bias in Amazon’s AI Recruitment Process

³ProPublica: Ethnic Bias in COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) Risk Assessment Algorithm

user interface design in the usability of integrated development environments (IDEs), suggesting a design approach that favors simplicity and user-centric controls. They highlighted the role of incorporating HCI theories, such as those proposed by Green and Petre [44] regarding cognitive dimensions in visual programming, to enhance IDE design and usability. Studies by Mason and Dave [45] explored the benefits of VPLs in reducing the complexity associated with programming, thereby making these tools more accessible to novices. Further exploring the educational impact of VPLs, Noone and Mooney [46] examined their effects on learning programming, observing that VPLs can lead to increased interest among students. They cited the example of Scratch [17], a block-based language recognized for its ability to lessen the cognitive load on learners, thus enabling them to concentrate more on understanding programming concepts rather than tackling the intricacies of syntax. This approach has been integrated into new educational taxonomies designed to leverage the advantages of VPLs in the educational domain [47]. However, further explorations suggested that flow-based languages could offer a more intuitive understanding of programming concepts for beginners compared to block-based languages [45]. Meanwhile, Ray [48] delved into the ecosystem surrounding VPLs, reporting their extensive use in system simulation and multimedia, as well as the predominance of open-source environments. Despite their advantages in visualizing programming logic, facilitating logical understanding, and enhancing portability across various devices, VPLs faced challenges such as poor user interfaces, slow code generation, a lack of standardized models, and an absence of abstraction layers that hindered their growth. Then, Kuhail et al. [49] pointed out the lack of studies analyzing evidence-based visual programming approaches in domains beyond robotics, IoT, and education, highlighting an emerging interest in interactive displays, AI context, and data science. They reported a sharp increase in VPL research publications between 2017 and 2019, focusing on block-based and flow-based languages. Key evaluation metrics identified in their survey included completion time, number of errors, perceived usability, usefulness, workload, and cognitive dimensions [44]. The authors emphasized the need for integrating conversational agents and ML models to aid end-users in developing and debugging visual programming projects, suggesting a forward path for enhancing the accessibility and efficiency of VPLs. In their recent work, El Kamouchi et al. [50] study the use of low-code/no-code (LC/NC) technologies in web/mobile development and healthcare, observing a widespread adoption in AI-powered systems. They emphasize the advantages of LC/NC technologies in reducing costs and accelerating development, while also pointing out ongoing challenges, like restrictions associated with proprietary software and performance issues.

Across the surveyed literature, the authors identify challenges and limitations of VPL-based tools, such as inadequate user interfaces, the absence of standardized models, limited user-friendliness for beginners, and the complexity inherent in ML-based applications. This review addresses this research gap by examining the application of ML techniques through VPLs, including the presence of efforts to enhance trust and comprehension in ML decision-making processes.

4. Methodology

Following Kitchenham and Charters [51] framework, our analysis began with a planning phase dedicated to reviewing the existing literature on VPLs. This preliminary investigation highlighted a gap in literature on VPL-based systems within the realm of ML for domain experts. Then, we formulated and concurred on the research questions and established a review protocol. This protocol outlined the search strategy and determined the criteria for including and excluding studies. Following the retrieval of articles from selected databases, we carried out the execution phase, characterized by a two-step screening process. Initially, articles were screened based on their titles and abstracts, followed by a more detailed examination using the defined exclusion criteria. Throughout this second stage, the pertinence of each paper to our review was evaluated. In the final phase, the articles that met our criteria were analyzed to respond to the research questions and to report the findings.

In this section, we outline the rationale behind our research questions (4.1), detail the search process (4.2), define the exclusion criteria (4.3), and present the paper selection derived from this procedure (4.4), which resulted in the identification of 38 most pertinent articles published between 1994 and 2024 from a pool of 2,363 collected papers.

4.1. Research Questions

The research questions are crafted to explore the application of VPL-based tools in the ML context for domain specialists, aiming to uncover areas that require further exploration. Our aim is to examine the use of VPL-based tools, identify the application domains generating the most interest, investigate the types of VPLs employed, and how user experience and usability have been assessed. Such questions will offer an overview of the field, including both technological facets, user and application considerations. Our literature review addresses the following research questions:

RQ1: Which VPL-based tools have been used in designing ML-based systems? We aim at uncovering the technical features of VPL-based tools within the ML application, fostering a deeper understanding of their strengths and limitations.

RQ2: Which kind of VPL-based tools for ML-based system design are available, and in what ways have they been implemented? By investigating the various types of VPLs used (e.g., block-based or flow-based), we aim at revealing underexplored areas and potential limitations in current methodologies.

RQ3: What are the ML application domains where VPL-based tools find their use? This question aims to highlight the domains that have been the focus of research, shedding light on explored areas and opportunities for further development.

RQ4: What access modalities are available for designing ML-based systems? Exploring the range of access modalities will enable us to identify potential limitations within existing solutions.

RQ5: What is the background of users who have used VPL-based tools in ML application domains? We seek to identify user profiles, whether the primary users are computing experts or domain experts.

RQ6: How have the usability and user participation of VPL-based systems been evaluated? Grasping how usability and user participation assessments are applied can determine their

current scope and potential for advancements in research.

4.2. Search Process

We collected publications from the digital repositories of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) as of February 2024. We executed searches using a keyword string designed to capture studies intersecting the domains of VPLs and ML. Our search strategy employed the following keywords and phrases: ("visual programming language" OR "visual language" OR "visual programming" OR "visual programming environment" OR "visual environment") AND ("graphical user interface" OR "graphical interface" OR "software" OR "visual block" OR "visual graph" OR "Block based" OR "Flow based") AND ("machine learning" OR "deep learning" OR "data mining")

4.3. Exclusion criteria

We defined a specific set of selection criteria to assess the relevance of papers to our study. These criteria were applied as follows: 1) the papers must be authored in English; 2) each paper must include a title, abstract, and keywords for accurate identification in order to maintain the integrity of the selection process; 3) the focus of the papers must be on the application of VPLs in the context of ML. Studies that concentrate solely on interaction with a single object were not considered; 4) the papers extending four pages or more were included as providing enough content for a thorough analysis.

