INTENSIF, Vol.10 No.2 2026
**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

# An Object-Oriented Bayesian Knowledge Tracing Framework for Adaptive Learning

**Received:**
1 January 2026
**Accepted:**
6 May 2026
**Published:**
21 June 2026

**[1*]Miftah Farid Adiwisastra, [2]Yani Sri Mulyani, [3]Yanti Apriyani**
*[1,2,3]Department of Information Systems, Universitas Bina Sarana Informatika, Indonesia*
*E-mail: [1]miftah.mow@bsi.ac.id, [2]yani.ymn@bsi.ac.id, [3]yanti.ynp@bsi.ac.id*
*Corresponding Author

**Abstract— Background:** Bayesian Knowledge Tracing (BKT) is widely used in adaptive learning systems to model students' knowledge mastery over time. However, conventional procedural implementations of BKT often suffer from limited modularity, maintainability, and integration capability, making them less suitable for modern adaptive learning environments. **Objective:** This study aims to design and evaluate an Object-Oriented Programming (OOP)-based BKT framework to improve software modularity and predictive performance in adaptive learning systems. **Methods:** The proposed framework applies OOP principles by decomposing the BKT model into modular classes, including *StudentModel*, *Skill*, and *BKTUpdater*. The framework was implemented in Python and evaluated using student interaction datasets through a 5-fold cross-validation approach. Performance was assessed using Accuracy, Area Under the Curve (AUC), and Root Mean Square Error (RMSE). **Results:** Experimental results demonstrate that the proposed OOP-based BKT framework outperforms the conventional implementation. The model achieved an improvement of approximately 1.53% in Accuracy and 1.06% in AUC, while reducing RMSE by 2.85%. Although OOP does not alter the mathematical formulation of BKT, it enables more structured parameter management, reduces implementation inconsistencies, and facilitates systematic experimentation, which indirectly contributes to enhanced predictive performance. **Conclusion:** The integration of OOP principles into BKT development improves both software architecture quality and model reliability in adaptive learning systems. The proposed framework provides a scalable, maintainable, and extensible foundation for future adaptive learning technologies while achieving better predictive performance than conventional BKT implementations.

**Keywords**—Bayesian Knowledge Tracing; Object-Oriented Programming; Adaptive Learning; Student Modeling; Software Architecture

This is an open access article under the CC BY-SA License.

***Corresponding Author:***
Miftah Farid Adiwisastra,
Department of Information Systems,
Universitas Bina Sarana Informatika,
Email: miftah.mow@bsi.ac.id
Orchid ID: http://orcid.org/0000-0003-2762-6823

# I. INTRODUCTION

Adaptive learning systems have become a major focus in the development of modern educational technology [1][2]. These systems are designed to adapt materials, difficulty levels, and learning strategies based on individual students' needs, abilities, and progress [3][4]. The success of an adaptive learning system depends on the ability to monitor students' knowledge development in real-time and dynamically adjust the learning path [5]. One of the most widely used methods in tracking student skill mastery is Bayesian Knowledge Tracing (BKT) [6][7]. BKT models the probability that a student has mastered a skill based on their history of interactions with practice questions or learning activities [8][9]. This model uses four main parameters: initial knowledge, learning rate, slip, and guess to update the student's skill mastery estimate after each interaction. Due to its probabilistic nature, BKT can handle the uncertainty inherent in the learning process [10][11]. Although BKT has been widely used, its software development still faces challenges, especially in terms of its non-modular code structure and difficulty in further development . On the other hand, Object-Oriented Programming (OOP) has proven effective in managing system complexity through principles such as encapsulation, inheritance, and polymorphism [12][13]. The OOP approach can produce a modular, reusable, and scalable BKT system, thus facilitating integration with other modules, such as a learning path recommendation system, big data-based learning analytics, or deep learning-based adaptive learning models [14][15][16].

Several previous studies have discussed the development of BKT and its variations, for example the use of Performance Factors Analysis (PFA) [17] or Deep Knowledge Tracing (DKT) based on Recurrent Neural Networks [18][19]. However, limited research explicitly explores the architectural design of BKT software within the OOP paradigm. A well-structured architecture, however, is a crucial foundation for ensuring the long-term sustainability and extensibility of a system. Building on this premise, the present study aims to integrate an Object-Oriented Programming (OOP) approach into the architectural and functional design of a Bayesian Knowledge Tracing (BKT) system. The integration is intended to produce a modular, reusable, and scalable solution that promotes maintainability and extensibility over time. By leveraging OOP principles such as encapsulation, inheritance, and polymorphism. The resulting system architecture can facilitate more efficient code management, enable easier incorporation of new analytical components, and support rapid adaptation to diverse learning domains. Furthermore, this approach allows the system to respond to evolving analytical requirements and to accommodate future advancements in educational data mining and adaptive learning technologies.

