Journal of Information Systems Education

Volume 36
Issue 4
Fall 2025

Teaching Tip

Toward Open-Source Cloud-Based Visual Machine Learning Platform: A Human-Interface Usability Study

AbdElRahman ElSaid, Yao Shi, Mohamed Wiem Mkaour, Mahmoud Altalouli, and Anas Tawfik

Recommended Citation: ElSaid, A., Shi, Y., Mkaour, M. W., Altalouli, M., & Tawfik, A. (2025). Teaching Tip: Toward Open-Source Cloud-Based Visual Machine Learning Platform: A Human-Interface Usability Study. *Journal of Information Systems Education*, 36(4), 342-351. https://doi.org/10.62273/QTWE9132

Article Link: https://jise.org/Volume36/n4/JISE2025v36n4pp342-351.html

Received:	February 11, 2025
First Decision:	May 11, 2025
Accepted:	August 1, 2025
Published:	December 15, 2025

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 36(4), 342-351, Fall 2025
https://doi.org/10.62273/QTWE9132

Teaching Tip
Toward Open-Source Cloud-Based Visual Machine Learning Platform: A Human-Interface Usability Study

AbdElRahman ElSaid
College of Science & Engineering
University of North Carolina Wilmington
Wilmington, NC 28403, USA
elsaida@uncw.edu

Yao Shi
Cameron School of Business
University of North Carolina Wilmington
Wilmington, NC 28403, USA
shiy@uncw.edu

Mohamed Wiem Mkaour
College of Innovation & Technology
University of Michigan-Flint
Flint, MI 48109, USA
mmkaouer@umich.edu

Mahmoud Altalouli
Department of Education, Languages, and Instructional Design
State University of New York Brockport
Brockport, NY 14420, USA
maltalouli@brockport.edu

Anas Tawfik
Booking.com
Amsterdam, Netherlands
anas.tawfik@booking.com

ABSTRACT

The growing integration of artificial intelligence (AI) into everyday life necessitates a transformation in machine learning (ML) education and development practices, empowering end users with domain knowledge to independently design, train, test, and deploy specialized ML models. However, the technical complexity of ML, particularly in areas such as neural networks, presents a significant barrier for those users. To overcome this challenge, it is essential to reduce the cognitive burden associated with coding, algorithm configuration, and system setup. This study introduces an early-stage prototype of an open-source, cloud-based visual ML platform aimed at lowering this barrier. The platform enables users to configure, execute, and monitor ML workflows through an intuitive graphical interface, eliminating the need for programming skills or environment setup. To evaluate the platform's usability and user-friendliness, a user study was conducted involving participants from diverse academic backgrounds. Participants engaged with both visual and command-line versions of the system and completed a structured questionnaire. The results revealed a strong preference for the visual interface, especially among users with limited technical experience. These findings suggest that intuitive, no-code platforms can significantly reduce entry barriers and foster broader engagement with ML in educational settings.

Keywords: Machine learning, Visual inquiry tool, Cloud platform, User friendliness

1. INTRODUCTION

Machine Learning (ML) has emerged as an essential pillar of today's digital landscape, exerting a profound influence across numerous fields. Traditionally, the development and application of ML models have been in the purview of computer science (Mason, 2013). Computer scientists are technically proficient but usually lack knowledge of the intricacies of the specific areas to which models are applied (Berthold, 2019; Luna-Reyes, 2018; Schreck & Veeramachaneni, 2016). Therefore, collaborative work has always existed between computer scientists and people in respective domains, such as information systems and data science. However, the wide-spread adoption of ML across different fields underscores the need for a paradigm that places domain practitioners at the core when it comes to designing, training, testing, and deploying a customized ML model (Sundberg & Holmström, 2024; Yang et al., 2018). This paradigm shift presents a significant challenge, particularly in the context of ML education (Topi, 2019; Wang & Wang, 2021). Hence, there is a need for an abstraction or virtualization layer that conceals ML complexities from users, enabling broader participation and fostering innovation across a more diverse population.

Recognizing the convergence of domain knowledge and ML, we proposed a cloud-based virtual platform to enhance ML education. The platform offers a computational environment beyond conventional implementations to solve the issue of bridging between domains and ML methodologies. To this end, the platform abstracts away many of the technical complexities, eliminating the need for programming. Instead, users access powerful ML capabilities through a fully visual interface cloud-based interface. This interface is oriented toward the learner, working to eliminate the mystery behind model creation and tuning processes. The platform serves as a comprehensive, cloud-based educational ecosystem, enabling intuitive ML pipeline development alongside networked and scalable learning experiences.

This manuscript presents the initial phase of a project aimed at developing an open-source, fully visual, cloud-based machine learning platform specifically designed for students and users who lack a background in computer science. The platform enables users to configure, train, and deploy ML models through an intuitive graphical interface without the need for programming skills or manual system configuration. At this stage, the study focuses on evaluating the platform's usability and user-friendliness, ensuring that individuals with limited technical expertise can effectively engage with ML workflows. A user study involving participants from diverse academic disciplines was conducted to compare their experiences using both the visual prototype and a traditional command-line interface. This foundational work contributes to the broader educational objective of democratizing access to ML, aligning with recent calls in the literature to lower the barrier to AI and ML education across disciplines (Altalouli & Curry, 2023).