4.4. Paper selection

We found a total of 2,363 articles across the chosen digital libraries. Among them, 1,538 articles were sourced from the Institute of Electrical and Electronics Engineers (IEEE) library, with the other 825 articles coming from the ACM Digital Library. We compiled references to these articles in BibTeX format, subsequently processing them with 'bibtexparser' and 'pandas' Python libraries. Details of the selection procedure are concisely illustrated in Fig. 1 by the PRISMA flow diagram.

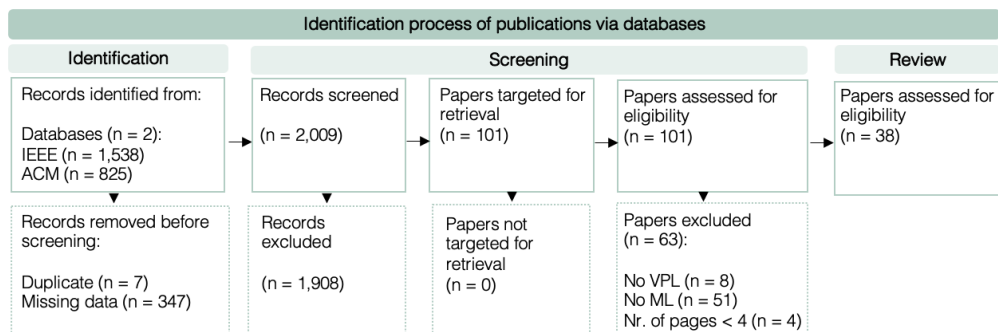


Figure 1: Identification process for paper selection

During the initial screening phase of the 2,363 articles, which considered the title, abstract, keywords, and authors, we removed 7 duplicates through manual review. Additionally, 347 articles were excluded due to missing data necessary for correct identification of the paper, and 1,908 articles were removed for not aligning with our research focus (such as those exclusively discussing either VPLs or ML applications, or systematic reviews solely on VPLs or ML). Then, we applied the exclusion criteria to the remaining 101 articles to ascertain their final relevance and suitability for inclusion in our study. Afterward, we excluded an additional 63 articles, primarily for lack of relevant VPLs aspects linked to ML application, that is minimal or no use of VPL-based tools, and insufficient length. Finally, the search phase led to a collection of 38 articles.

5. Literature Analysis

The publication timeline presented in Tab. 1 reveals an early phase of exploration for VPLs within the context of ML during the early 1990s. Despite the initial introduction of VPLs in the 1980s, this period is marked by an evolution in the field, which has led to the sophisticated technologies we see today. Since 2018, a significant rise in interest towards VPLs has emerged, mirroring the need of user-friendly tools alongside the escalating complexity and wide adoption of AI-based systems.

Table 1

Distribution of selected articles per year

1994	2005	2006	2013	2014	2015	2017	2018	2019	2020	2021	2022	2023	Total
1	1	1	2	1	1	1	5	5	3	5	5	7	38

A synthesis of the primary themes related to our research questions is provided in Fig. 2. The diagram features boxes corresponding to each research question, organizing categories found in the content analysis of the collected articles. Each category provides the count of associated contributions. In the box for UX evaluation methods, articles are cross-referenced as studies can employ diverse evaluation metrics (see Tab. 2 in Appendix for the full list of evaluation metrics).

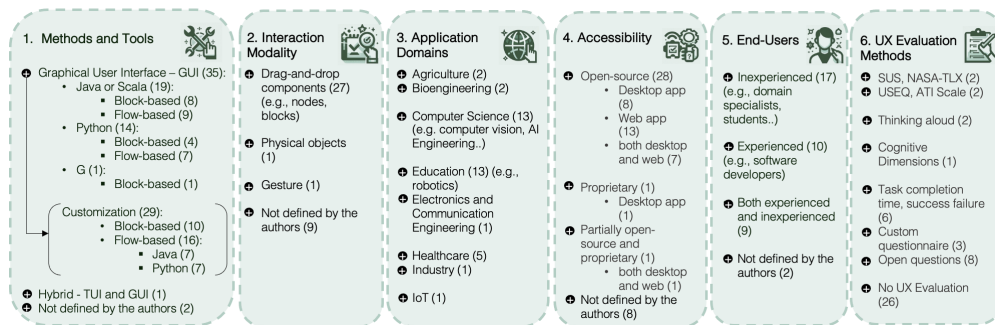


Figure 2: Summary of the key aspects of the research questions and literature review contributions

5.1. Methods and Tools for VPLs in ML

The methods and tools section in Fig. 2 aim at addressing two research questions, **RQ1**: “Which VPL-based tools have been used in designing ML-based systems?” and **RQ2**: “What types of VPL-based tools for ML-based system design are available, and how have they been implemented?”

Software is typically developed using text-based programming languages, such as Java and Python, and is often coupled with user interfaces to enhance the understanding of complex concepts through interactive engagement and manipulation [25]. These interfaces may include GUIs, TUIs, and NUIs. Given this context and that many publications do not thoroughly detail the visual language used for ML application (whether block or flow-based) or specify the programming language employed, our study focuses on the information explicitly provided by the authors. This review uncovered 35 tools providing GUIs for ML-based system design. We identified 19 Java-based tools — including 8 block-based examples such as Prompt Sapper [52] and 9 flow-based like Visual Apriori [53] — alongside 14 Python tools, of which 4 are block-based (e.g., Milo [54]).

5.1.1. Customization

Customization is pivotal for users aiming to tailor ML-based systems to specific domain requirements. However, in this review many tools were not explicitly described by their authors as customizable. While some authors have highlighted their products’ customizability, including features like the creation of new nodes or blocks and the input of parameters for fine-tuning activities, findings show that such customization, beyond basic parameter adjustments, often demands computing expertise. This requirement can limit accessibility for novices. Among the 29 customizable tools, 10 are block-based and 16 are flow-based, suggesting a potential prevalence of flow-based tools.