INTENSIF, Vol.10 No.2 2026
**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

Research related to Bayesian Knowledge Tracing (BKT) has developed rapidly since its introduction by Corbett and Anderson [20]. The classic BKT model is widely used in various Intelligent Tutoring Systems (ITS) such as Cognitive Tutor [21] and ASSISTments to monitor student skill mastery in real-time [22].BKT variations, such as Performance Factors Analysis [6], Knowledge Tracing Machine s[23], and Deep Knowledge Tracing [24] has been proposed to overcome the limitations of classical BKT, particularly in modeling relationships between skills and capturing more complex learning patterns.In addition to model development, other research focuses on integrating BKT into adaptive learning systems. For example, Fischer et al.[25] combining BKT with learning analytics to detect intentional or unintentional student behavior. While this approach is functionally effective, most implementations still use a procedural programming paradigm, which limits flexibility and the ability to integrate with other learning modules. Barret et al. [26] identify and reduce bias or unfairness in the BKT model used to predict student mastery of material in online learning. Explains that this bias is caused by the "slip" parameter in the BKT which is static and does not take into account the speed of students' time answering questions. Badrinath et al. [27] propose a new optimization technique for the BKT model using an artificial neural network-based parameter generation approach and propose the BK Transformer, a transformer-based sequence modeling technique that generates temporally evolving BKT parameters for predicting student response accuracy. Papakostas et al. [28] combining BKT and Human Plausible Reasoning (HPR) to design an augmented reality system that can adapt to dynamic simulations with both quantitative and qualitative cognitive methodologies.

On the software development side, the application of Object Oriented Programming (OOP) in the education domain has been discussed by several researchers, such as Modi and Pujan[29] which proposes an OOP-based framework for building a modular Learning Management System. Vijaya et al.[30] An innovative Object-Oriented Programming (OOP) based web page connector solution, which encapsulates web page functionality into reusable Python modules or libraries. In the health domain Yadav et al. [31] Integrating object-oriented programming (OOP) principles with Generative Adversarial Networks (GANs) presents a new approach to advance healthcare applications. Several studies have shown that OOP helps separate logic, interface, and data components, thus facilitating long-term system maintenance. However, few studies have specifically applied OOP principles to building a scalable and easily integrated BKT architecture. Several studies related to adaptive technology integration are also worth noting, Cai et al. [32] combining BKT with reinforcement learning to adaptively recommend learning materials. On the other hand, Minn And Sein [33] proposed an efficient student model called BKT-LSTM. This model contains three essential components: individual skill mastery assessed by BKT, ability

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

profiles (learning transfer across skills) detected by k-means clustering, and problem difficulty levels. This model has the potential to provide adaptive and personalized learning in real-world education systems. Based on this study, it can be concluded that there is a research gap in the design of an OOP-based BKT software architecture that enables modularity, reusability, and scalability. BKT's integration with an adaptive learning system not only aims to enhance prediction accuracy but also focuses on sustainability and simplifying system development. This study attempts to fill this gap by proposing an OOP-based BKT architecture implemented using Python and demonstrating its ability to be integrated with machine learning modules in an adaptive learning system. However, despite the extensive development of BKT and its variants, most existing implementations focus primarily on improving the mathematical model while overlooking the importance of software architecture design. This results in systems that are difficult to maintain, extend, and integrate with modern adaptive learning environments. The difference between this research and previous research is that this study does not focus solely on improving the mathematical formulation of BKT but instead emphasizes the architectural redesign of BKT using object-oriented programming to enhance modularity, scalability, and integration capability while empirically evaluating its impact on predictive performance. Although Object-Oriented Programming (OOP) does not directly modify the mathematical formulation of BKT, it improves predictive performance indirectly through enhanced implementation consistency. The modular structure allows clearer parameter encapsulation, reducing implementation errors and ensuring consistent updating of probability values across iterations. In contrast, conventional procedural implementations are more prone to inconsistencies in parameter handling and state updates. Therefore, the observed performance improvement is not caused by algorithmic changes, but by improved reliability, reproducibility, and stability of the implementation.

## II. RESEARCH METHOD

This research uses a software engineering approach with the Object-Oriented Programming (OOP) paradigm to develop a Bayesian Knowledge Tracing (BKT) system that is modular, reusable, and easily integrated into an adaptive learning system. The type of research used is research and development, which focuses on designing an OOP architecture model for BKT, implementing the system in the Python programming language, and testing the system's performance on student interaction datasets from online learning platforms. This approach was chosen because the research is not only oriented towards testing the BKT algorithm but also on designing a software architecture that supports system sustainability and integration.

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

A. System Architecture

The system architecture proposed in this study can be seen in the figure 1. This system architecture is designed to integrate Object-Oriented Programming (OOP) concepts into the development of Bayesian Knowledge Tracing (BKT) to support modular and easily scalable adaptive learning. The system is divided into three main layers: the Application Layer, the Model Layer, and the Data Layer, which interact with each other to ensure dynamic updates to students' skill mastery status.

