Design of an AI-Enhanced Performance Evaluation Module in Project Management Platforms[*]

Viacheslav Zosimov[1,†], Oleksandra Bulgakova [1,*,†] and Victor Perederyi[2,†]

[1] *Odesa National University of Technology, Odesa, Kanatnaya Str., 65039, Ukraine*
[2] *Admiral Makarov National University of Shipbuilding, 9, Heroes of Ukraine ave., Mykolaiv, 54007, Ukraine*

Abstract

This paper introduces an AI-driven project management framework designed to enhance the evaluation of individual performance within collaborative environments. The core of the system lies in its ability to continuously analyze user activity data and transform behavioral patterns such as task completion rate, time deviations, communication activity, and task complexity into interpretable and adaptive performance scores. The framework integrates an intelligent analytics module based on machine learning. The resulting model allows for real-time scoring and explainable feedback, supporting data-driven decision-making in dynamic project settings. By utilizing interpretable AI and modeling feature interactions explicitly, the system bridges a critical gap in modern project management tools – namely, the lack of personalized, explainable, and adaptive evaluation mechanisms. The integration of this framework enables project teams to monitor performance proactively, improve transparency, and adapt management strategies in alignment with evolving work behaviors and collaboration dynamics.

Keywords

project management system, GIA GMDH, gradient boosting, behavioral analytics, machine learning, intelligent decision support, predictive modeling

1. Introduction

The digital transformation of project work has driven the evolution of tools for managing tasks, teams, and resources. As project complexity and distributed collaboration grow, traditional methods are being enhanced by web systems and AI. Modern IT projects demand flexibility, remote access, and intelligent automation. Integrating AI into project management enables data-driven decisions, predictive planning, and adaptive workflows – from passive tools to intelligent systems.

Recent research emphasizes the need for more intelligent and adaptive project management tools. According to [1], project success strongly correlates with maturity in project planning and monitoring tools, especially when those tools enable real-time responsiveness and transparent communication. Similarly, [2] stress the increasing role of information systems in aligning operational tasks with strategic goals.

A wide variety of commercial platforms has been developed to support project management. Tools such as Jira, Asana, Trello, ClickUp, Monday.com, Microsoft Project, and Wrike are widely used in practice [3-9]. Jira is favored by large Agile teams for its flexibility and deep integration options but is often criticized for its steep learning curve and overcomplexity for smaller projects [10]. Asana offers a simpler interface but lacks robust scalability [11], while Trello provides excellent visual task tracking but limited analytics and performance monitoring capabilities [12]. Each of these systems offers unique advantages, depending on the size of the team, the complexity of the project, and the degree of automation required. However, previous comparisons of such tools often rely on subjective assessments, such as "high usability" or "low scalability", without

[*] *ITPM-2025: VI International Workshop "IT Project Management" (ITPM 2025), May 22, 2025, Kyiv, Ukraine*
[1*] Corresponding author.
[†] These authors contributed equally.
zosimovvv@gmail.com (V. Zosimov); sashabulgakova2@gmail.com (O. Bulgakova); viperkms1@gmail.com (V. Perederyi)
0000-0003-0824-4168 (V. Zosimov); 0000-0002-6587-8573 (O. Bulgakova); 0000-0002-9241-3034 (V. Peredery)
© 2025 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 ISSN 1613-0073

CEUR-WS.org/Vol-4023/paper18.pdf

providing concrete metrics. To address this gap, the following table 1 presents a quantitative comparison of selected platforms based on measurable indicators.