Block-based In the category of Java-developed block-based tools, the literature mentions tools such as Scratch [55, 56], along with its implementations including Tooee [57], LevelUp [58], and Interactive Machine Learning Sandbox [59], as well as TinyML (an implementation of ML Blocks) [60]. Within the Python ecosystem for block-based tools, examples include DeepBlocks [61] and GNU Radio Companion [62]. Additionally, the review identifies tools offering customizable components in both Java and Python, such as Rupai (Blockly) [63].

Flow-based Among the seven tools identified as Python-based and flow-based, all offer customization capabilities, highlighting Python’s popularity and its suitability for such applications⁴. Examples are Orange [64] (along with its implementations like Goldenberry [65, 66]), DL-IDE [67], SMILE (Simple Machine Learning) [68], DeepVisual [69], and Graphical AI [70]. In the category of Java and flow-based tools, examples include aFlux [71], Rapsai (Rapid Application Prototyping System for AI) [72, 73], RapidMiner [74], KNIME [75], Yale [76], Node-RED [77], and OneLabeler [78].

When considering tool integrations and implementations as separate entities, the distribution of GUI-equipped tools that are both flow-based and customizable, and written in either Java

⁴IEEE Spectrum: The Top Programming Languages 2023

or Python, appears to be nearly balanced. Moreover, focusing solely on the aspect of GUI and customization — irrespective of the programming language used — the majority emerges as flow-based, with 16 tools, compared to 10 that are block-based. This disparity can partly be due to the fact that not all authors specify the programming languages utilized.

Some cases, such as Marcelle [79], CO-ML [80], and WEKA - Machine Learning workbench [81], exhibit particular ambiguities. For Marcelle and CO-ML, the available documentation falls short of specifying whether these tools are developed using Java, Python, or a mix of both, and it does not categorize them explicitly as block-based or flow-based. While WEKA is identified as Java-based, its documentation lacks clear information on the VPL approach it employs. These omissions may highlight the complexities involved in implementing VPLs within ML design.

Finally, our review identified few tools that play a role in providing methods to clarify the inner workings of black-box models or to elucidate ML mechanisms, for instance mentioning XAI techniques. Among such tools, Gest [82], CO-ML [80], Mix & Match [83], and Rapsai [73].

5.2. Interaction Modality

The interaction modality of these tools can affect their accessibility and usability for users with limited experience. Our analysis indicates that drag-and-drop is the primary interaction modality, enabling users to easily manipulate and connect nodes or blocks in a visual workspace. This modality is typical of block-based and flow-based languages, with our review reporting 27 tools. However, examples like Mix & Match [83] explore alternative approaches. It is a hybrid physical-digital toolkit that integrates GUIs with tangible tokens that users manipulate to design ML-based systems and perform typical ML tasks, such as supervised and unsupervised classification. Another example is Gest [82] - ML gesture recognition system, through which children utilize a sensor to engage with ML concepts.

5.3. Application Domains

Equipping domain experts with the necessary tools to participate in the design process can support the development of unbiased and trustworthy ML-based systems. For instance, leveraging their specialized knowledge can enable increased control over recommender systems, which shape our choices by tailoring search results to our queries, thus influencing our consumption patterns, public opinion, and societal perceptions [6, 7]. Such control can prevent these systems from filtering and prioritizing information based on opaque criteria, like browsing habits and user demographics [8]. Given this premise, the third box in Fig. 2 delves into research question **RQ3**: ‘What are the ML application domains where VPL-based tools find their use?’

This review reports that VPLs are mainly utilized within the field of computer science (13 papers), such as DeepGraph [84]. In the education sector, VPLs are also reflected in 13 papers, with tools like Scratch [55, 56, 57, 58, 59] being employed to introduce children to the concepts underlying ML processes. In industry sector, tools like PaddlePaddle [85] can empower companies to train their employees to become experts proficient in both ML processes and business applications. In healthcare, VPLs provide valuable tools for domain experts, as shown in 5 papers. For instance, KNIME [75] is used to develop an ML-based system aimed at predicting hospital admissions. Similarly, RapidMiner [74] is applied in biomedical informatics for visual

workflow design, thereby enhancing healthcare decisions and facilitating the early diagnosis and prediction of diseases. The Workflow Designer [86] enables users to prototype and manage complex ML workflows, such as those involving electroencephalography signals. Additionally, there are tools for managing ML pipelines in the cloud for specific applications, such as diabetes treatment, using Lemonade [87]. In these cases, widely used ML models, including K-Nearest Neighbors, Naïve Bayes, Decision Trees, Support Vector Machines, and Deep Neural Networks, have been deployed and assessed.

5.4. Accessibility

In this section, we investigate the accessibility of VPL-based tools, as it can affect the participation of a broader audience. Easy accessibility can enhance inclusivity, improve the overall user experience, and increase usability for all users. Tools that allow end-user modifications can be adapted to specific domains, preventing exclusionary experiences. We assess accessibility through two modalities: ease of user modification and mode of access. For the first modality, we evaluate whether the tool is open-source, which enables users to freely inspect, modify, and enhance it, or proprietary, which includes restrictions imposed by the owner. For the second modality, we examine the access method of the application development environment, whether through a web browser or a desktop application. We evaluated both modalities together, recognizing that the ease of user modification represents a deeper form of accessibility. Therefore, we address **RQ4** ‘What access modalities are available for designing ML-based systems?’, by exploring these two access modalities (see Accessibility box in Fig. 2).

Aligned with existing literature on VPLs in the IoT domain [48], we expected a prevalence of open-source web applications. Our review indeed confirmed this expectation, with 28 papers indicating a preference for open-source environments, of which 13 specifically favor web applications such as [60, 63, 79, 72, 52, 70, 87, 84]. This tendency reflects a strategic effort to extend access more broadly and address the accessibility hurdles that proprietary desktop-based platforms (e.g., LabVIEW [88]) present.

In our review, we identified eight papers featuring examples of open-source applications developed specifically for desktop environments, including [75, 64, 65, 66, 69]. Additionally, we found seven applications, such as [59, 58, 55, 56, 57], that are developed for both desktop and web platforms. Finally, we found an application [83] that employs a hybrid model combining TUIs and GUIs. This application incorporates both open-source and proprietary components, and is partially developed for both web and desktop platforms.