2. THE CLOUD-BASED MACHINE LEARNING PLATFORM

Training well-structured, handcrafted models on representative data samples is essential for developing generalizable ML models. Unoptimized architectures often contain redundant components that add noise instead of improving predictive accuracy. Therefore, optimizing model structures for specific tasks is crucial to achieving better performance and more reliable outcomes.

Neural Architecture Search (NAS) offers a promising solution through a range of metaheuristic algorithms (Darwish et al., 2020; Elsken et al., 2017, 2019) designed to automate the engineering of ML model architectures. Figure 1 illustrates a typical sampling-based NAS process (ElSaid, 2020; ElSaid et al., 2019; ElSaid et al., 2021; ElSaid et al., 2020), in contrast to evolution-based NAS approaches (Ororbia et al., 2019), where the architecture iteratively refines itself by eliminating non-contributory structural elements based on performance metrics. The optimization begins with an initial structure, followed by the selective removal of components, with each iteration evaluated and logged. This process continues, leveraging historical performance data to guide the generation of improved neural structures, ultimately converging toward a near-optimal solution within a feasible timeframe. For a more interactive understanding, the frames in Figure 1 are animated on the experiment's instructional webpage (https://examm.onrender.com[1]), visually demonstrating the progression from the initial model to the best-performing configuration. (Note that the provided URL may take a few minutes to load on the first visit, as it is hosted on a free platform with limited server resources.)

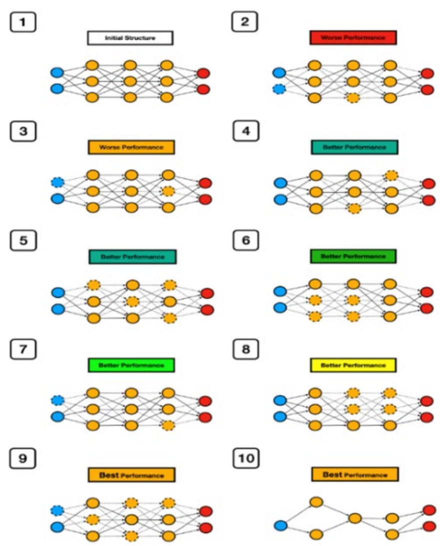


Figure 1. Process of Architecture Building

NAS is a relatively new but rapidly growing area in ML, often described as "the likely next deep learning" (Miikkulainen, 2020). However, leveraging its full potential requires significant technical knowledge and skill. Currently, there is a lack of integrated ML development platforms that make NAS accessible, particularly for users with domain knowledge but limited technical backgrounds. While major ML platforms offer some NAS capabilities, they each have notable limitations. Google AutoML includes NAS but lacks transparency in model architecture and tuning process. Microsoft Azure ML Studio provides a visual interface but offers limited flexibility for model customization. IBM Watson

Studio supports hyperparameter optimization, yet NAS functionality is not explicitly exposed or easily accessible (Roy et al., 2019).

This study represents the first step toward building a platform that makes NAS accessible: an open-source, cloud-based ML cyberinfrastructure that empowers end users to design, build, and deploy their own ML models using their own data. Although NAS techniques have traditionally required advanced expertise, integrating NAS into visual tools—as demonstrated in this platform—can empower beginners to create and optimize sophisticated neural networks without writing any code. By abstracting technical complexity, NAS becomes an exploratory and educational tool that fosters conceptual understanding through interaction. Unlike conventional ML approaches that rely on predefined architectures and manual hyperparameter tuning, NAS optimizes model structures automatically, making the process more accessible and forgiving for novices. This supports experimentation and accelerates learning, allowing users to observe how changes in structure influence model performance. Moreover, it introduces learners to the frontiers of modern ML—offering them not just experience with neural networks, but also with advanced optimization techniques that underpin much of state-of-the-art AI today.

In this phase of our exploration, we focus on the design and evaluation of an intuitive, user-centered interface tailored to the needs of primary stakeholders—researchers, scholars, and learners with limited coding experience. Figure 2 presents a high-level schematic of the proposed system, which integrates components for data uploading and streaming, storage, processing, computing resource allocation, ML model design, optimization, management, and data output reporting and visualization.

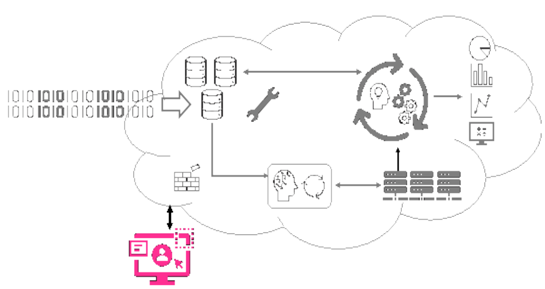


Figure 2. Infrastructure of the Open-Source Cloud-Based ML

The interface—highlighted in magenta in the Figure 2—serves as the focal point of this study. It is deliberately designed to abstract away computational complexities, enabling users to engage with machine learning workflows without the burden of technical overhead. By emphasizing usability and user-friendliness, this study aims to validate the effectiveness of the platform prototype in supporting non-technical users through an accessible and intuitive interface.