Table 1
Comparison of popular project management systems [3-9]

System	Ease of use	Scalability	Customizability	AI capabilities
Jira	~2–4 weeks for full team onboarding	Unlimited users; enterprise-scale	1000+ plugins, REST API, custom workflows	Basic automation + ML via plugins
Asana	1–3 days for team onboarding	Up to 500 users per workspace	Moderate: Zapier, API, limited custom rules	Rule-based automation; no predictive AI
Trello	<1 day (very intuitive)	~50 active boards/user; up to 10 collaborators (free)	Power-Ups (1 free, unlimited paid), API	No native AI; manual configurations only
ClickUp	~1 week onboarding with support	Up to enterprise-level teams (1000+ users)	High: custom fields, automation, API	AI assistant in beta
Monday.com	1–3 days average onboarding	Scales to 2000+ users; 50,000+ items per workspace	Extensive templates, automation builder, API	Native AI assistant, beta-stage
MS Project	2–3 weeks (complex interface)	Scales across organizations (via MS 365)	Full custom fields, scripting, Power BI	Integrates with AI via Azure ML, no native AI
Wrike	5–7 days onboarding	Enterprise-ready (5000+ users possible)	400+ integrations, advanced workflow engine	AI-based risk prediction, task prioritization

Despite their popularity, current systems often lack built-in, objective methods to evaluate employee performance or provide automated recommendations based on predictive analytics. This gap is recognized in recent studies: for instance, [10, 12] suggest that integrating AI into project management could significantly improve forecasting accuracy and decision support, particularly through machine learning methods. Similarly, [11, 13] call for more intelligent systems that reduce the reliance on subjective managerial judgment. Several commercial project management platforms, such as Asana, Jira, and ClickUp, offer built-in analytics dashboards, yet they remain limited in terms of interpretability and personalization. Asana provides rule-based automation and workload overviews, but lacks granular, explainable performance modeling. Jira integrates agile metrics and predictive issue ranking via third-party plugins, though these often rely on proprietary algorithms that do not expose internal logic. In contrast, the approach proposed in this paper introduces a transparent, formula-based model derived from user activity data, which not only enables real-time evaluation but also ensures explainability – a key requirement in human-centric AI for project environments.

The proposed web-based system supports efficient project management, resource use, and productivity. It enables global access and simplifies collaboration, speeding up goal achievement. Integrated AI enhances decision-making through automated analysis and task prediction, helping assess team performance, detect bottlenecks, and refine planning.

The system is both a management tool and a strategic solution that drives results through intelligent automation. Its scientific novelty lies in applying interpretable machine learning for real-time performance evaluation, offering transparent, formula-based insights rather than black-box predictions.

2. System Architecture and Design

The development of a modern web-based project management platform requires a modular and scalable architecture capable of supporting various functionalities, including user interaction, task management, and intelligent analytics.

2.1. Software Modeling

The system is based on a classic three-tier architecture (Fig. 1), ensuring modularity, scalability, and clear separation of concerns. Users interact with the system through a web interface, which sends requests and displays responses. The application server handles core business logic, user management, task operations, and AI-powered analysis. The database layer stores project data, user profiles, task statuses, and AI-generated insights. Data is transmitted via encrypted channels to ensure security, and the architecture supports both vertical and horizontal scaling based on system load.

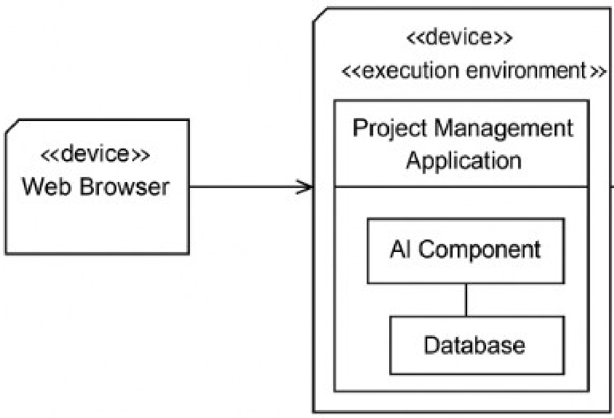

Figure 1: The deployment diagram.

The component diagram (Fig. 2) presents the modular structure of the system, divided into three logical layers: Frontend, Backend, and Database Access. This separation enhances flexibility, scalability, and maintainability by isolating the user interface, business logic, and data operations.

The Frontend layer includes the user interface elements such as the task board, project navigation, login forms, and real-time notifications. It interacts with the backend via an API Gateway, which handles routing, authentication, and load balancing.

The Backend layer consists of several services: user management, project and task handling, messaging and comments, AI analytics for performance evaluation and risk prediction, and a notification engine for system alerts and reminders.