5.5. End-users

This section aims at addressing the research question **RQ5**: ‘What is the background of users who have used VPL-based tools in ML application domains?’ VPLs leverage visual representations of programming logic to offer an intuitive approach to software design, making them particularly accessible to users with little to no programming experience. This review reveals their application by domain specialists (17 papers) working in sectors like healthcare and agriculture, as well as students within educational settings. In nine papers, VPLs have been utilized across various proficiency levels, with expectations of more in-depth use by experienced practi-

tioners, such as for ML integration. However, their tool interface can facilitate the prototyping process of ML-based systems by domain experts in healthcare and education. In the computer science domain (10 papers), including computer vision, IoT, and AI engineering, experts have utilized VPL-based tools to mitigate syntax errors and identify areas for improvement in ML processes.

Research provides examples demonstrating that collaboration in co-design activities can effectively engage children in the development of new Intelligent User Interfaces (IUIs) by using modalities such as speech, gesture, and writing. This participation can empower them to conceptualize and propose ideas for complex technical systems that integrate AI processes [89]. This review reports some initiatives aimed at enhancing collaboration among practitioners with diverse levels of expertise, such as Marcelle [79], CO-ML [80], and Rapsai [72]. Similarly, the Mix & Match tool [83] employs a hybrid model combining TUIs and GUIs to foster collaborative design efforts.

5.6. User Experience Evaluation Methods

A key aspect of the study was to examine the extent of user participation in evaluating their interaction with the proposed VPL-based tool. In addressing **RQ6** ‘How have usability and user participation in VPL-based systems been assessed?’, our analysis revealed that 12 studies conducted evaluations on usability and user experience of these systems (see Tab. 2 in Appendix). The other studies focused on computational performance, employing traditional ML evaluation metrics like accuracy, F1 score, and loss. These 12 studies evaluate the usability of VPL-based systems employing a range of methods, such as Likert scales for open-ended questions and custom questionnaires (5 papers), task completion times (6 papers), the think aloud protocol (2 papers), Affinity for Technology Interaction (ATI) Scale and USE Questionnaire (2 papers), and both NASA-TLX and SUS assessments. NASA-TLX and SUS were mainly used in two studies: one assessing the usability and cognitive load of a flow-based system for junior data scientists in comparison to tabular and code-based representations [67], and another evaluating the effectiveness of diverse VPLs for domain experts in healthcare, biomedical laboratories, and education [90]. A study [83] employed the USE Questionnaire (USEQ) to measure user satisfaction concerning usability, satisfaction, and ease of use. The review reported studies employing the Likert scale [59] focusing on the design of a prototype VPL-based system, where participants rated aspects such as interface components, visualization clarity, and system interaction. In certain instances, more tailored evaluation criteria were utilized, such as custom questionnaires [70] exploring users’ experiences with the VPL-based tool through specific questions on their preferences for developing AI/ML graphically and their favored programming languages. Another paper [52] leveraged the cognitive dimensions framework [44] to assess usability at different developmental stages of the VPL-based prototype.

The sample sizes of user studies varied, from 4 to 30 participants, with an average participation of 17 individuals. Two studies each had 30 participants [82, 52], while one study did not specify participant numbers [79] (see Tab. 2). In terms of participant demographics, seven out of the twelve studies disclosed an age range of participants from 10 to 56 years old. However, five of the twelve studies [54, 58, 70, 79, 78] omitted details on participants’ ages, with [70] lacking almost any information regarding its participants. In a study [82], the age of participants ranged

from 10 to 13, as the research aimed at developing methods to teach ML concepts to children. Additionally, six out of the twelve studies detailed the gender distribution among participants, which was not always even; only [83] showed a balanced gender distribution. In the case of [52], gender information was provided for 18 out of 30 participants. Across all studies, out of a total of 122 participants, 44% were female, 46% were male, and 10% were not specified. Overall, the data indicate that the assessment of users' participation with VPL-based tools has been underemphasized, with a greater focus placed on the computational efficiency of the systems deployed and users' experience.

6. Discussion and Conclusions

The integration of AI across various industries primarily relies on ML-based systems, which, despite their transformative potential, can be complex and inaccessible to those without a technical background. This review examines how VPLs can mitigate these barriers, thereby making the design of ML-based systems more accessible to domain specialists. It investigates the application of ML processes through VPLs, aiming to identify tools that democratize AI by addressing both existing challenges and potential areas for future research. Given that leveraging the expertise of domain specialists can enhance trust and trustworthiness in ML decisions, this study investigates the extent to which VPL-based tools integrate interpretability techniques and promote collaborative work environments. Through a systematic examination of 38 articles, selected from an initial pool of 2,363, this review sheds light on the potential of VPLs to contribute to the democratization of AI and enhance its accessibility.

Employed technologies Our findings reveal that ML-based system development primarily employs GUIs based on flow-based programming languages, allowing for user customization. The programming language used, whether Java or Python, alongside the choice of a flow-based design, doesn't inherently limit customization capabilities. However, the focus on customization features suggests that GUIs interfaces can be more easily manipulated by users with computing expertise. The review also reports a limited number of tools that contribute to demystifying the operations of black-box models or explaining ML mechanisms, for example, by incorporating XAI techniques. The accessibility of tools for domain specialists can be influenced by the interaction modality. Our analysis shows that drag-and-drop functionality is the predominant mode of interaction, simplifying complex tasks and enhancing user experience. Despite their user-friendly design for beginners, our review reveals that VPLs are mainly used by computing experts for technological developments, and in education to teach ML concepts to students. Recent research in education is exploring advancements in ML and sensor technology to augment interactive learning experiences. For example, we identified efforts to introduce ML-based gesture recognition systems that utilize physical input devices through the use of sensors [82]. Such systems can enhance the understanding of ML concepts among novices by enabling them to collect data, design ML models, and iteratively refine these models based on feedback. The significant evolution in microprocessors, memory, cameras, and sensors over the past decade has facilitated gestural interaction, signifying a shift toward NUIs [36]. Contemporary literature provides evidence of tools that embody NUI principles directly. For

instance, InteractML [91] that simplifies the development and adjustment of ML models for creators of all backgrounds, using a node-based graph and virtual reality interface, with minimal programming required. Although these ongoing technological advancements are expected to generate a wave of innovative applications in the near future, this review reports poor literature on the integration of NUIs with VPLs.