At this stage, the primary objective is to assess how easily users can interact with the platform, rather than to evaluate their understanding of complex ML concepts. This approach is essential for ensuring that the platform lowers the entry barrier for non-expert users, laying the groundwork for future research that will examine its impact on deeper ML learning and conceptual mastery. As the platform matures, subsequent phases of the research will include more structured evaluations of learning outcomes, enabling a comprehensive assessment of its educational effectiveness.

3. METHODOLOGY

We conducted an experiment to assess a prototype system's ability to support users with varying levels of expertise in ML and NAS. Participants were presented with two versions of an ML training, testing, and neural structure optimization platform, and their feedback was collected to assess user experience.

3.1 Versions of Machine Learning Systems

3.1.1 Command-Line Version. The first version is a command-line system, where users are required to download a software package, compile it, and initiate a job using predefined command-line options. The command-line version was implemented in C++ for Linux-based systems and required multiple dependencies, including OpenMPI (Chandra et al., 2001) to support parallel optimization (Ororbia et al., 2019; ElSaid, 2020; ElSaid et al., 2021; ElSaid et al., 2020). While this version is efficient, its reliance on complex setup procedures and external libraries presents significant usability challenges, particularly for non-technical users.

3.1.2 Visual Cloud-Based Version. The second version is a prototype of a visual, cloud-based platform designed to simplify user interaction with ML and NAS workflows. In this version, users interact with a graphical front-end that translates their inputs into command-line arguments, which are then executed on the backend. This architecture enables users to initiate and monitor ML jobs without engaging directly with the command line.

Figure 3 displays a screenshot of the editor page of the prototype, which is publicly accessible at https://examm.onrender.com/editor. This visual prototype offers a seamless front-end to back-end workflow. The frontend is built with React, TypeScript, Ant Design, and eCharts, providing an interactive and responsive user experience. It includes a visual neural network chart and an editor where users can set command-line parameters. These parameters are validated, formatted, and submitted to the backend, which then triggers the command-line execution. Real-time output is streamed back to the interface, allowing users to monitor execution, search logs, and share session IDs for collaborative work. The backend, developed using Node.js, TypeScript, and the TSed framework, manages real-time communication via Socket.io, thereby enabling live streaming of command-line output to the user interface. Instead of relying on a traditional database, the system uses caching mechanisms to dynamically store command-line output.

The cloud-based version features a graphical user interface (GUI) that allowed users to configure NAS parameters, including (1) database, training, and testing data files, (2) input and output parameters ornodes, (3) the number of hidden layers and nodes per layer, (4) time-lag and offset settings, and (5) the chosen NAS method (EXAMM or ANTS) and its related parameters. Although the two NAS methods use different strategies—ANTS begins with a fully connected superstructure and prunes redundant connections (ElSaid et al., 2020), while EXAMM starts with only input and output nodes and evolves the structure incrementally (Ororbia et al., 2019)—the GUI abstracts these technical differences, providing a unified and intuitive configuration experience.

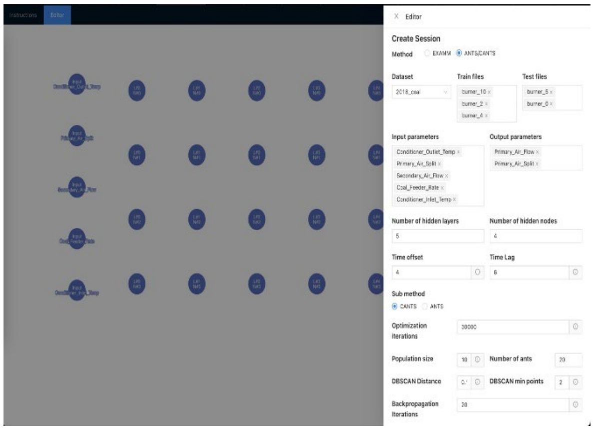


Figure 3. Editor Page of the Prototype

By automating backend processes, the system allows users to concentrate on configuring parameters and interpreting results, thereby making advanced NAS tools more accessible to a broader audience. Moreover, the platform integrates both manual configuration options and automated NAS optimization, empowering users to compare their initial model designs with the final architectures generated by EXAMM and ANTS. This dual approach ensures a balance between user flexibility and ease of use, supporting both exploratory experimentation and guided optimization within a unified interface.

3.2 Experiment Design

3.2.1 Experiment Setup. The experiment was conducted online through a publicly accessible instructional website (https://examm.onrender.com), which provided participants with comprehensive resources for engaging with the prototype system. This online experiment setup offered flexibility, accommodating participants' varying schedules and technical environments—an important consideration for educational studies involving geographically dispersed learners. The website provided the following key resources.

3.2.1.1 Background on NAS Methods. An overview of EXAMM and ANTS, the two core NAS methods used in the prototype. This section explained their underlying mechanisms and typical applications, providing participants with essential context for completing the experimental tasks.

3.2.1.2 Video Walkthrough. A detailed, step-by-step video guide demonstrating how to install, configure, and use both the command-line and visual cloud-based versions of the platform. This resource was designed to reduce cognitive load for non-technical users and support self-guided learning.

3.2.1.3 Setup Instructions. Comprehensive documentation covering the installation process for the command-line version, troubleshooting common dependency issues, and navigating the visual interface.