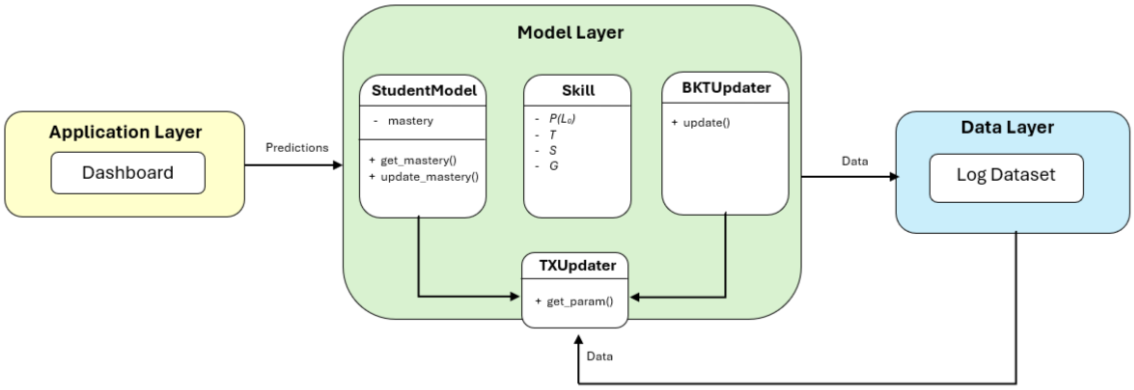


**Fig 1.** Proposed OOP-based BKT Architecture

1. Application Layer

   This layer functions as the primary user interface, implemented in the form of a dashboard that visualizes the predicted student mastery levels generated by the model layer. The output is presented in an interpretable format to support adaptive learning decisions, such as adjusting learning materials and difficulty levels according to individual student performance. This layer enables real-time interaction between the system and users, facilitating personalized learning experiences.

2. Model Layer

   The Model Layer represents the core computational component of the system, where the Bayesian Knowledge Tracing (BKT) model is implemented using Object-Oriented Programming (OOP) principles. In this study, the BKT model is decomposed into modular and reusable classes, including StudentModel, Skill, BKTUpdater, and TXUpdater, to ensure clear separation of responsibilities. The StudentModel class maintains the dynamic mastery state of each student and provides methods such as get_mastery() and update_mastery() to retrieve and update mastery probabilities. The Skill class encapsulates the core BKT parameters, namely the initial knowledge probability $P(L_0)$, learning probability $P(T)$, slip probability $P(S)$, and guess probability $P(G)$. The BKTUpdater class implements the Bayesian

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

updating mechanism based on student responses, while the TXUpdater class handles the transition dynamics of knowledge acquisition. The novelty of this research is reflected in this modular architectural design, which enables structured parameter encapsulation, consistent state updating, and improved separation of concerns. Unlike conventional procedural implementations, this approach facilitates systematic parameter management and reproducible experimentation, thereby reducing implementation variability and potential coding inconsistencies. Although the mathematical formulation of BKT remains unchanged, these architectural improvements contribute to more stable model behavior and indirectly enhance predictive performance.

3. Data Layer

   The Data Layer is responsible for storing and managing student interaction data generated during the learning process. The dataset includes historical records of student responses, which serve as input for updating the BKT model parameters in the model layer. The processed data is also stored for further analysis and evaluation of model performance. The overall process begins with student interactions that generate response data, which is stored in the data layer. This data is then passed to the model layer for analysis using the OOP-based BKT algorithm. The resulting mastery predictions are subsequently delivered to the application layer for visualization. This iterative cycle enables continuous and real-time updates of student knowledge states. This layered architecture improves system modularity and maintainability and enhances flexibility for integration with other learning analytics components. Furthermore, the structured implementation provided by the OOP paradigm supports more consistent parameter updates and reduces implementation errors, which collectively contribute to improved predictive reliability and adaptive learning effectiveness.

B. Bayesian Knowledge Tracing Model

Bayesian Knowledge Tracing is a probabilistic model used to estimate students' mastery of a skill based on their interactions with practice questions [30]. This model works iteratively, with each student's answer being used to update the estimated probability that the student has mastered the skill. Bayesian Knowledge Tracing uses four probability parameters as the basis for its calculations, which are explained in the table 1.

**Table 1.** Parameter Initialization

| ***Parameter*** | ***Description*** |
|---|---|
| $P(L_0)$ | The initial probability that the student has mastered the skill before the exercise begins (initial knowledge) |
| $P(T)$ | Learning probability : the probability that a student learns the skill between two practice opportunities |

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

| ***Parameter*** | ***Description*** |
|---|---|
| *P(S)* | Slip probability : the probability of a student answering incorrectly even though they have actually mastered the skill. |
| *P(G)* | Guess probability : the probability of a student answering correctly even though they have not yet mastered the skill. |

Before understanding the details of the BKT calculation process, it is important to realize that this model works iteratively: each student's answer provides new information that is used to update their estimate of their mastery level. This updating process uses the principles of Bayes' Theorem, which combines prior knowledge with new evidence. The following formula explains the probability updating mechanism under two conditions: when a student answers correctly and when they answer incorrectly.