Application domains, accessibility and evaluation metrics Our study revealed that beyond education and computer science, few domains, such as the healthcare sector, have adopted VPL-based tools (e.g., KNIME and RapidMiner). This finding highlights an opportunity to further explore the capabilities of flow-based programming languages in specific domains. By assessing their limitations and identifying possible enhancements, we can broaden the reach of VPLs to a more diverse audience. The findings indicate a significant trend toward adopting open-source platforms accessible through web applications, consistent with earlier research insights. VPLs can accommodate various expertise levels, simplifying complex tasks for novice users and empowering computing experts. However, they are primarily utilized by novices in educational settings and by experts in computer science. This evidence may explain the lack of initiatives aimed at encouraging collaboration between novices and experts. Finally, the variety in evaluation methods, from tailored custom metrics to broader questionnaires like SUS and NASA TLX, highlights the lack of standardized methodologies for evaluating the usability of VPL-based systems, along with user participation and experience.

In summary, our review of VPL-based tools in the ML context reveals a common problem. A significant number of these tools are not customizable, lack features for interpretability, or require substantial computing expertise for effective application. This finding shows a gap in research towards developing ML-based systems that are readily accessible to domain specialists without deep computing knowledge. By integrating XAI techniques, we could improve understanding of ML decision-making processes. To address this gap, our future work will introduce PyFlowML⁵, a prototype developed within an open-source, flow-based environment tailored for widespread adoption. With a focus on customizability and user-friendliness, we plan to assess whether PyFlowML can streamline ML processes and integrate XAI techniques, thereby improving trust and trustworthiness among novices. Currently being tested by both experts and end-users, we plan to compare its usability with tools like KNIME. This comparison could contribute to set benchmarks for developing VPL-based tools designed to foster the participation of domain specialists in the ML-based systems design.

Limitations This systematic literature review aims to explore how VPL-based tools can engage domain experts in designing trustworthy ML-based systems from a HCI perspective. This study's robustness could be influenced by factors like study selection, drawing primarily from digital libraries like IEEE Xplore and ACM, which house a vast collection of conference papers and journal articles relevant to our focus. However, the coverage of these libraries, while valuable, is not all-encompassing, potentially affecting the comprehensiveness of our findings.

⁵YouTube: PyFlowML Demo

Acknowledgments

Research partly funded by PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - “FAIR - Future Artificial Intelligence Research” - Spoke 1 “Human- centered AI”, funded by the European Commission under the NextGeneration EU programme.

References

- [1] E. Commission, White paper on artificial intelligence - a european approach to excellence and trust, Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52020DC0065&from=EN>, 2020.
- [2] S. Makridakis, The forthcoming artificial intelligence (ai) revolution: Its impact on society and firms, In: *Futures*, Volume 90, Pages 46-60, 2017. doi:<https://doi.org/10.1016/j.futures.2017.03.006>.
- [3] E. Brynjolfsson, A. McAfee, The second machine age: Work, progress, and prosperity in a time of brilliant technologies, Book published by WW Norton & Company, 2014.
- [4] D. Baidoo-Anu, L. Owusu Ansah, Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning, In: *Journal of AI*, Volume 7, Number 1, Pages 52-62, 2023. doi:[10.61969/jai.1337500](https://doi.org/10.61969/jai.1337500).
- [5] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, N. Diakopoulos, Grand challenges for hci researchers, In: *Interactions*, Volume 23, Number 5, Pages 24–25, 2016. doi:[10.1145/2977645](https://doi.org/10.1145/2977645).
- [6] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, R. Burke, Feedback loop and bias amplification in recommender systems, In: *arXiv*, 2020. doi:[10.48550/ARXIV.2007.13019](https://doi.org/10.48550/ARXIV.2007.13019).
- [7] S. Milano, M. Taddeo, L. Floridi, Recommender systems and their ethical challenges, *AI & SOCIETY*, Volume 35, Issue 4, Pages 957–967, 2020. doi:[10.1007/s00146-020-00950-y](https://doi.org/10.1007/s00146-020-00950-y).
- [8] M. Makhortykh, A. Urman, R. Ulloa, Detecting race and gender bias in visual representation of ai on web search engines, In: *Communications in Computer and Information Science*, Pages 36-50, Springer International Publishing, 2021, 2021. doi:[10.1007/978-3-030-78818-6_5](https://doi.org/10.1007/978-3-030-78818-6_5).
- [9] Y. N. Harari, Why technology favors tyranny, In: *The Atlantic*, Volume 322, Number 3, Pages 64-73, 2018.
- [10] N. R. Council, Beyond productivity: Information technology, innovation, and creativity, Book published by National Academies Press, 2003.
- [11] B. Shneiderman, Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered ai systems, In: *ACM Transactions on Interactive Intelligent Systems (TiiS)*, Volume 10, Number 4, Pages 1-31, 2020.
- [12] G. Fischer, End-user development: Empowering stakeholders with artificial intelligence, meta-design, and cultures of participation, In: *Proceedings*, Springer-Verlag, Berlin, Heidelberg, 2021. doi:[10.1007/978-3-030-79840-6_1](https://doi.org/10.1007/978-3-030-79840-6_1).
- [13] H. Lieberman, F. Paternò, M. Klann, V. Wulf, End-user development: An emerging paradigm, In: *End User Development*, Pages 1-8, Springer, 2006.