The instructional materials were designed to familiarize participants with the prototype's interface and core functionalities, prioritizing usability over technical complexity. Since the primary objective of the study is to evaluate the platform's accessibility and user-friendliness, the content emphasized practical interaction rather than the underlying mechanics of the NAS methods (e.g., EXAMM and ANTS) integrated into the system. Future iterations of the platform will refine this instructional strategy based on participant feedback.

3.2.2 Experiment Process. Participants were asked to complete a series of tasks using both the command-line and visual cloud-based versions of the ML platform. These tasks were designed to evaluate technical proficiency, usability, and overall user experience.

For the command-line version, participants were instructed to download, compile, and execute the command-line version of the platform. They manually configured NAS parameters and initiated training and testing jobs using predefined command-line options. This task was intended to highlight the technical challenges often encountered by users without programming experience and served as a baseline for comparison with the visual interface.

For the visual cloud-based version, participants performed equivalent ML tasks using the graphical user interface of the prototype. This version required no coding and allowed users to configure and run NAS workflows through an intuitive, web-based interface. The goal of this task was to assess the accessibility, ease of use, and educational potential of the visual system, particularly in lowering the barrier to entry for learners new to ML.

3.2.3 Survey. Following the experiment process, participants were asked to complete a survey. Given the exploratory nature of the research, the primary objective was to collect practical user feedback rather than to conduct a psychometric or statistical analysis. The questions were informed by domain knowledge, user experience principles, and insights from prior studies on ML usability. The questionnaire, presented in Table 1, includes the following components.

3.2.3.1 Self-Assessment of Technical Skills. Question 1 asks participants to rate their confidence in using Linux command-line tools. This measure provides context for interpreting their ease-of-use ratings.

3.2.3.2 Comparative Usability Assessment. Questions 2 through 5 ask participants to evaluate the convenience and accessibility of the visual platform in comparison to the command-line version. These items were designed to capture user perceptions of ease of use, interface intuitiveness, and overall user experience across both platform versions.

3.2.3.3 Qualitative Feedback. Question 6 consists of open-ended prompts inviting participants to share their experiences, challenges, and suggestions for improvement. These responses can provide deeper insights into the prototype's usability, highlighting user perspectives that may not be captured through quantitative measures alone.

No.	Questions
1	I can use Linux command-line confidently (1: least confident, 5: most confident)
2	I tried (check all that apply): • EXAMM on visual cloud-based platform • ANTS on visual cloud-based platform • EXAMM on command-line platform • ANTS on command-line platform
3	How convenient is the visual ML cloud-based platform? • Very inconvenient • Inconvenient • Normal • Convenient • Very convenient
4	Compared to the command-line platform: How convenient the visual ML cloud- based platform? • I could not use the command-line platform • Much easier to use • Somehow easier to use • Same level of convenience • More complicated • Much harder to use
5	If could not use the command-line platform: Why could not use the command-line platform? • N/A • I do not use Linux (or Linux based operating system) • The installation instructions were not clear • The instructions to use the platform were not clear • Missing libraries and installation errors • Too much work • Other
6	Please provide any suggestions about either platform (visual cloud-based and/or command-line)

Table 1. Survey Questionnaire

3.2.4 Participant Recruitment and Demographics. To evaluate the usability and user-friendliness, participants were recruited from a range of academic backgrounds. This inclusive approach aligns with the overarching goal of ML tools accessible to non-technical learners—a critical consideration for educators seeking to integrate ML concepts into their curricula.

The study involved 247 volunteer participants, representing a mix of faculty members, graduate students, and undergraduate students from a wide range of disciplines—including computer science, business, engineering, humanities, and social sciences. This broad representation was intentionally sought to capture the experiences of users with varying levels of technical proficiency, from experienced programmers to individuals with deep subject knowledge but no formal training in computing. Participants were recruited from universities across the East Coast, Midwest, and Caribbean regions (i.e., Puerto Rico and the U.S. Virgin Islands) of the United States, ensuring both geographic and academic diversity. This diversity provided insights into the usability of the prototype across different educational contexts, supporting the broader objective of developing a ML tool that is both accessible and effective for a wide spectrum of learners.

4. RESULTS

The experiment involved 247 participants from different backgrounds, including both individuals with and without computing experience. A significant portion (48.2%) reported above-average confidence in using the Linux operating system (see Figure 4). This suggests that operating the command-line version of the system via a Linux terminal was not a major barrier for most participants. This confidence was reflected in usage patterns: 43.7% of participants attempted the command-line version of EXAMM, while 37.7% tried the command-line version of ANTS. However, the majority preferred the visual interface, with 81% and 83% of participants using the visual versions of EXAMM and ANTS, respectively (see Figure 5).