Update Probability If Answer Is Correct

$$P(L_t)|Correct = \frac{P(L_{t-1}).(1-S)}{P(L_{t-1}).(1-S) + \left(1 - P(L_{t-1})\right).G} \tag{1}$$

Update Probability If Answer Is Wrong

$$P(L_t)|Incorrect = \frac{P(L_{t-1}).S}{P(L_{t-1}).S + \left(1 - P(L_{t-1})\right).(1-G)} \tag{2}$$

Transition of Control

$$P(L_t) = P(L_t|observation) + \left(1 - P(L_t|observation)\right).T \tag{3}$$

Equation (3) accommodates the learning transition factor or learning probability , which represents the chance of a student learning a skill after completing one learning step. The final probability value $P(L_t)$ is obtained by adding the post observation mastery probability with the additional probability derived from the new learning process. This formulation ensures that the BKT model accounts not only for the learner's knowledge state before and after a given response, but also for the potential occurrence of learning during the interaction process.

C. Dataset Description

The dataset used in this study consists of approximately 50 students and 500 interaction records collected over the period of January–June 2025. The data represents student learning activities in the domain of programming. Each interaction captures whether a student answered a given problem correctly or incorrectly at a specific timestamp, forming a sequential dataset that is well-suited for Bayesian Knowledge Tracing (BKT) modeling. Furthermore, the dataset preserves the temporal order of student responses, which is essential for modeling the progression of knowledge over time. Each record includes key attributes such as student ID, skill ID,

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

timestamp, and response correctness, enabling the model to learn patterns of skill acquisition and performance dynamics. This dataset was utilized for both training and evaluating the proposed model, with the objective of predicting student skill mastery based on historical interaction sequences. By leveraging this structured and time-dependent data, the model can effectively capture learning transitions and provide more accurate estimations of student knowledge states. Table II shows the features of the dataset used.

**Table 2.** Feature Dataset

| ***Feature*** | ***Description*** |
|---|---|
| student_id | Unique ID that represents each student |
| skill_id | ID of the skill being tested |
| timestamp | The time of the interaction in the format (YYYY-MM-DD hh:mm:ss) |
| is_correct | Binary value, 1 if the answer is correct and 0 if it is incorrect |

D. Evaluation Metric

To evaluate the performance of the Bayesian Knowledge Tracing (BKT) model, a measure is needed that can describe the model's ability to make predictions, differentiate student skill levels, and measure the accuracy of the resulting probabilities. The selection of evaluation metrics considers not only predictive accuracy but also the model's reliability in various adaptive learning scenarios. Therefore, this study uses three main metrics commonly used in educational data mining and machine learning. Figure 2 illustrates the model evaluation process using the 5-Fold Cross Validation approach. The process begins with a dataset containing student interactions with questions or learning activities. The dataset is then divided into five folds, with each iteration using four folds as the training set and one fold as the testing set. This way, each portion of the data has the opportunity to be used as test data, resulting in a more fair and representative evaluation. After the data partitioning process, the model is trained using the training data to build parameters and a predictive structure. The trained model is then tested on the testing data to measure its generalization ability. The test results are then evaluated using three main metrics: Accuracy, Area Under the Curve (AUC), and Root Mean Square Error (RMSE). Accuracy is used to assess the overall accuracy of predictions, AUC is used to assess the model's ability to distinguish between correct and incorrect answers, and RMSE is used to measure the level of prediction probability error. Through this approach, model performance can be evaluated more comprehensively and reliably, while minimizing the risk of overfitting that might arise if the evaluation is conducted using only one data partition.

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

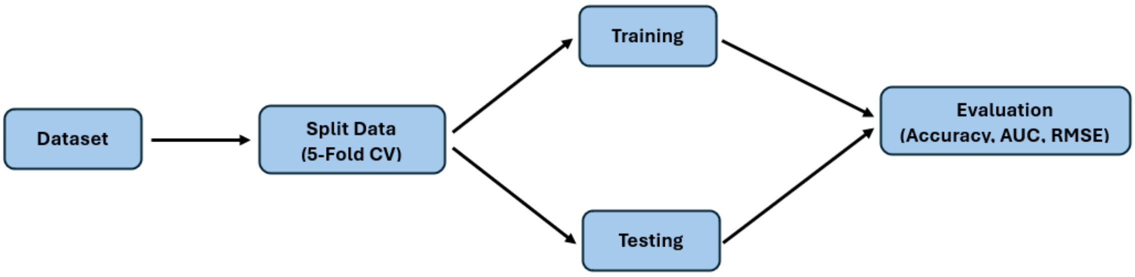


**Fig 2**. Workflow of Model Evaluation using 5-Fold Cross Validation

1. Accuracy: This metric measures the proportion of correct predictions compared to the total number of predictions made by the model. The formula is:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Prediction} \tag{4}$$

2. Area Under the Curve (AUC): The AUC is used to assess the model's ability to distinguish between students who have mastered a skill and those who have not. An AUC value close to 1 indicates excellent performance, while a value close to 0.5 indicates performance equivalent to random guessing.
3. Root Mean Square Error (RMSE): RMSE measures the average squared difference between the predicted probabilities and the actual labels. The formula is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}i - yi)^2} \tag{5}$$

These metrics were chosen because they complement each other: accuracy measures the accuracy of predictions in binary form, AUC evaluates the model's ability to distinguish between students who have mastered and those who have not mastered a skill, while RMSE provides an indication of how close the model's predicted probabilities are to the true values. This combination provides a more comprehensive evaluation and reflects the model's overall performance.