- [14] G. Fischer, Understanding, fostering, and supporting cultures of participation, In: *Interactions*, Volume 18, Number 3, Pages 42-53, 2011.
- [15] G. Fischer, D. Fogli, A. Mørch, A. Piccinno, S. Valtolina, Design trade-offs in cultures of participation: Empowering end users to improve their quality of life, In: *Behaviour & Information Technology*, Volume 39, Number 1, Pages 1-4, 2020. doi:10.1080/0144929X.2020.1691346.
- [16] B. Shneiderman, *Human-centered ai*, Book published by Oxford University Press, 2022.
- [17] S. Dasgupta, B. M. Hill, Scratch community blocks: Supporting children as data scientists, In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Pages 3620-3631, Denver, Colorado, USA, CHI '17, 2017. doi:10.1145/3025453.3025847.
- [18] F. Paternò, V. Wulf, *New perspectives in end-user development*, Book published by Springer, 2017.
- [19] A. Halfaker, R. S. Geiger, Ores: Lowering barriers with participatory machine learning in wikipedia, In: *Proc. ACM Hum.-Comput. Interact.*, Volume 4, CSCW2, Article 148, Pages 1-37, 2020. doi:10.1145/3415219.
- [20] C. Kelleher, R. Pausch, Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers, *ACM Comput. Surv.*, Vol. 37, No. 2, Article 83, Publication date: June 2005, 2005. doi:10.1145/1089733.1089734.
- [21] J. C. Brancheau, J. C. Wetherbe, Key issues in information systems management, *MIS quarterly*, pp. 23-45, 1987.
- [22] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, S. Wiedenbeck, The state of the art in end-user software engineering, *ACM Comput. Surv.*, Vol. 43, No. 3, Article 21, Publication date: April 2011, 2011. doi:10.1145/1922649.1922658.
- [23] P. H. Winston, *Artificial intelligence*, Addison-Wesley Longman Publishing Co., Inc., 1984.
- [24] F. Paternò, End user development: Survey of an emerging field for empowering people, In: *International Scholarly Research Notices*, Volume 2013, Hindawi, 2013.
- [25] M. A. Kuhail, S. Farooq, R. Hammad, M. Bahja, Characterizing visual programming approaches for end-user developers: A systematic review, In: *IEEE Access*, Volume 9, Pages 14181-14202, 2021. doi:10.1109/ACCESS.2021.3051043.
- [26] B. A. Myers, Taxonomies of visual programming and program visualization, In: *Journal of Visual Languages and Computing*, Volume 1, Number 1, Pages 97-123, March 1990, 1990.
- [27] M. M. Burnett, M. J. Baker, A classification system for visual programming languages, In: *Journal of Visual Languages and Computing*, Volume 5, Number 3, Pages 287-300, September 1994, 1994.
- [28] T. Turchi, A. Malizia, Fostering computational thinking skills with a tangible blocks programming environment, In: *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Pages 232-233, 2016. doi:10.1109/VLHCC.2016.7739692.
- [29] C. Ardito, M. F. Costabile, G. Desolda, R. Lanzilotti, M. Matera, A. Piccinno, M. Picozzi, User-driven visual composition of service-based interactive spaces, In: *Journal of Visual Languages & Computing*, Volume 25, Number 4, Pages 278-296, 2014, 2014. doi:https://doi.org/10.1016/j.jvlc.2014.01.003.
- [30] M. M. Burnett, A. L. Ambler, Interactive visual data abstraction in a declarative visual

- programming language, In: Journal of Visual Languages & Computing, Volume 5, Number 1, Pages 29-60, 1994, 1994. doi:<https://doi.org/10.1006/jvlc.1994.1003>.
- [31] D. D. Hils, Visual languages and computing survey: Data flow visual programming languages, In: Journal of Visual Languages & Computing, Volume 3, Number 1, Pages 69-101, 1992, 1992. doi:[https://doi.org/10.1016/1045-926X\(92\)90034-J](https://doi.org/10.1016/1045-926X(92)90034-J).
 - [32] K. N. Whitley, L. R. Novick, D. Fisher, Evidence in favor of visual representation for the dataflow paradigm: An experiment testing labview's comprehensibility, International Journal of Human-Computer Studies, Vol. 64, No. 4, pp. 281-303, 2006.
 - [33] B. McNeel, S. Davidson, Grasshopper, Online resource available at <http://www.grasshopper3d.com/>, 2023. URL: <http://www.grasshopper3d.com/>.
 - [34] M. M. Burnett, D. W. McIntyre, Visual programming, In: Computer-Los Alamitos, Volume 28, Pages 14-14, IEEE Institute of Electrical and Electronics, 1995, 1995.
 - [35] T. Turchi, A. Malizia, A human-centred tangible approach to learning computational thinking, EAI Endorsed Transactions on Ambient Systems, Vol. 3, No. 9, 2016.
 - [36] D. A. Norman, Natural user interfaces are not natural, Interactions, Vol. 17, No. 3, pp. 6-10, 2010.
 - [37] M. Turek, Explainable ai (xai), DARPA, 2018. URL: <https://www.darpa.mil/program/explainable-artificial-intelligence>.
 - [38] M. Ibrahim, M. Louie, C. Modarres, J. Paisley, Global explanations of neural networks: Mapping the landscape of predictions, In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES '19), 2019.
 - [39] A. Das, P. Rad, Opportunities and challenges in explainable artificial intelligence (xai): A survey, arXiv preprint arXiv:2006.11371, 2020.
 - [40] D. Slack, S. Hilgard, E. Jia, S. Singh, H. Lakkaraju, Fooling lime and shap: Adversarial attacks on post hoc explanation methods, In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), 2020. URL: <https://doi.org/10.1145/3375627.3375830>. doi:10.1145/3375627.3375830.
 - [41] D. D. Hils, Visual languages and computing survey: Data flow visual programming languages, Journal of Visual Languages & Computing, Vol. 3, No. 1, pp. 69-101, 1992. doi:10.1016/1045-926X(92)90034-J.
 - [42] M. Boshernitsan, M. S. Downes, Visual programming languages: A survey, Computer Science Division, University of California, Los Angeles, CA, USA, 2004.
 - [43] J. M. Rouly, J. D. Orbeck, E. Syriani, Usability and suitability survey of features in visual aids for non-programmers, Proceedings of the 6th Workshop on Evaluation and Usability of Programming Languages and Tools, PLATEAU '14, pp. 31-42, 2014. doi:10.1145/2688204.2688207, portland, Oregon, USA.
 - [44] T. R. G. Green, M. Petre, Usability analysis of visual programming environments: A 'cognitive dimensions' framework, In: Journal of Visual Languages & Computing, Volume 7, Number 2, Pages 131-174, 1996. doi:<https://doi.org/10.1006/jvlc.1996.0009>.
 - [45] D. Mason, K. Dave, Block-based versus flow-based programming for naive programmers, In: 2017 IEEE Blocks and Beyond Workshop, Pages 25-28, 2017. doi:10.1109/BLOCKS.2017.8120405.
 - [46] M. Noone, A. Mooney, Visual and textual programming languages: A systematic review of the literature, Journal of Computers in Education, Vol. 5, pp. 149-174, 2018.

