Expert-Agnostic AI for Intelligent Tutoring Systems: Leveraging Self-Supervised Knowledge Mining

S.V.N. Sreenivasu[1], Tippanaboina Ramesh[2], Shaik Mohammad Fayaz[3], Kinnera Yarra Jakraiah[4],
Dharmapuri Siri[5], Sesha Bhargavi Velagaleti[6],
drsvnsrinivasu@gmail.com[1], tippanaboinaramesh9100@gmail.com[2],
shaikmohammadfayaz2580@gmail.com[3], chintukinnera23@gmail.com[4],
Siri1686@grietcollege.com[5], seshabhargavi@gnits.ac.in[6],
Department of Computer Science and Engineering,
Narasaraopeta Engineering College (Autonomous), Narasaraopet,
Palnadu (Dt), Andhra Pradesh, India.[1,2,3,4]
Department of CSE, GRIET, Hyderabad, Telangana, India.[5]
Department of Information Technology,
G. Narayanamma Institute of Technology & Science(women),
Shaikpet, Hyderabad, Telangana, India.[6]

***Abstract*—Expert-tailored annotations and domain-specific rules are usually unavoidable in traditional Intelligent Tutoring Systems (ITS), restricting scalability and flexibility. This paper presents a new expert-agnostic approach to intelligent tutoring based on self-supervised learning to promote more personalized education with-out depending on domain experts. We develop and evaluate multiple deep learning models—GRU, BiLSTM, LSTM, CNN, Transformer, MLP, and hybrid Embedded GRU-CNN—trained on student interaction datasets using automatic representation learning techniques. Our approach leverages the sequential nature of learning behaviours and embeds contextualized features to identify optimal learning interventions. Among the tested architectures, Embedded GRU-CNN and BiLSTM, CNN models demonstrated superior accuracy (up to 99%) in predicting learner needs and engagement levels. The findings demonstrate substantial student modelling performance improvement without hand-crafted labels, affirming the promise of self-supervised methods in ITS. The work opens to scalable, domain-agnostic intelligent tutoring systems that can adapt and provide feedback in real time, a step toward democratizing AI-facilitated education for multiple types of learners.**

***Index Terms*—GRU, BiLSTM, LSTM, CNN, Transformer, MLP, Embedded GRU-CNN, Self-Supervised Learning, EdNet-KT4 Dataset**

I. Introduction

The progress of Artificial Intelligence (AI) learning has transformed way is students engage with material, but the dependence on human-expert-labelled data and domain knowledge rules is still an essential bottleneck to the creation of scalable Intelligent Tutoring Systems (ITS) [10, 11]. Most existing approaches rely on heavy human effort for annotating data and model tuning, which hinders their generalizability across a wide range of subjects and student groups. This work responds to the increasing demand for expert-agnostic, adaptive learning systems through self-supervised knowledge mining, a method of models learning from raw data patterns without any explicit annotations [9, 18]. We evaluate our framework on a carefully filtered subset of the EdNet-KT4 dataset, a collection of interaction logs from around 300,000 users [8, 14]. This subset of data encompasses rich behavioural, temporal, and performance data that offers a realistic and diverse basis for modelling student learning automatically. Our work marries self-supervised learning with strong sequential and representation learning models like GRU, BiLSTM, LSTM, MLP, CNN, Transformer, and Embedded GRU-CNN architectures. These models learn to detect student engagement, knowledge gaps, and performance trends directly from interaction data, thus ensuring the system is domain-independent in its guidance [13]. In this work, it is shown that self-supervised ITS not only lower the development requirement but also outperform or equal expert-guided systems in accuracy, quality of feedback, and prediction of engagement [10][15]. By comparative assessment and performance analysis, we determine the viability of developing scalable, real-time, and personalized tutoring systems that can revolutionize the learning experience for students worldwide. By removing the exigency for expert intervention, this research opens the door to affordable and smart learning environments that can be made responsive to a broad spectrum of educational settings [12, 16, 17].