The Database Access layer provides a secure interface for data persistence, managing logs and storing structured data in a relational database.

This architecture enables rapid development and allows integration with external systems, such as AI pipelines or domain-specific parsers (e.g., WDOL).

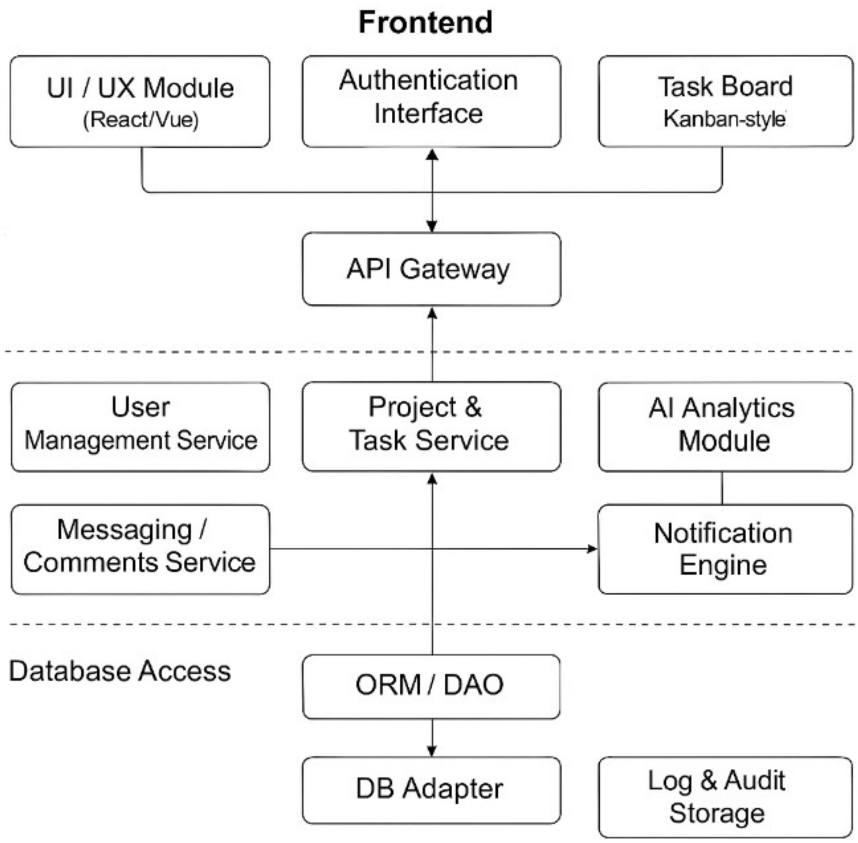

Figure 2: The component diagram.

To represent the internal data model of the system, a UML class diagram was developed (Fig. 3), defining the structure of core entities and their relationships. The model includes six main classes: User, Project, Task, Board, Comment, and AI_Score, each corresponding to a key aspect of system functionality.

The User class stores participant data, including roles and login metadata. Users can be assigned multiple tasks and contribute comments. The Project class represents collaborative initiatives, with attributes such as title, description, and lifecycle dates, and is linked to multiple tasks and boards for workflow visualization.

The Task class is central, containing data on deadlines, priority, and status, and serving as a hub for user activity and feedback. The Comment class captures messages linked to tasks, facilitating communication and documentation. The Board class organizes tasks visually by stage or status. The AI_Score class enables intelligent analytics by storing AI-generated performance evaluations, connecting task outcomes to individual users for productivity insights. Each class includes attributes like UUIDs, timestamps, and enumerated statuses to ensure integrity and machine-readability. Class relationships are defined using multiplicity indicators (e.g., one-to-many, many-to-one), forming the basis for database design and API structure.