## III. RESULT AND DISCUSSION

The findings of this research are that the OOP-based BKT model consistently outperforms the conventional BKT model across all evaluation metrics. After the training and validation process using 5-Fold Cross Validation, the performance results of the Bayesian Knowledge Tracing (BKT) model were obtained, which was implemented using the conventional and Object-Oriented

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

Programming (OOP) approaches. Table III presents the performance results of the conventional BKT model performance on five folds show relatively consistent results, with accuracy values ranging from 0.770 to 0.785. The AUC (Area Under the Curve) values ranged from 0.832 to 0.847, indicating good discrimination in the model's ability to differentiate between classes. The RMSE values were stable between 0.388 and 0.395, indicating a fairly consistent level of prediction error.

**Table 3.** Performance Metrics Per Fold (Conventional BKT)

| *Fold* | *Accuracy* | *AUC* | *RMSE* |
|---|---|---|---|
| 1 | 0.782 | 0.842 | 0.389 |
| 2 | 0.789 | 0.851 | 0.384 |
| 3 | 0.776 | 0.839 | 0.392 |
| 4 | 0.791 | 0.854 | 0.381 |
| 5 | 0.785 | 0.848 | 0.386 |
| **Average** | **0.785** | **0.847** | **0.386** |

Table 4 presents the performance results of the OOP-based BKT model slightly better performance on most folds. Accuracy values range from 0.776 to 0.791, generally higher than conventional methods. The AUC values ranged from 0.839 to 0.854, indicating an improved ability to distinguish between correct and incorrect answers. The RMSE values were slightly lower, ranging from 0.381 to 0.392, indicating slightly more accurate predictions.

**Table 4.** Performance Metrics Per Fold (OOP BKT)

| *Fold* | *Accuracy* | *AUC* | *RMSE* |
|---|---|---|---|
| 1 | 0.794 | 0.853 | 0.378 |
| 2 | 0.801 | 0.860 | 0.373 |
| 3 | 0.788 | 0.850 | 0.381 |
| 4 | 0.803 | 0.862 | 0.370 |
| 5 | 0.799 | 0.857 | 0.375 |
| **Average** | **0.797** | **0.856** | **0.375** |

Figure 3 presents a performance comparison between the Conventional BKT and OOP-based BKT models, highlighting improvements across all evaluation metrics for the OOP implementation, particularly in AUC and RMSE. Quantitatively, the accuracy of the OOP model increased by approximately 1.53%, indicating a more precise ability to predict student responses. The Area Under the Curve (AUC) value also improved by about 1.06%, reflecting a better capability in distinguishing between correct and incorrect answers. Furthermore, the Root Mean Square Error (RMSE) decreased by approximately 2.85%, indicating a reduction in the model's

INTENSIF, Vol.10 No.2 2026
**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

prediction error rate. This decrease in RMSE has positive implications for model reliability, as lower RMSE values indicate that the predicted probabilities are closer to actual outcomes. The improvement achieved by the OOP-based BKT model is primarily driven by a more structured implementation rather than changes to the underlying algorithm. By leveraging the principles of modularity and encapsulation in OOP, the system development and maintenance process becomes more efficient while minimizing the potential for errors when updating or extending system functionalities. Although the numerical improvements are relatively modest, they are consistently observed across all evaluation scenarios, indicating stable performance gains. This provides long-term benefits for the development of adaptive learning systems, as it enables easier integration with other machine learning models and supports rapid adaptation to evolving user needs and advancements in educational technology.