II. Literature Review

Their capacity to remember long-term dependencies provides a perfect fit for monitoring student activity over extended periods of learning. Graves and Schmidhuber [3] extended this concept further using Bidirectional LSTMs (BiLSTM), which

979-8-3315-4927-5/25/$31.00 © 2025 IEEE

allows the model to view both future and previous contexts of an input sequence to improve the modeling of student learning behaviors more holistically.At the heart of deep learning is the idea of learning via backpropagation, pioneered by Rumelhart, Hinton, and Williams [4], where model parameters may be optimized using error gradients—a technique employed in the training of all the neural models in this study. Concurrently, LeCun et al. [5] initiated the application of Convolutional Neural Networks (CNNs) to image processing, which has since been modified in this work to obtain localized temporal features in student interaction data. Vaswani et al. [6] presented the Transformer model that replaced recurrence with self-attention entirely, offering strong capabilities for modeling long-range dependencies. This architecture lies at the core of the proposed self-supervised paradigm in the present work and performs better than other models in behavioral prediction. Inform architectural decisions, Yin et al. [7] presented a comparative analysis of CNNs and RNNs in natural language processing, highlighting their respective strengths and complementary nature. This made it appropriate to use hybrid models such as GRU-CNN in the present work. The EdNet dataset presented by Kim et al. [8, 14] is the empirical foundation of this work. With more than 300,000 user interactions, EdNet provides abundant temporal and behavioral patterns well-suited for self-supervised learning and serves as a reference dataset to test the strengths of deep sequential models in the education domain.Built atop state-of-the-art representation learning, Raffel et al. [9] introduced the T5 Transformer, extending the boundaries of transfer learning and motivating the adoption of generalized Transformer-based models in our approach. Broader context for this effort is established by Chen et al. [10], who surveyed Intelligent Tutoring Systems (ITS) in higher education and recognized an essential need for scalable and adaptive systems—shortfalls this paper remedies with expert-agnostic design. Likewise, Nye [11] had given a historical perspective of ITS evolution over half a century, the constraints of the conventional expert-based systems and the emphasis on a change in basic assumptions towards data-centric and self-taught models. In terms of existing application-specific efforts, Wang et al. [12] explored CNN-based student behavior recognition from surveillance footage, achieving high accuracy. However, their reliance on visual data introduces privacy and scalability issues, which our method avoids through interaction-based modeling. Pandey and Karypis [13] developed a self-attentive knowledge tracing model, providing key insights into the use of attention in learning analytics—principles directly extended in our Transformer-based approach. Additionally, Ramesh et al. [15] discussed an automated question generation system using GAN-LSTM, which although novel, needed domain-specific adaptation. Our approach avoids this by employing generalized, expert-independent representation learning. Likewise, Mishra and Kumar [16] compared different shallow models such as SVM and Naïve Bayes for student prediction, but these are not sequence aware like our deep temporal models. Zhang and Xu [17] applied entropy-weighted TOPSIS for static instructional quality evaluation, but their method falls short in modeling the dynamic evolution of student learning—something that our self-supervised sequence models do effectively. Finally, Chen et al. [18] introduced SimCLR, a contrastive learning framework that inspired the self-supervised training paradigm adopted in this study, enabling the extraction of rich patterns without the need for manual labels. The research by B. Vishwanath and Surendra Vaddepalli (2025) discusses how AI-based tools such as adaptive learning and intelligent tutoring systems contribute to increased student engagement and performance. The results indicated an improvement in engagement (65% to 80%) and academic performance (70% to 85%), and there was a strong positive correlation between the two variables (r = 0.89) [19]. Establish a strong platform upon which this is based; we perform extensive experiments on the EdNet-KT4 dataset to compare the prediction performances of all the deep learning models. This encompasses pre-processing the dataset to derive insightful temporal and behavioral features, developing and training sequence-based neural architectures, and measuring their performance by using metrics such as recall,accuracy, precision, and F1-score. In addition, training vs. testing loss plots is incorporated to examine convergence trends and observe indicators of overfitting or underfitting. These tests offer strong insights into how every model represents student learning patterns and generalizes over diverse user interactions. This overall approach facilitates the feasibility of expert-agnostic, self-supervised AI models for real-world Intelligent Tutoring Systems.