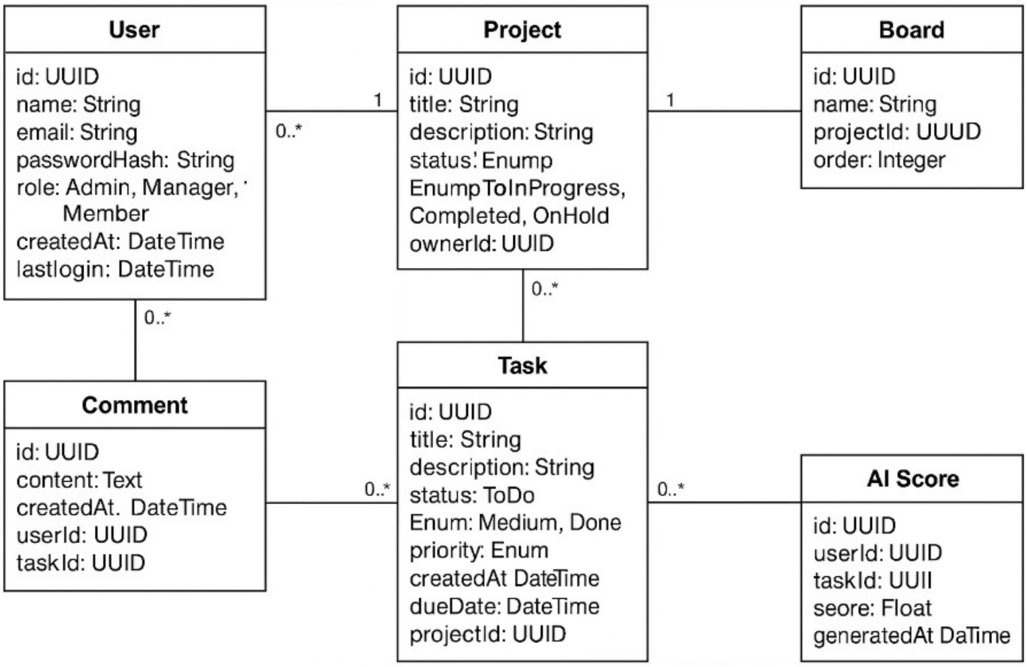

Figure 3: The class diagram.

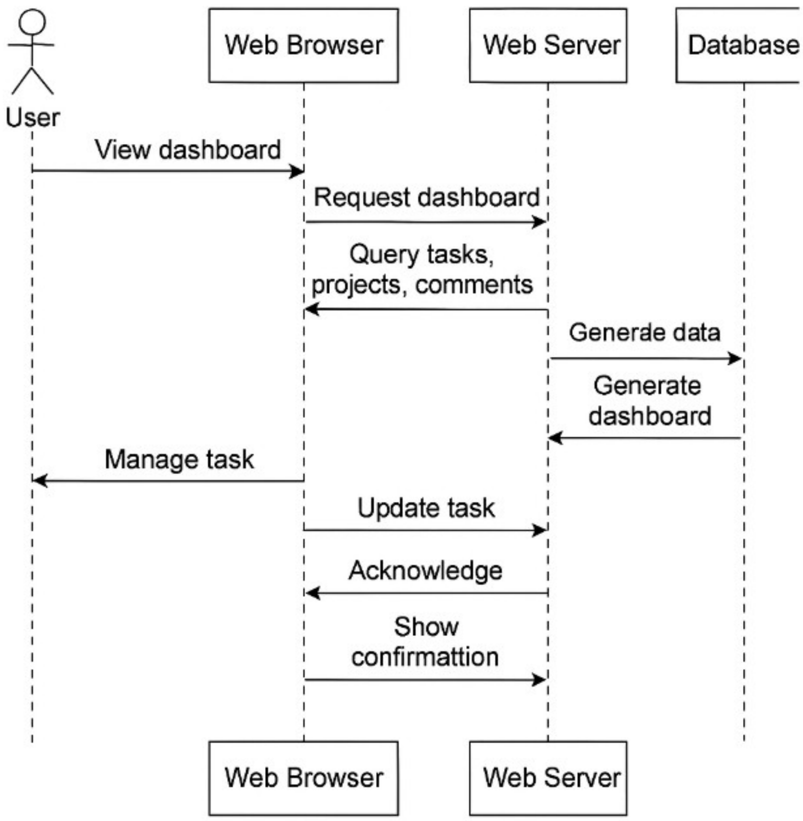

Figure 4: The sequence diagram.