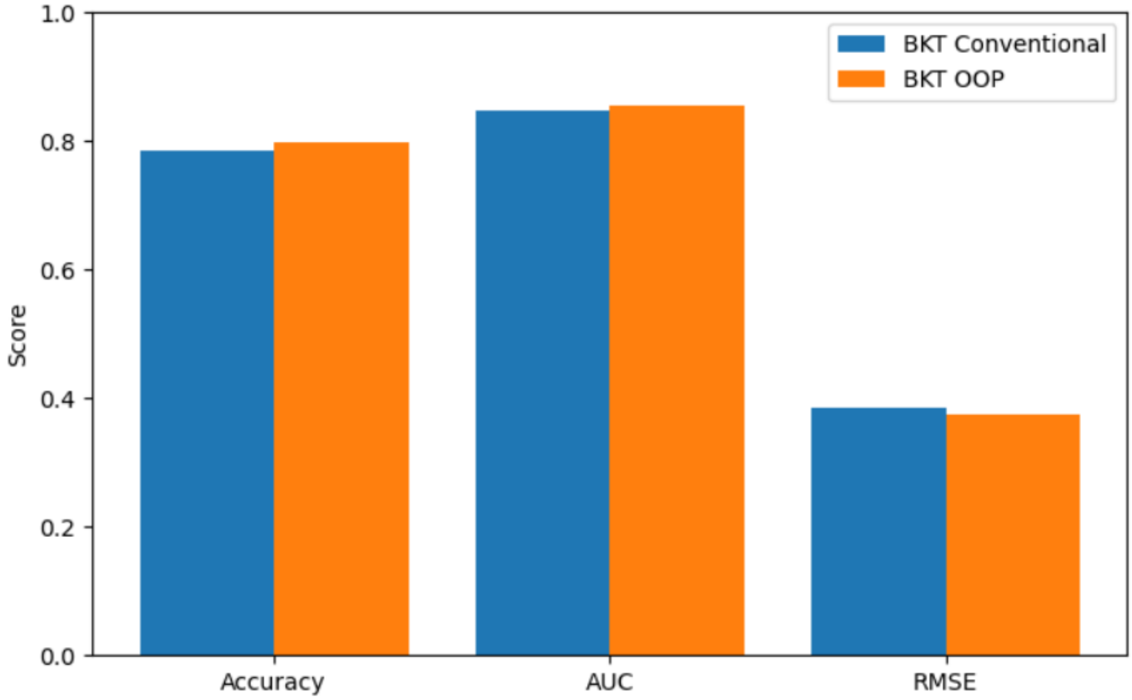


**Fig 3.** Average Performance Comparison: BKT Conventional vs BKT OOP.

The results of this study demonstrate that the OOP-based implementation of Bayesian Knowledge Tracing (BKT) consistently outperforms the conventional approach across all evaluation metrics. Specifically, the model achieves an approximate 1.53% increase in accuracy and a 1.06% improvement in AUC, indicating enhanced predictive precision and improved discrimination between correct and incorrect responses. In addition, the 2.85% reduction in Root Mean Square Error (RMSE) confirms a decrease in prediction error, suggesting improved model reliability. It is important to emphasize that these performance gains are not attributed to

INTENSIF, Vol.10 No.2 2026
**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

modifications in the mathematical formulation of BKT, but rather to improvements in the implementation structure. The modular and encapsulated design enabled by the OOP paradigm supports more systematic parameter management, consistent state updating, and reduced implementation variability. This leads to more stable model behavior and indirectly enhances predictive performance. These findings are in line with previous studies that highlight the importance of structured system design in improving model reliability and reproducibility in data-driven learning systems. From a practical perspective, the improved predictive performance allows adaptive learning systems to more accurately estimate student knowledge states and generate more relevant learning recommendations, thereby enhancing learning effectiveness. Although the numerical improvements are relatively modest, they are consistently observed across all evaluation scenarios, indicating stable and reliable performance gains. Although statistical significance testing was not conducted in this study, the observed improvements were consistently evident across all cross-validation folds. This consistency suggests that the performance gains are stable and not the result of random variation. Nevertheless, future work will incorporate formal statistical significance tests, such as paired t-tests, to further validate the robustness of the proposed approach. However, this study has several limitations. The experiments were conducted on a single dataset with limited parameter tuning, which may affect the generalizability of the results. Future research should explore more diverse datasets and incorporate advanced parameter optimization techniques to further validate and extend the proposed approach.

## IV. CONCLUSION

This study compares the performance of conventional Bayesian Knowledge Tracing (BKT) with Object-Oriented Programming (OOP) based BKT in the scenario of predicting student knowledge. Experimental results show that the OOP approach provides significant performance improvements. This improvement indicates that the implementation of BKT within the OOP framework is capable of producing more accurate, consistent, and efficient predictions compared to conventional methods. These advantages can be attributed to the modularity, encapsulation, and ease of code maintenance in the OOP paradigm, which allows for more systematic management of model parameters. The practical implication of these findings is that the development of adaptive learning systems based on BKT can leverage OOP to improve development speed and prediction quality. For future research, testing on more diverse datasets and more comprehensive exploration of parameter optimization is recommended to ensure the generalizability of the results. In addition, the integration of OOP-based BKT with other machine learning models can be a potential research direction to improve the system's adaptability to

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

complex and dynamic learner characteristics. Overall, this study confirms that the application of the OOP paradigm in the development of BKT is not only relevant for improving model performance, but also for building a more sustainable, scalable, and easily integrated software architecture foundation for future adaptive learning systems.

**Author Contributions:** *Miftah Farid Adiwisastra*: Conceptualization, Methodology, Writing - Original Draft, Software, Writing-Review & Editing, Supervision. *Yani Sri Mulyani*: Investigation, Data Curation, Writing-Original Draft. *Yanti Apriyani*: Investigation, Data Curation.