- [47] D. Saito, A. Sasaki, H. Washizaki, Y. Fukazawa, Y. Muto, Program learning for beginners: Survey and taxonomy of programming learning tools, Presented at the 2017 IEEE 9th International Conference on Engineering Education (ICEED), pp. 137-142, 2017. doi:10.1109/ICEED.2017.8251181.
- [48] P. P. Ray, A survey on visual programming languages in internet of things, Scientific Programming, Vol. 2017, 2017.
- [49] M. A. Kuhail, S. Farooq, R. Hammad, M. Bahja, Characterizing visual programming approaches for end-user developers: A systematic review, IEEE Access, Vol. 9, pp. 14181–14202, 2021.
- [50] H. E. Kamouchi, M. Kissi, O. E. Beggar, Low-code/no-code development: A systematic literature review, Presented at the 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–8, 2023.
- [51] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, 2007. Issue 2, January 2007.
- [52] Y. Cheng, J. Chen, Q. Huang, Z. Xing, X. Xu, Q. Lu, Prompt sapper: a llm-empowered production tool for building ai chains, ACM Transactions on Software Engineering and Methodology, 2023.
- [53] A. Mahanti, R. Alhaji, Visual interface for online watching of frequent itemset generation in apriori and eclat, Fourth International Conference on Machine Learning and Applications (ICMLA'05), pp. 6–pp, 2005.
- [54] A. Rao, A. Bihani, M. Nair, Milo: A visual programming environment for data science education, 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 211–215, 2018.
- [55] W. Shi, Z. Dong, L. Zhang, Graphical platform of intelligent algorithm development for object detection of educational drone, 2021 China Automation Congress (CAC), pp. 6780–6784, 2021.
- [56] P. Plaza, M. Castro, J. M. Sáez-López, E. Sancristobal, R. Gil, A. Menacho, F. García-Loro, B. Quintana, S. Martin, M. B. et al., Promoting computational thinking through visual block programming tools, 2021 IEEE Global Engineering Education Conference (EDUCON), pp. 1131–1136, 2021.
- [57] Y. Park, Y. Shin, Tooee: A novel scratch extension for k-12 big data and artificial intelligence education using text-based visual blocks, IEEE Access, Vol. 9, pp. 149630–149646, 2021.
- [58] T. Reddy, R. Williams, C. Breazeal, Levelup–automatic assessment of block-based machine learning projects for ai education, 2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1–8, 2022.
- [59] G. Nodalo, J. M. S. III, J. Valenzuela, J. A. Deja, On building design guidelines for an interactive machine learning sandbox application, Proceedings of the 5th International ACM In-Cooperation HCI and UX Conference, pp. 70–77, 2019.
- [60] R. Williams, M. Moskal, P. D. Halleux, MI blocks: A block-based, graphical user interface for creating tinyml models, 2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1–5, 2022.
- [61] T. Calò, L. D. Russis, Towards a visual programming tool to create deep learning models, Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 38–44, 2023.

- [62] R. Anil, R. Danyamol, H. Gawande, R. Gandhiraj, Machine learning plug-ins for gnu radio companion, 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), pp. 1–5, 2014.
- [63] M. H. Masum, T. S. Rifat, S. M. Tareeq, H. Heickal, A framework for developing graphically programmable low-cost robotics kit for classroom education, Proceedings of the 10th International Conference on Education Technology and Computers, pp. 22–26, 2018.
- [64] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevár, M. Milutinović, M. Možina, M. Polajnar, M. Toplak, A. S. et al., Orange: data mining toolbox in python, Journal of Machine Learning Research, Vol. 14, No. 1, pp. 2349–2353, 2013.
- [65] S. Rojas-Galeano, N. Rodriguez, Goldenberry: Eda visual programming in orange, Proceedings of the 15th annual conference companion on Genetic and evolutionary computation, pp. 1325–1332, 2013.
- [66] L. P. Garzón-Rodríguez, H. A. Diosa, S. Rojas-Galeano, Deconstructing gas into visual software components, Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 1125–1132, 2015.
- [67] S. G. Tamilselvam, N. Panwar, S. Khare, R. Aralikkatte, A. Sankaran, S. Mani, A visual programming paradigm for abstract deep learning model development, Proceedings of the 10th Indian Conference on Human-Computer Interaction, pp. 1–11, 2019.
- [68] I. Khodnenko, S. V. Ivanov, A. Lantseva, A lightweight visual programming tool for machine learning and data manipulation, 2020 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 981–985, 2020.
- [69] C. Xie, H. Qi, L. Ma, J. Zhao, Deepvisual: a visual programming tool for deep learning systems, 2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC), pp. 130–134, 2019.
- [70] A. Shen, Y. Sun, Graphicalai: A user-centric approach to develop artificial intelligence and machine learning applications using a visual and graphical language, 2021 4th International Conference on Data Storage and Data Engineering, pp. 52–58, 2021.
- [71] T. Mahapatra, I. Gerostathopoulos, C. Prehofer, S. G. Gore, Graphical spark programming in iot mashup tools, 2018 fifth international conference on Internet of Things: systems, management and security, pp. 163–170, 2018.
- [72] R. Du, N. Li, J. Jin, M. Carney, X. Yuan, R. Iyengar, P. Yu, A. Kowdle, A. Olwal, Experiencing rapid prototyping of machine learning based multimedia applications in rapsai, Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–4, 2023.
- [73] R. Du, N. Li, J. Jin, M. Carney, S. Miles, M. Kleiner, X. Yuan, Y. Zhang, A. Kulkarni, X. L. et al., Rapsai: Accelerating machine learning prototyping of multimedia applications through visual programming, Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pp. 1–23, 2023.
- [74] M. Bjaoui, H. Sakly, M. Said, N. Kraiem, M. S. Bouhlel, Depth insight for data scientist with rapidminer «an innovative tool for ai and big data towards medical applications», Proceedings of the 2nd International Conference on Digital Tools & Uses Congress, pp. 1–6, 2020.
- [75] R. Tsoni, V. Kaldis, I. Kapogianni, A. Sakagianni, G. Feretzakis, V. S. Verykios, A machine learning pipeline using knime to predict hospital admission in the mimic-iv database, 2023