The sequence diagram in Fig. 4 illustrates the system's dynamic behavior during a typical use case such as updating a task's deadline, by modeling interactions between the user interface, server logic, and data storage in time sequence.

The process begins when a user modifies a task attribute via the web interface. The client sends an HTTP request to the server, which authenticates the user and validates the input. Upon successful validation, the server updates the relevant task in the database. Once the update is confirmed, a response is returned to the client. If AI-based tracking is enabled, the system may also trigger a background recalculation of the related performance score using the analytics module.

The UML diagrams collectively demonstrate the system's modular design, internal logic, and consistent functionality. They validate its deployment feasibility, interaction flow, and data management processes.

2.2. AI Analytics

The proposed system features an AI-based analytics module that automates user activity assessment, generates performance indicators, and forecasts task completion dynamics.

A key goal of this module is to provide objective, adaptive evaluation of individual and team productivity. Unlike manual or rule-based methods, it uses machine learning to interpret behavior in context, revealing inefficiencies such as recurring delays or resource underuse that traditional tools may overlook.

The module also detects early deviations from normal work patterns. By monitoring task flow and user activity, it anticipates risks such as missed deadlines, uneven workload, or reduced collaboration, enabling timely interventions. Its predictive function evaluates historical and live data to estimate task completion probabilities, forecast workload distribution, and identify workflow inefficiencies. These capabilities improve planning accuracy and promote balanced responsibility across team members.

The following indicators were identified for the modeling:

- x_1 – number of tasks completed within the reporting period (tasks/day).
- x_2 – average time to complete a task (minutes/task).
- x_3 – deviation from estimated duration per task, defined as t_{actual} - $t_{estimated}$.
- x_4 – frequency of overdue tasks (% of total).
- x_5 – number of task reassignments initiated or received (events/week).
- x_6 – number of comments posted on tasks (comments/day).
- x_7 – number of responses to colleagues' messages (responses/day).
- x_8 – participation in non-task interactions (e.g., approvals, mentions) (events/week).
- x_9 – weighted task complexity index, calculated based on task priority, dependencies, and historical execution duration.

y is a continuous-valued performance score, scaled in the range from 0 to 100, where 0 corresponds to minimum observable productivity and 100 reflects the top observed efficiency under comparable task conditions. This score is intended to be interpretable by both system administrators and end users. For internal processing, the score may be decomposed into intermediate dimensions, such as technical efficiency, time management, and communicative engagement, each of which can be analyzed separately or jointly.

The dataset covers a period of 12 calendar weeks (84 days), during which the activity of 10 employees within a single project team was recorded.

The dataset includes all events related to task lifecycle management, communication exchanges, status transitions, as well as metadata concerning task complexity and deviations from planned timelines.

So, the data characteristics: m = 9, n = 7800 (total volume) n_A = 2/3·n, n_B = 1/3·n.

To construct a model for automated performance prediction, this study adopts the gradient boosting method. The task is formulated as a supervised regression problem, where the goal is to approximate a function f: $R^n \to R$ that maps an input feature vector x to a continuous performance score $y \in [0, 100]$. Gradient Boosted Decision Trees (GBDT) build a strong predictor through the iterative combination of weak learners – typically shallow regression trees – trained to minimize the residual error of previous approximations [14]. At each stage m, a new model hm(x) is fit to the negative gradient of the loss function L, evaluated with respect to the current prediction Fm-1(x). The updated model is expressed as:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{1}$$

with initialization:

$$F_0(x) = argmin \sum_{i=1}^{N} L(y_i, \hat{y}_i) \tag{2}$$

where γ_m is the learning rate; $h_m(x)$ is the m regression tree; y_i are the true performance scores; $L(y_i, \hat{y}_i)$ is the mean squared error. Each regression tree $h_m$partitions the input space based on threshold splits across the features and assigns a constant value to each leaf node. These trees are weak learners by design – often with maximum depth 3–5 – which reduces overfitting and allows the ensemble to generalize well across different patterns of user behavior. Early stopping is applied based on validation loss convergence, and cross-validation is used to fine-tune key hyperparameters such as tree depth, number of estimators, and subsampling ratio. The final output of the model is a predicted productivity score, which reflects the user's overall effectiveness in task completion and team interaction.