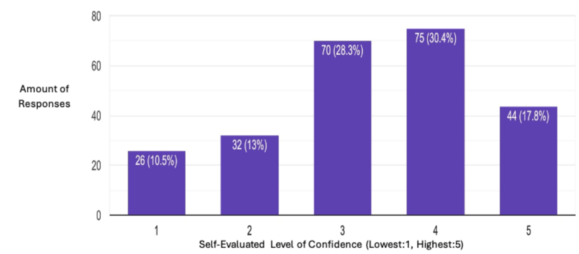


Figure 4. Participant Confidence Levels in Using Linux

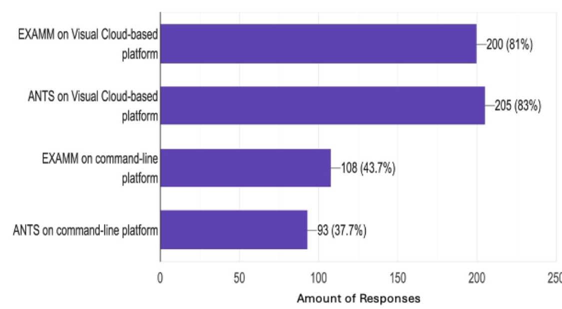


Figure 5. Comparison of Visual and Command-Line Usage for EXAMM and ANTS

Among the participants who used the visual versions of EXAMM and ANTS, 18.2% found them very convenient, 52.6% found them convenient, 20.6% rated them as neutral, 3.6% found them inconvenient, and 4.9% considered them very inconvenient (see Figure 6).

When asked to compare the visual interface with the command-line version (see Figure 7), 33.6% of participants indicated they were unable to use the command-line system and therefore could not make a comparison. Among those who could, 49% reported that the visual system was easier to use—30.4% said it was much easier, and 18.6% said it was somewhat easier. Only 13% felt that both systems offered a similar level of convenience. Meanwhile, about 4% found the visual system more complicated, and just 0.4% reported that it was much harder to use compared to the command-line version.

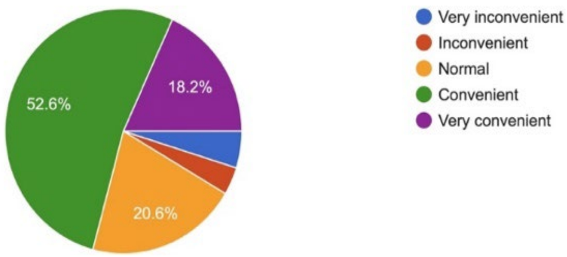


Figure 6. Participant Perceptions of Prototype Convenience

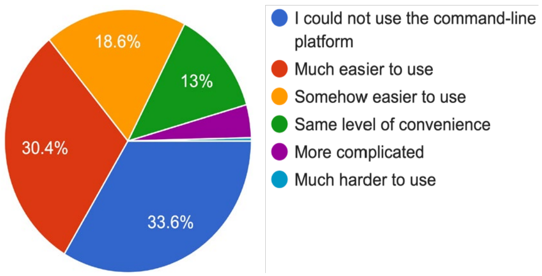


Figure 7. Participant Comparison of Visual Interface and Command-line Usability

Among the participants who were unable to use the command-line version, approximately 29% reported that they do not use Linux. Another 8.5% cited missing libraries required for compiling the command-line version, while 6.5% felt that the setup process was too labor-intensive. Additionally, 4.9% found the usage instructions unclear, and 3.2% specifically mentioned that the setup instructions lacked clarity. Furthermore, 15.5% selected "Other" and provided personalized explanations that echoed similar themes—for example: "While I have Linux on my work machine, I don't have it on my personal machine."

The last survey question provided qualitative insights into how participants perceived both versions of the system. Figure 8 presents a word cloud generated from participants' open-ended responses, highlighting frequently used terms that describe their experiences with the visual and command-line versions. While the word cloud does not offer a formal sentiment analysis, it provides a general sense of the language participants used. Terms such as "visual," "cloud-based," "easy," "UI," "clear," "intuitive," and "helpful" were commonly associated with the visual platform, suggesting a generally positive user experience. In contrast, words like "install," "error," "libraries," "bash," and "dependencies" were often linked to technical setup challenges encountered with the command-line version.

Importantly, these qualitative findings align with the structured questionnaire responses, in which participants rated the visual interface as significantly more convenient and accessible. This consistency between qualitative and quantitative feedback reinforces the conclusion that the visual platform effectively reduced usability barriers—particularly for users with limited technical expertise.

Figure 8. Word Cloud of Open-Ended Survey Responses

5. DISCUSSION

This study underscores the importance of developing accessible, cloud-based visual ML platforms that empower users to engage with ML tasks without requiring extensive technical expertise. The findings present several noteworthy pedagogical implications.

First, integrating such platforms into educational environments can democratize ML education, making it more inclusive for students across disciplines—from computer science to business, biology, and even academic writing. This interdisciplinary approach aligns with Altalouli and Curry's (2023) emphasis on embedding academic tools within broader learning contexts to foster meaningful engagement with complex analytical skills.

Second, the user-friendly interface of the visual ML platform examined in this study plays a crucial role in lowering entry barriers for students with limited coding experience. By enabling interaction with ML models through intuitive graphical tools, the platform allows learners to focus on applying ML techniques to real-world problems. Participant feedback revealed a strong preference for the visual interface over the command-line version, highlighting the value of simplifying access to technical workflows.

Third, the integration of NAS into this visual platform carries important implications for machine learning education, particularly for beginners. By making a traditionally complex process both accessible and interactive, NAS allows learners to engage not only with model building but also with the core principles of optimization—an essential yet often underexplored aspect of ML curricula. This dual exposure bridges theoretical and applied learning, offering students a unique opportunity to experiment with and observe the evolution of model structures in real time. More importantly, it provides them with hands-on access to one of the most innovative areas of modern ML research, helping demystify deep learning while fostering deeper engagement. Such exposure can inspire confidence, stimulate curiosity, and encourage students from non-technical backgrounds to explore advanced topics that might otherwise seem out of reach.