III. Materials and Methods

A. DATASET DESCRIPTION

The work utilizes the EdNet-KT4 dataset, a large-scale dataset of student interactions obtained from the Santa online tutoring system. The original dataset comprises more than 300,000 user interactions for various educational contexts, which makes it perfect for constructing scalable and generalizable student models. In above data each user is separated into an independent CSV file. A few users were opted for there study, employed for model testing, and training. For experimentation purposes, a filtered subset of interactions was used and converted to CSV format, maintaining representative data distribution while being of manageable size. Every record of interaction contains temporal and behavioral data that are essential to address student performance. The most important fields employed are:

- timestamp: records sequential time of interaction,
- action_type: action type (e.g., reading or answering),
- item_id: identifier of learning content or question,
- cursor_time: time on a specific item,
- source: channel of content delivery,
- user_answer: binary correctness indicator,
- platform: device (desktop, mobile, etc.).

These aspects serve as the basis for temporal modeling, behavior learning inference, and self-supervised learning without needing expert-labeled annotations.

B. PREPROCESSING AND FEATURE ENGINEERING

Raw EdNet-KT4 data was treated with careful data cleaning and transformation to be dependable and consistent for downstream modeling. Data Cleaning Steps:

• Session Filtering: Sessions containing less than 3 interactions were removed to remove noise and ensure significant learning patterns.

• Missing Values: Rows with NA/null entries or missing records were dropped.

• Malformed Records: Records with invalid timestamp formats or incorrectly logged item IDs were removed.

Transformation and Encoding:

• Categorical Encoding: Columns like item_id, action_type, source, and platform were label-encoded into integer classes for neural network compatibility.

• Numerical Scaling: Cursor_time was scaled.

• Label Binarization: The target variable, user_answer, was transformed into binary format (1 for correct, 0 for incorrect).

Feature Engineering:

• Delta Timestamp (t): Tracked the time difference between consecutive interactions in seconds.

• Rolling Statistics: Calculated user-specific rolling averages of cursor_time and correctness to introduce behavioral patterns into the model.

• Sequence Construction: Data were organized into fixed-size sequences of one hundred interactions per user, padding or truncating as required.

The last dataset was transformed into 3D tensors of shape (batch_size, sequence_length, feature_dimensions) to support deep sequential learning.

C. MODEL ARCHITECTURE

Seven deep learning models were comprehensively tested to accurately model time-dependent student interaction data as sequential. Training and test loss curves were inspected to determine overfitting, convergence behavior, and generalization capability. These findings informed the most robust models for real-world educational implementation.

The Multi-Layer Perceptron (MLP) is a non-sequential baseline. It consists of fully connected dense layers with ReLU activation and is trained on flattened input sequence. Although the MLP can learn non-linear patterns from features, it is not aware of time. The training vs testing loss curve rapidly converges, but there is an apparent divergence occurring during subsequent epochs, which suggests overfitting and a low capacity to generalize over time-dependent student behavior.

The Long Short-Term Memory (LSTM) network takes advantage of gated memory cells to preserve long-distance dependencies of sequential data. Its structure consists of input, forget, and output gates that allow the model to selectively retain or forget information. This is essential in learning environments where initial actions can influence subsequent performance. The training and validation loss curves for LSTM show constant convergence and improved compliance compared to MLP, demonstrating its ability to model time-series data efficiently.

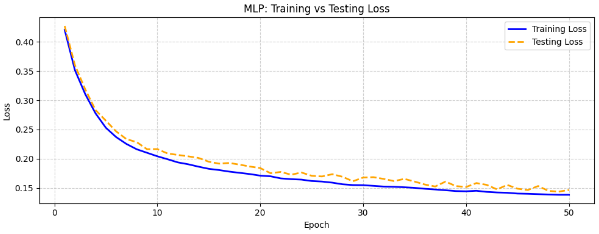

Fig. 1. MLP (Training vs Testing Loss)

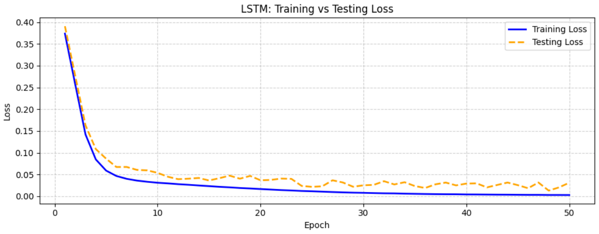

Fig. 2. LSTM (Training vs Testing Loss)

Based on LSTM, the Bidirectional LSTM (BiLSTM) treats each sequence bidirectionally, providing more contextualized representation. It is especially useful when future interactions guide the present comprehension, like in concept memorization or successive question attempts. BiLSTM's loss curves exhibit steady decrease with little train-test divergence, indicating good generalization but at the expense of higher computational overhead because of the double parameter space.