The trained gradient boosting model demonstrated a strong ability to approximate performance scores based on behavioral indicators. On the validation dataset (20% holdout), the model achieved a RMSE of approximately 4.83 points, with a MAE of 3.45 points, both measured on the performance score scale from 0 to 100. This level of error is acceptable in the context of behavioral prediction, where subjective variance in human performance evaluation is inherently high. In addition to its predictive accuracy, the model yielded interpretable insights through feature importance analysis. The top contributors to the final prediction included: tasks_completed – accounting for over 35% of the total model variance, reopened_tasks and time_deviation – jointly contributing 30%, comments_posted and responses_sent – adding around 20%, with the remaining importance distributed among collaboration and complexity indicators.

Figure 5 illustrates the internal logic of one of the trees in the ensemble, revealing how the model discriminates between users with stable, timely task execution patterns and those prone to delays or rework.

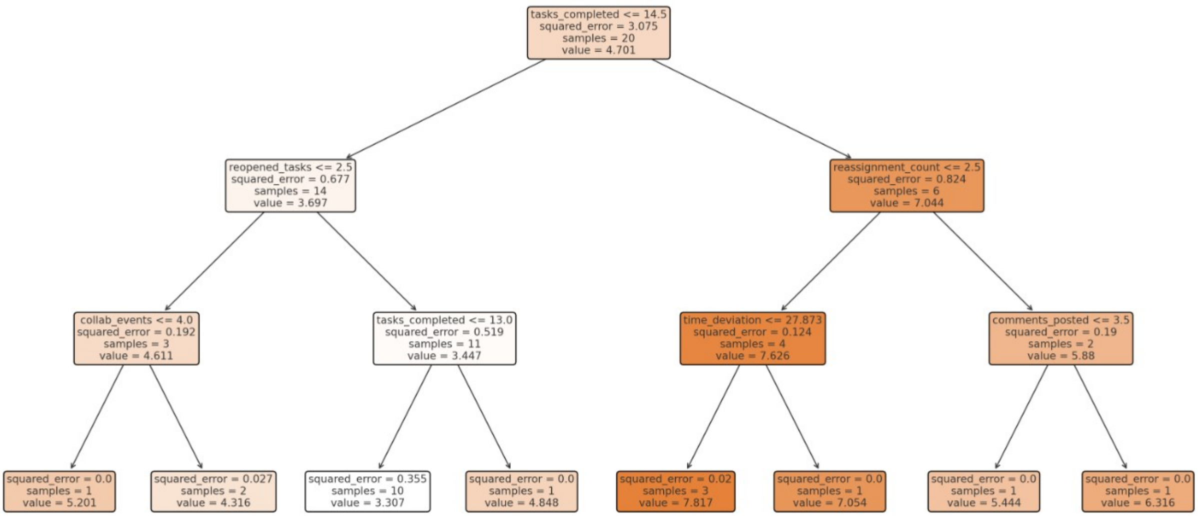

Figure 5: The visualization of a decision tree.

As an alternative to tree-based ensemble methods, this study investigates the application of the GIA GMDH. General characteristics of GMDH structural elements are described in [15-16].

Formally, in general case, a layer of the GIA GMDH may be defined as follows [17]:

1. The input matrix is $X_{r+1}=(y_1^r,\dots,y_F^r,x_1,\dots,x_m)$ for a layer r+1;

2. The operators of the kind:

$$y_l^{r+1}=f(y_i^r,y_j^r),l=1,2,\dots,C_F^2,i,j=\overline{1,F},\tag{3}$$
$$y_l^{r+1}=f(y_i^r,x_j),l=1,2,\dots,F_m,i=\overline{1,F},j=\overline{1,m}$$

may be applied on the layer r+1 to construct linear, bilinear and quadratic partial descriptions:

$$z=f(u,v)=a_0+a_1u+a_2v\tag{4}$$
$$z=f(u,v)=a_0+a_1u+a_2v+a_3uv$$
$$z=f(u,v)=a_0+a_1u+a_2v+a_3uv+a_4u^2+a_5v^2$$