- 14th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1–6, 2023.
- [76] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler, Yale: Rapid prototyping for complex data mining tasks, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 935–940, 2006.
 - [77] R. Machhamer, J. Altenhofer, K. Ueding, L. Czenkusch, F. Stolz, M. Harth, M. Mattern, A. Latif, S. Haab, J. H. et al., Visual programmed iot beehive monitoring for decision aid by machine learning based anomaly detection, *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–5, 2020.
 - [78] Y. Zhang, Y. Wang, H. Zhang, B. Zhu, S. Chen, D. Zhang, Onelabeler: A flexible system for building data labeling tools, *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2022.
 - [79] J. Françoise, B. Caramiaux, T. Sanchez, Marcelle: composing interactive machine learning workflows and interfaces, *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 39–53, 2021.
 - [80] T. Tseng, J. K. Chen, M. Abdelrahman, M. B. Kery, F. Hohman, A. Hilliard, R. B. Shapiro, Collaborative machine learning model building with families using co-ml, *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, pp. 40–51, 2023.
 - [81] G. Holmes, A. Donkin, I. H. Witten, Weka: A machine learning workbench, *Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference*, pp. 357–361, 1994.
 - [82] T. Hitron, Y. Orlev, I. Wald, A. Shamir, H. Erel, O. Zuckerman, Can children understand machine learning concepts? the effect of uncovering black boxes, *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–11, 2019.
 - [83] A. Jansen, S. Colombo, Mix & match machine learning: An ideation toolkit to design machine learning-enabled solutions, *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 1–18, 2023.
 - [84] Q. Hu, L. Ma, J. Zhao, Deepgraph: A pycharm tool for visualizing and understanding deep learning models, *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 628–632, 2018. doi:10.1109/APSEC.2018.00079.
 - [85] R. Bi, T. Xu, M. Xu, E. Chen, Paddlepaddle: A production-oriented deep learning platform facilitating the competency of enterprises, *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp. 92–99, 2022. doi:10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00046.
 - [86] P. Ježek, L. Vařeka, Workflow designer - a web application for visually designing eeg signal processing pipelines, *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 368–373, 2019. doi:10.1109/BIBE.2019.00072.
 - [87] W. dos Santos, L. F. M. Carvalho, G. de P. Avelar, A. Silva, L. M. Ponce, D. Guedes, W. Meira, Lemonade: A scalable and efficient spark-based platform for data analytics, *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 745–748, 2017. doi:10.1109/CCGRID.2017.142.
 - [88] D. Kaya, M. Türk, Comparing the performance of the kernel functions in the lda-svm

based classification algorithm in the labview environment, 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), pp. 1-4, 2018. doi:10.1109/IDAP.2018.8620788.

- [89] J. Woodward, Z. McFadden, N. Shiver, A. Ben-hayon, J. C. Yip, L. Anthony, Using co-design to examine how children conceptualize intelligent interfaces, Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–14, 2018. doi:10.1145/3173574.3174149, cHI '18, Montreal QC, Canada.
- [90] C. Schütze, A. Groß, B. Wrede, B. Richter, Enabling non-technical domain experts to create robot-assisted therapeutic scenarios via visual programming, Companion Publication of the 2022 International Conference on Multimodal Interaction, pp. 166–170, 2022.
- [91] C. Hilton, N. Plant, C. G. Díaz, P. Perry, R. Gibson, B. Martelli, M. Zbyszynski, R. Fiebrink, M. Gillies, Interactml: Making machine learning accessible for creative practitioners working with movement interaction in immersive media, Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology, Art. No. 23, n.pag., 2021. doi:10.1145/3489849.3489879, vRST '21, Osaka, Japan.

A. User-based testing details

Table 2: User-based testing details

Paper	Procedure (N. Tasks)	Evaluation Methods	N. Users	Users Age	Users Type	User Proficiency	Users Gender
[54]	Predefined task (1)	Open questions	20	n/a	university students	inexpert	n/a
[58]	Predefined task (2)	Likert scale, custom questionnaire, Task completion time	25	n/a	university students	inexpert	16 female, 7 males
[67]	Predefined task (3)	Task completion time, SUS, NASA TLX, open questions	18	19-24 years old	university students	expert and inexpert	7 females, 11 males
[82]	Predefined task (3)	Open questions	30	10-13 years old	children	inexpert	10 females, 20 males
[59]	Predefined task (1)	Likert scale, open questions	10	19-25 years old	university students and professionals	expert and inexpert	n/a
[70]	Predefined task (1)	Open questions	4	n/a	n/a	n/a	n/a
[79]	Predefined task (2)	Custom questionnaire, Think-aloud protocol	n/a	n/a	university students and professionals	expert and inexpert	n/a
[90]	Predefined task (1)	ATI Scale, SUS, NASA TLX, Think-aloud protocol	9	26-54 years old (mean = 41)	professionals	inexpert	7 females, 2 males
[78]	Predefined task (1)	Task completion time, Open questions	8	n/a	professionals	expert	n/a
[83]	Predefined task (2)	Likert scale, open questions, USEQ, Task completion time	12	18-34 years old	university students	inexpert	6 females, 6 males
[52]	Predefined task (4)	Likert scale, Cognitive Dimensions, Task completion time	30	18-25 years old	university students	expert and inexpert	8 females, 10 males, 12 not specified
[73]	Predefined task (2)	Likert scale, open questions, custom questionnaire, Task completion time	22	26-56 years old	professionals	expert	n/a