Fourth, the platform's cloud-based architecture supports scalable, flexible learning and collaboration. Students can access the system from any location, work with large datasets, and collaborate on projects—features that are particularly relevant in today's remote and hybrid learning environments. The visual tools also enhance interpretability, helping students bridge the gap between ML concepts and practical application. This is especially beneficial for learners who may find code-intensive ML tools challenging.

Fifth, while the findings support the platform's potential for educational use, it is important to acknowledge its current limitations. The prototype evaluated in this study is not a fully featured ML system; rather, it was designed to assess usability and accessibility. It does not yet evaluate learning outcomes or conceptual understanding of ML topics. Moreover, its reliance on cloud infrastructure may pose challenges in low-connectivity settings. Future development will focus on expanding functionality, integrating assessment tools, and conducting classroom-based evaluations to better understand its pedagogical impact.

Last, although this study does not directly measure instructional outcomes, it contributes to information systems (IS) education by demonstrating that usability-focused, accessible platform design can broaden engagement and participation in ML learning. These insights can guide curriculum design and instructional strategies, helping educators integrate ML platforms that make learning more accessible, interactive, and inclusive. It provides a pathway for students from diverse backgrounds to develop practical ML skills, encourages collaboration and experimentation, and ultimately prepares learners for the demands of a data-driven world.

6. RECOMMENDATIONS FOR EDUCATORS

To enhance ML education across diverse disciplines, we provide several general recommendations derived from the comparative evaluation of command-line and cloud-based visual ML platforms.

6.1 Lowering Barriers to ML Education

The study revealed that students—particularly those with limited programming experience—expressed a strong preference for the visual ML platform over the traditional command-line interface. This preference underscores the platform's effectiveness in reducing entry barriers, allowing learners to engage meaningfully with ML workflows without acquiring advanced coding skills. Additionally, one of the most impactful features of the platform, as highlighted by participant feedback, is its ability to visually represent model behavior—transforming ML concepts into tangible, interpretable outputs. This capability is particularly valuable for non-technical students, as it bridges the gap between theoretical knowledge and practical application.

6.1.1 Recommendation. Integrate visual ML platforms into introductory ML courses to help students focus on foundational concepts without the complexity of programming. This strategy promotes broader participation, especially among students from non-technical backgrounds.

6.1.2 Example. Using a drag-and-drop interface, students can design, train, and evaluate models with ease. This enables them to concentrate on essential ML principles—such as feature selection, model performance, and evaluation metrics—without being impeded by programming syntax or debugging challenges.

6.2 Facilitating Practical ML Projects

The positive feedback from participants—many of whom came from non-computational backgrounds—demonstrates that offering a visual, cloud-based ML platform significantly enhances learner engagement in practical ML projects by eliminating the technical overhead associated with programming and computing infrastructure. This finding aligns with existing research indicating that no-code and visual ML tools enable students to concentrate on higher-order tasks such as data preparation, model selection, and performance evaluation, rather than the intricacies of coding (Sundberg & Holmström, 2024).

The accessibility of these platforms is particularly beneficial for students in disciplines such as business, biology, or economics, where the application of ML to domain-specific problems is increasingly essential. By lowering technical barriers, visual ML tools promote broader participation in ML education and foster inclusive, interdisciplinary learning environments. This approach supports ongoing efforts to democratize artificial intelligence and enhance digital literacy across diverse academic fields (Cai et al., 2025).

6.2.1 Recommendation. Educators are encouraged to design assignments that leverage the visual platform to guide students through complete ML workflows—including data cleaning, model configuration, and result interpretation—without requiring programming knowledge. Customized modules can also be developed to help non-technical learners apply ML techniques to real-world problems within their respective domains.

6.2.2 Example. In a classification task, students can upload a dataset, select an appropriate model, and evaluate performance using metrics such as accuracy or precision through an intuitive graphical interface. For instance, in an economics course, students might use the platform to predict unemployment trends or GDP fluctuations, focusing on model logic and real-world implications rather than programming syntax.

6.3 Encouraging Experimentation and Discovery

Participant feedback indicated that the visual interface encourages experimentation by enabling users to modify models and instantly observe the outcomes. This interactivity cultivates a hands-on learning environment in which students are motivated to explore different algorithms, adjust hyperparameters, and assess model behavior.

6.3.1 Recommendation. Incorporate the platform into lab sessions where students can actively experiment with various ML algorithms and dynamically tune parameters. This exploratory approach reinforces conceptual understanding through immediate and visual feedback.

6.3.2 Example. Students can adjust neural network parameters—such as the number of layers or the learning rate—and instantly visualize the impact on model accuracy. This direct feedback loop bridges theoretical instruction with practical application, deepening students' comprehension of ML dynamics.

6.4 Enabling Cloud-Based Learning and Collaboration

The platform's cloud-based architecture enables students to access it from any device, making it particularly well-suited for remote instruction and collaborative learning. Participants in the study emphasized the convenience of not having to install software or configure hardware—common barriers that often hinder participation in ML assignments.