3. For any description, the optimal structure is searched by combinatorial optimization; e.g.:

$$f(u,v)=a_0d_1+a_1d_2u+a_2d_3v\tag{5}$$

Then the best model will be described as $f(u,v,d_{opt})$, where

$$d_{opt}=\underset{l=\overline{1,q}}{argmin}\,CR_l,q=2^p-1,f_{opt}(u,v)=f(u,v,d_{opt})\tag{6}$$

4. The algorithm stops when the condition $CR^r>CR^{r-1}$ is checked

The best variant is chosen based on the minimum criterion CR, meaning the complexity of the partial model is optimized (6).

$$\hat{y}=5.8x_1+0.4x_4+0.3x_5+0.1x_9x_6+0.23x_3x_7+0.2x_2^2\tag{7}$$

$R^2(n_B)100\%$ = 81%

The structure of the model includes both linear and nonlinear components, as well as interaction effects between variables, which reflects real-world processes in project management environments. x_1 – number of tasks completed has the largest magnitude, confirming its role as the primary indicator of productivity. Moderate contributions are made by features related to execution discipline, such as x_4 – frequency of overdue tasks and x_5 – task reassignments, which aligns with the principles of team reliability and workload stability. Particular attention should be paid to the interaction terms x_9x_6 and x_3x_7 , which represent the synergy between task complexity and user engagement, as well as between schedule deviations and communication activity. These terms demonstrate that high engagement in complex or delayed tasks may offset otherwise negative performance indicators. The presence of a quadratic term x_2^2 indicates a nonlinear effect of execution speed: both excessively fast and excessively slow task completion can reduce overall performance, while an optimal time range maximizes the score.

3. Results

To assess the practical applicability of the developed predictive models, a comparative analysis was conducted between two algorithmic approaches: GBDT and the GIA-GMDH. The aim of the comparison was to evaluate not only the predictive performance, but also computational efficiency, model complexity, and interpretability – factors that are particularly relevant in decision-support systems intended for managerial use. The results of the comparison are summarized in Table 2.

The comparative results indicate that the Gradient Boosting model outperforms the GIA GMDH model in terms of prediction accuracy, achieving lower RMSE and MAE values on the validation dataset. This makes it a suitable choice for systems where high-precision evaluation of user performance is a top priority, particularly in large-scale deployments with sufficient computational resources.

However, the GIA GMDH model demonstrates significant advantages in terms of model simplicity, interpretability, and transparency. Its analytical form allows managers and analysts to explicitly understand how individual behavioral indicators influence performance scores. The reduced number of parameters and faster training time also make it well-suited for systems operating in real-time or resource-constrained environments.

Thus, the selection between these two models should be guided by the specific requirements of the application domain. If maximum accuracy and scalability are essential, GBDT is recommended. In contrast, if model explainability, control over behavior-to-score mapping, or ease of integration is prioritized, GIA GMDH presents a more appropriate solution.

Table 2
Comparative evaluation of predictive models

Metric	Gradient boosting	GIA GMDH
RMSE	4.83	5.37
MAE	3.45	3.91
Training time (sec)	2.10	1.20
Number of model parameters	150	28
Model depth / layers	3	4

4. Discussion and Conclusions

This research explored the integration of intelligent analytics into a web-based project management system with the goal of enhancing individual performance assessment through data-driven methods. Two machine learning algorithms GBDT and the GIA-GMDH were investigated and compared to identify the most appropriate approach for integration into the system.

The comparative analysis revealed that while GBDT offered slightly higher predictive accuracy, GIA GMDH demonstrated greater transparency, interpretability, and analytical compactness. These qualities are essential for real-time feedback and managerial control within the context of project execution. The GIA GMDH model also provided an explicit mathematical formula that enables the direct calculation of performance scores based on a concise set of behavioral indicators, such as task volume, communication, timing, and complexity.