All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no specific grant from any funding agency

**Conflicts of Interest:** The authors declare no conflict of interest.

**Data Availability:** Data is available on github
https://github.com/miftahadiwisastra/bkt/blob/main/student_data_.csv

**Informed Consent:** There were no human subjects

**Animal Subjects:** There were no animal subjects.

**ORCID**:
Miftah Farid Adiwisastra: https://orcid.org/0000-0003-2762-6823
Yani Sri Mulyani: https://orcid.org/0000-0002-5412-3941
Yanti Apriyani: https://orcid.org/0009-0000-5145-6727

## REFERENCES

[1] W. Strielkowski, V. Grebennikova, A. Lisovskiy, G. Rakhimova, and T. Vasileva, “AI-driven adaptive learning for sustainable educational transformation,” *Sustainable Development*, vol. 33, no. 2, pp. 1921–1947, Apr. 2025, doi: 10.1002/SD.3221.

[2] S. G. Essa, T. Celik, and N. E. Human-Hendricks, “Personalized Adaptive Learning Technologies Based on Machine Learning Techniques to Identify Learning Styles: A Systematic Literature Review,” *IEEE Access*, vol. 11, pp. 48392–48409, 2023, doi: 10.1109/ACCESS.2023.3276439.

[3] M. Liu and D. Yu, “Towards intelligent E-learning systems,” *Educ. Inf. Technol. (Dordr).*, vol. 28, no. 7, pp. 7845–7876, Jul. 2023, doi: 10.1007/S10639-022-11479-6/METRICS.

[4] E. G. Rincon-Flores *et al.*, “Improving the learning-teaching process through adaptive learning strategy,” *Smart Learning Environments*, vol. 11, no. 1, pp. 27-, Dec. 2024, doi: 10.1186/S40561-024-00314-9/FIGURES/9.

[5] D. Otoo-Arthur and T. L. van Zyl, “A scalable heterogeneous big data framework for e-learning systems,” in *2020 international conference on artificial intelligence, big data, computing and data communication systems (icABCD)*, 2020, pp. 1–15.

[6] S. Pu, G. Converse, and Y. Huang, “Deep performance factors analysis for knowledge tracing,” in *International Conference on Artificial Intelligence in Education*, 2021, pp. 331–341.

[7] W. Wang, T. Liu, L. Chang, T. Gu, and X. Zhao, "Convolutional recurrent neural networks for knowledge tracing," in *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2020, pp. 287–290.

[8] D. Shimada and K. Okada, "Reliability coefficient for Bayesian knowledge tracing models," *Technology, Knowledge and Learning*, pp. 1–20, 2025.

[9] B. J. Reiser, "Why Scaffolding Should Sometimes Make Tasks More Difficult for Learners," *Computer Support for Collaborative Learning*, pp. 255–264, Jan. 2023, doi: 10.4324/9781315045467-37.

[10] Š. G. Ines, G. Ani, and G. Angelina, "Twenty-Five Years of Bayesian knowledge tracing: a systematic review," *User Model. User-adapt. Interact.*, vol. 34, no. 4, pp. 1127–1173, Sep. 2024, doi: 10.1007/S11257-023-09389-4/TABLES/12.

[11] T. Lei, Y. Yan, and B. Zhang, "An Improved Bayesian Knowledge Tracking Model for Intelligent Teaching Quality Evaluation in Digital Media," *IEEE Access*, vol. 12, pp. 125223–125234, 2024, doi: 10.1109/ACCESS.2024.3452272.

[12] A. Koti, S. L. Koti, A. Khare, and P. Khare, "Beyond the Paradigm: Unraveling the Limitations of Object-Oriented Programming," *Multifaceted Approaches for Data Acquisition Processing and Communication*, pp. 95–103, Jan. 2024, doi: 10.1201/9781003470939-13/BEYOND-PARADIGM-UNRAVELING-LIMITATIONS-OBJECT-ORIENTED-PROGRAMMING-ABHISHEK-KOTI-SRI-LASYA-KOTI-AKHIL-KHARE-PALLAVI-KHARE.

[13] D. Nikolaeva, "Exploring the Basic Principles of Functional and Object-Oriented Programming," *2025 34th International Scientific Conference Electronics, ET 2025 - Proceedings*, 2025, doi: 10.1109/ET66806.2025.11204033.

[14] J. Whitman, M. Travers, and H. Choset, "Learning Modular Robot Control Policies," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 4095–4113, Oct. 2023, doi: 10.1109/TRO.2023.3284362.

[15] Z. Li, L. Shi, J. Wang, A. I. Cristea, and Y. Zhou, "Sim-GAIL: A generative adversarial imitation learning approach of student modelling for intelligent tutoring systems," *Neural Comput. Appl.*, vol. 35, no. 34, pp. 24369–24388, Dec. 2023, doi: 10.1007/S00521-023-08989-W/FIGURES/13.