6.4.1 Recommendation. Utilize the platform for group projects, allowing students to collaboratively build, train, and evaluate models within a shared environment. The cloud infrastructure ensures equitable access, enabling full participation regardless of students' individual computing resources.

6.4.2 Example. In a collaborative assignment, students can divide responsibilities such as data cleaning, feature engineering, model training, and result interpretation within the same cloud-based workspace. This setup fosters teamwork while reinforcing the end-to-end ML process in an accessible and inclusive environment.

7. CONCLUSION

This study demonstrates that a user-friendly and cloud-based visual ML platform offers an effective approach to integrating ML education across a broad spectrum of academic disciplines. By abstracting technical complexities, the platform substantially lowers the barriers to entry, thereby enhancing both usability and learning experiences, particularly for individuals with limited programming skills.

This study acknowledges limitations that inform directions for future research and development. The current system remains a prototype, designed primarily to evaluate usability and user-friendliness rather than to serve as a fully operational ML deployment tool. While the prototype effectively demonstrates the potential of cloud-based visual ML interfaces, future work should focus on developing it into a fully functional platform that moves beyond usability testing and enables a more comprehensive assessment of its educational impact.

Additionally, ongoing development efforts should aim to refine and release the system as an open-source platform. This would promote broader adoption, foster community-driven contributions, and support long-term sustainability through collaborative innovation.

Lastly, to enhance the generalizability of the findings, future research should aim to include a more diverse participant group. This may involve K-12 educators, professionals from non-technical sectors, and participants from international academic institutions. Expanding the participant base in this way will help ensure that the platform is adaptable to a wide range of educational contexts and use cases, thereby strengthening its relevance and applicability across diverse learning environments.

8. ACKNOWLEDGEMENTS

The authors thank the journal editors and the anonymous reviewers for insightful and constructive feedback throughout the review process. The second author received support from the Summer Research Fund provided by the Cameron School of Business at the University of North Carolina Wilmington.

9. REFERENCES

Altalouli, M., & Curry, M.J. (2023). Integrating Awareness of Academic Reading Into Teaching Writing in an Additional Language. In H. Mohebbi & Y. Wang (Eds.), *Insights into Teaching and Learning Writing: A Practical Guide for Early Career Teachers* (pp. 13-26). Castledown. https://doi.org/10.29140/9781914291159-2

Berthold, M. R. (2019). What Does It Take to Be a Successful Data Scientist? *Harvard Data Science Review*, 1(2). https://doi.org/10.1162/99608f92.e0eaabfc

Cai, C., Zhu, G., & Ma, M. (2025). A Systemic Review of AI for Interdisciplinary Learning: Application Contexts, Roles, and Influences. *Education and Information Technologies*, 30, 9641-9687. https://doi.org/10.1007/s10639-024-13193-x

Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. & Menon, R. (2001). *Parallel Programming in OpenMP*. Morgan Kaufmann.

Darwish, A., Hassanien, A. E., & Das, S. (2020). A Survey of Swarm and Evolutionary Computing Approaches for Deep Learning. *Artificial Intelligence Review*, 53, 1767-1812. https://doi.org/10.1007/s10462-019-09719-2

ElSaid, A., Benson, S., Patwardhan, S., Stadem, D., & Desell, T. (2019). Evolving Recurrent Neural Networks for Time Series Data Prediction of Coal Plant Parameters. *Proceedings of International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 488-503), Springer, Cham. https://doi.org/10.1007/978-3-030-16692-2_33

ElSaid, A., Karns, J., Lyu, Z., Ororbia, A. G., & Desell, T. (2021). Continuous Ant-Based Neural Topology Search. *Proceedings of International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 291-306), Springer, Cham. https://doi.org/10.1007/978-3-030-72699-7_19

ElSaid, A., Ororbia, A. G., & Desell, T. J. (2020). Ant-Based Neural Topology Search (ANTS) for Optimizing Recurrent Networks. *Proceedings of International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 626-641), Springer, Cham. https://doi.org/10.1007/978-3-030-43722-0_40

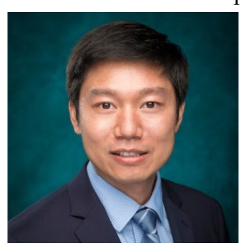
ElSaid, A. A. (2020). *Nature-Inspired Topology Optimization of Recurrent Neural Networks*. Rochester Institute of Technology ProQuest Dissertations & Theses, 2020. 28261478.

Elsken, T., Metzen, J.-H., & Hutter, F. (2017). *Simple and Efficient Architecture Search for Convolutional Neural*