Based on these findings, the GIA-GMDH approach was selected as the preferred model for implementation within the intelligent layer of the project management system. The results validate the potential of combining structured behavioral data with interpretable machine learning models to enhance transparency and decision support in collaborative work environments. However, the study also has limitations. The dataset was based on a controlled pilot environment and may not fully reflect the complexity and variability of real-world project teams.

Future work should explore scaling the approach across diverse organizations, incorporating richer behavioral dynamics, and integrating context-aware or emotion-sensitive inputs.

Enhancements such as fuzzy logic or hybrid inference could further improve the system's robustness under uncertainty and incomplete data.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Serrador, J. Pinto, Does Agile work? – A quantitative analysis of agile project success, volume 33 (5) of International Journal of Project Management, 2015, pp.1040-1051. doi: 10.1016/j.ijproman.2015.01.006

[2] E. Karaman, M. Kurt, Comparison of project management methodologies: Prince 2 versus PMBOK for IT projects. Procedia - Social and Behavioral Sciences, 195, 2015, pp.305–314. doi: 10.1016/j.sbspro.2015.06.193

[3] Jira Software: Project and issue tracking, 2023, URL: https://www.atlassian.com/software/jira

[4] The work management platform for teams, 2023, URL: https://asana.com

[5] Trello. Organize anything, together, 2023, URL: https://trello.com

[6] ClickUp. Organize anything with ClickUp's Project Hierarchy. URL: https://clickup.com/hierarchy-guide

[7] What is monday.com and why do you need it? URL: https://enable.services/2024/05/16/what-is-monday-com-and-why-do-you-need-it/

[8] M. Keup, What Is Microsoft Project? Uses, Features and Pricing, 2022. URL: https://www.projectmanager.com/blog/what-is-microsoft-project

[9] E. Westbrooks, How AI automation is transforming workflows: Expert tips from Collaborate, 2025. URL: https://www.wrike.com/blog/enhancing-productivity-project-management/

[10] S. Jayanthi, J. J, I. Begum, E. Ramesh, R. Balaraman and S. Gunasree, "Leveraging Machine Learning to Predict IFC Project Success and Failure," 2024 First International Conference for Women in Computing (InCoWoCo), Pune, India, 2024, pp. 1-6, doi: 10.1109/InCoWoCo64194.2024.10863562.

[11] A. Lavagnon, K. Jeffrey, The "re-meaning" of project success: Updating and recalibrating for a modern project management, International Journal of Project Management, 2022, 40(7), pp. 835-848. doi: 10.1016/j.ijproman.2022.08.001.

[12] D.S. Adamantiadou, L. Tsironis, Leveraging Artificial Intelligence in Project Management: A Systematic Review of Applications, Challenges, and Future Directions. Computers 2025, 14, 66. doi: 10.3390/computers14020066

[13] V. Zosimov, O. Bulgakova, A. Kudriavtsev "Empowering Partnerships in Business: a System for Real-Time Data Integration and Collective Analysis", 18th IEE Int. Conf. on Computer Science and Information Technologies CSIT-2023, 2023. doi: 10.1109/CSIT61576.2023.10324191

[14] . Jun, M. J. (2021). A comparison of a gradient boosting decision tree, random forests, and artificial neural networks to model urban land use changes: the case of the Seoul metropolitan area. International Journal of Geographical Information Science, 35(11), 2149–2167. doi: 10.1080/13658816.2021.1887490

[15] O. Bulgakova, V. Stepashko, V. Zosimov "Numerical study of the generalized iterative algorithm GIA GMDH with active neurons", Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, 2017, pp. 496–500. doi: 10.1109/STC-CSIT.2017.8098836.

[16] V. Stepashko, "Inductive modeling from historical perspective", Proc. Int. Conf. on Computer Science and Information Technologies CSIT-2017, Lviv: LNPU, 2017, vol. 1, pp. 537-542.

[17] V. Zosimov, O. Bulgakova, Web Data Displaying Approach Based on User's Semantic Profile Templates, IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 2020, pp. 428-431, doi: 10.1109/CSIT49958.2020.9321839