[16] Y. Lei, L. Zhang, and Z. Zhang, "A Data-Driven Model and Intelligent Recommendation Method for Individualized Teaching in Engineering Cost Courses," *Proceedings of 2025 2nd International Conference on* Artificial *Intelligence and Future Education, AIFE 2025*, vol. 1, pp. 541–546, Jan. 2026, doi: 10.1145/3785987.3786078/ASSET/0C1CB792-1ACA-4E2B-8508-A4EE0F160430/ASSETS/IMAGES/LARGE/IMAGE16.JPG.

[17] L. Zhang, P. I. Pavlik, X. Hu, J. L. Cockroft, L. Wang, and G. Shi, "Exploring the Individual Differences in Multidimensional Evolution of Knowledge States of Learners," *Lecture Notes in Computer* Science *(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14044 LNCS, pp. 265–284, 2023, doi: 10.1007/978-3-031-34735-1_19/SAVE-RESEARCH.

[18] X. Zhou, Z. Zhang, X. Xie, and J. Zhang, "Deep learning based knowledge tracing in intelligent tutoring systems," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 21395-, Jul. 2025, doi: 10.1038/s41598-025-07422-7.

[19] M. M. Kuo, X. Li, P. H. Obiomon, L. Qian, and X. Dong, "Tracing Student Learning Outcome at Historically Black Colleges and Universities via Deep Knowledge Tracing," *IEEE Access*, vol. 13, pp. 61340–61349, 2025, doi: 10.1109/ACCESS.2025.3557316.

[20] O. Bulut, J. Shin, S. N. Yildirim-Erbasli, G. Gorgun, and Z. A. Pardos, "An Introduction to Bayesian Knowledge Tracing with pyBKT," *Psych 2023, Vol. 5, Pages 770-786*, vol. 5, no. 3, pp. 770–786, Jul. 2023, doi: 10.3390/PSYCH5030050.

**ISSN: 2580-409X (Print) / 2549-6824 (Online)**
**DOI:** https://doi.org/10.29407/intensif.v10i2.27174

[21] R. Subha, N. Gayathri, S. Sasireka, R. Sathiyabanu, B. Santhiyaa, and B. Varshini, "Intelligent Tutoring Systems using Long Short-Term Memory Networks and Bayesian Knowledge Tracing," *Proceedings - 2024* 5th *International Conference on Mobile Computing and Sustainable Informatics, ICMCSI 2024*, pp. 24–29, 2024, doi: 10.1109/ICMCSI61536.2024.00010.

[22] N. T. Heffernan and C. L. Heffernan, "The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching," *Int. J. Artif. Intell. Educ.*, vol. 24, no. 4, pp. 470–497, 2014.

[23] Z. Liu *et al.*, "Deep Learning Based Knowledge Tracing: A Review, A Tool and Empirical Studies," *IEEE Trans. Knowl. Data Eng.*, 2025.

[24] Z. Liu *et al.*, "Enhancing deep knowledge tracing with auxiliary tasks," in *Proceedings of the ACM web conference 2023*, 2023, pp. 4178–4187.

[25] C. Fischer *et al.*, "Mining big data in education: Affordances and challenges," *Review of research in education*, vol. 44, no. 1, pp. 130–160, 2020.

[26] J. Barrett, A. Day, and K. Gal, "Improving model fairness with time-augmented Bayesian knowledge tracing," in *Proceedings of the 14th Learning Analytics and Knowledge Conference*, 2024, pp. 46–54.

[27] A. Badrinath and Z. Pardos, "Optimizing Bayesian Knowledge Tracing with Neural Network Parameter Generation.," *Journal of Educational Data Mining*, vol. 17, no. 1, pp. 41–65, 2025.

[28] C. Papakostas, C. Troussas, A. Krouska, and C. Sgouropoulou, "Integrating Bayesian Knowledge Tracing and Human Plausible Reasoning in an Adaptive Augmented Reality System for Spatial Skill Development," *Information*, vol. 16, no. 6, p. 429, 2025.

[29] P. Modi, "Design and implementation of a cross-platform Student Management System app using OOP principles," 2025.

[30] J. Vijaya, A. Kulkarni, V. V. Ranjan, and V. Bajaj, "An Enhanced Object-Oriented Programming-Based Web Page Linker," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, 2024, pp. 1–6.

[31] T. Yadav, R. Punhani, and S. Hans, "Enhancing Healthcare Applications with Object-Oriented Generative Adversarial Networks," *Procedia Comput. Sci.*, vol. 259, pp. 269–278, 2025.

[32] D. Cai, Y. Zhang, and B. Dai, "Learning Path Recommendation Based on Knowledge Tracing Model and Reinforcement Learning," *2019 IEEE 5th International Conference on Computer and Communications, ICCC 2019*, pp. 1881–1885, Dec. 2019, doi: 10.1109/ICCC47050.2019.9064104.

[33] S. Minn, "BKT-LSTM: Efficient Student Modeling for knowledge tracing and student performance prediction," *arXiv preprint arXiv:2012.12218*, 2020.