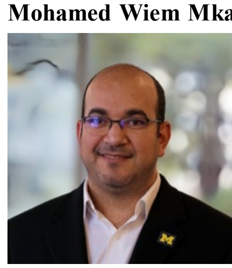
Networks. arXiv preprint arXiv:1711.04528. https://doi.org/10.48550/arXiv.1711.04528
Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 20(1), 1997-2017. https://www.jmlr.org/papers/volume20/18-598/18-598.pdf
Luna-Reyes, L. F. (2018). The Search for the Data Scientist: Creating Value From Data. *ACM SIGCAS Computers and Society*, 47(4), 12-16. https://doi.org/10.1145/3243141.3243145
Mason, D. (2013). Data Programming for Non-Programmers. *Procedia Computer Science*, 21, 68-74. https://doi.org/10.1016/j.procs.2013.09.011
Miikkulainen, R. (2020). Creative AI Through Evolutionary Computation. In: W. Banzhaf et al. (Eds.) *Evolution in Action: Past, Present and Future. Genetic and Evolutionary Computation* (pp. 265-269). Springer, Cham. https://doi.org/10.1007/978-3-030-39831-6_19
Ororbia, A., ElSaid, A., & Desell, T. (2019). Investigating Recurrent Neural Network Memory Structures Using Neuro-Evolution. *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 446-455). New York, NY: ACM. https://doi.org/10.1145/3321707.3321795
Roy, A., Qureshi, S., Pande, K., Nair, D., Gairola, K., Jain, P., ... & Kakarlapudi, A. V. (2019). Performance Comparison of Machine Learning Platforms. *INFORMS Journal on Computing*, 31(2), 207-225. https://doi.org/10.1287/ijoc.2018.0825
Topi, H. (2019). Reflections on the Current State and Future of Information Systems Education. *Journal of Information Systems Education*, 30(1), 1-9. http://jise.org/Volume30/n1/JISEv30n1p1.html
Schreck, B., & Veeramachaneni, K. (2016). What Would a Data Scientist Ask? Automatically Formulating and Solving Predictive Problems. *Proceedings of IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (440-451), IEEE. https://doi.org/10.1109/DSAA.2016.55
Sundberg, L., & Holmström, J. (2024). Teaching Tip: Using No-Code AI to Teach Machine Learning in Higher Education. *Journal of Information Systems Education*, 35(1), 56-66. https://doi.org/10.62273/CYPL2902
Wang, S., & Wang, H. (2021). Teaching Tip: A Teaching Module of No-Code Business App Development. *Journal of Information Systems Education*, 32(1), 1-8. https://jise.org/Volume32/n1/JISE2021v32n1pp1-8.html
Yang, Q., Suh, J., Chen, N. C., & Ramos, G. (2018). Grounding Interactive Machine Learning Tool Design in How Non-Experts Actually Build Models. *Proceedings of the 2018 Designing Interactive Systems Conference* (pp. 573-584). New York, NY: ACM. https://doi.org/10.1145/3196709.3196729

AUTHOR BIOGRAPHIES

AbdElRahman ElSaid is an assistant professor of computer science in the School of Science & Engineering at the University of North Carolina Wilmington. He earned his Ph.D. in Computing and Information Sciences from the Rochester Institute of Technology, with research focusing on NeuroEvolution and Computer Vision. His work spans neuro-evolutionary algorithms, deep learning, and scientific computing, with applications in domains such as aviation, agriculture, and biomedical imaging. Dr. ElSaid has published extensively in peer-reviewed journals including *Applied Soft Computing*, *Scientific Reports*, and *Software Impacts*, and regularly presents at premier conferences like GECCO and EvoStar. He has secured competitive research grants from NSF and Fulbright and actively contributes as a reviewer for journals such as *Nature Machine Intelligence* and *IEEE Access*.



Yao Shi is an assistant professor of information systems in the Congdon School of Supply Chain, Business Analytics, and Information Systems in the Cameron School of Business at the University of North Carolina Wilmington. He received his Ph.D. degree in business administration with a concentration in business information & technology from the University of Memphis. His research has been published in the *Journal of Database Management*, *International Journal of Medical Informatics*, *Journal of Information Systems Education*, and *Journal of Information Technology Management*, among others. He currently serves on the editorial board for the *Journal of Information Systems Education*. He has also served as an associate editor and mini track chair for several AIS conferences, including ICIS, AMCIS, and PACIS.



Mohamed Wiem Mkaour is currently an associate professor in the College of Innovation & Technology at the University of Michigan-Flint. He is the Graduate Director of the master's in software engineering and the master's in AI. His research interests are at the intersection of Software Engineering and AI. It includes software quality assurance, and systems refactoring, and he has co-authored over 140 peer-reviewed papers, including works appearing in top venues like TSE, TOSEM, EMSE, CHI, and ASE. Dr. Mkaouer has been PI/Co-PI on $4.5 Million in federally funded projects. He is the recipient of 6 best-paper / presentation awards, and he is the recipient of the Rochester Institute of Technology 2020 GCCIS best-emerging scholar award.

Mahmoud Altalouli is a faculty member in the Department of Education, Languages, and Instructional Design at the State University of New York (SUNY) Brockport, where he teaches TESOL and bilingual education courses. His research interests focus on critical thinking and its connection to reading, as well as the integration of artificial intelligence in higher education, particularly in the areas of reading and writing instruction.

Anas Tawfik is a senior software engineer at Booking.com, where he contributes to the Data Lineage team—building internal platforms that visualize and simplify the flow of data across the company. His work helps thousands of engineers and analysts understand how data moves, transforms, and connects, enabling greater trust and efficiency in Booking.com's data ecosystem. Beyond his role, Anas designs multi-tenant SaaS architectures, mentors emerging engineers, and explores AI-driven automation—turning experimental ideas into scalable, production-ready products. He is passionate about clean architecture, maintainability, and meaningful design, always striving to deliver elegant solutions to complex technical challenges.

INFORMATION SYSTEMS & COMPUTING ACADEMIC PROFESSIONALS

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)