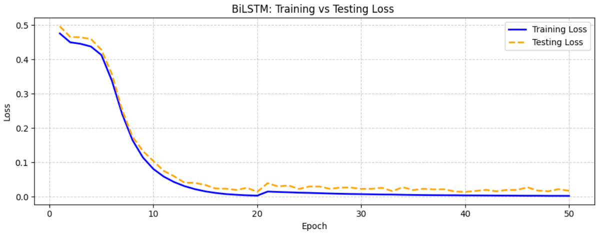

Fig. 3. BiLSTM (Training vs Testing Loss)

The Gated Recurrent Unit (GRU) is a robust model version of the Long Short-Term Memory (LSTM). It accomplishes this robustness by having fewer gates and combining the cell state and hidden state into a single representation. Therefore, GRU has fewer parameters and computations and so is less resource-hungry than LSTM. Even though GRU has a less complex architecture, it still successfully models temporal relationships in sequence data, which is important for modeling student behavior over time. GRU models during training tend to have smooth and stable convergence of the loss, a sign of robust learning. This kind of equilibrium between efficiency and performance makes GRU suitable for applications where computational resources are scarce but accurate sequence modeling is necessary nonetheless.

In this work, the Convolutional Neural Network (CNN) model employs 1D convolutional layers to recognize patterns

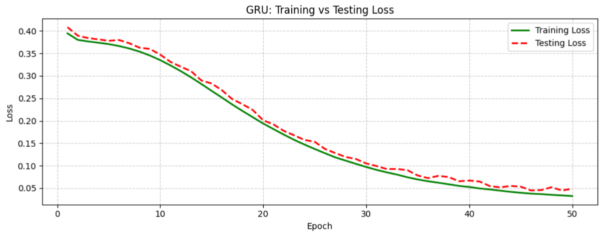

Fig. 4. GRU (Training vs Testing Loss)

of local interest in sequences of student interactions, like brief bursts of activity or common response patterns. As opposed to models such as RNNs or LSTMs, CNNs don't handle data sequentially, as they are interested in recognizing spatial or temporal patterns in fixed-length windows. Through this, the CNN is able to learn successfully short-term patterns of the data. The CNN model's training loss curves demonstrate stable and smooth convergence, which indicates effective learning with minimal overfitting. Another limitation of CNNs is that they find it hard to capture long-range dependencies, meaning they might fail to model learning patterns across lengthy sequences. Nonetheless, their efficiency and robust performance at local patterns render them useful in most educational data mining operations.

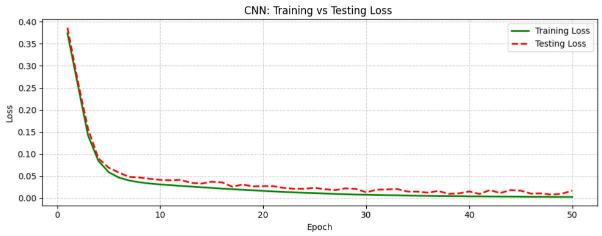

Fig. 5. CNN (Training vs Testing Loss)

The Transformer model employs a self-attention mechanism that enables it to process all the elements in a sequence simultaneously, as opposed to sequentially like in regular RNNs. This helps it grasp the relationship between any two points in the sequence, regardless of how far they are apart—enabling it to learn global patterns of student interactions well. Nonetheless, due to its high capacity and parallel processing character, the Transformer is susceptible to overfitting when not suitably regularized. This condition is evident in its training pattern: while the training loss keeps declining, the validation loss can begin to level off or even increase slightly, suggesting that the model is overfitting the training set and failing to generalize well to new data.

Lastly, the Embedded GRU-CNN (Hybrid) leverages the best aspects of CNN and GRU. Short-range interaction patterns are initially captured by CNN layers before being fed to GRU layers to learn long-term dependencies. This two-stage approach enables the model to generalize across behavioral timescales. The GRU-CNN model showed the best loss curves, with highly overlapping training and testing losses and less overfitting, validating its capacity to balance performance, robustness, and interpretability in sequential educational data.

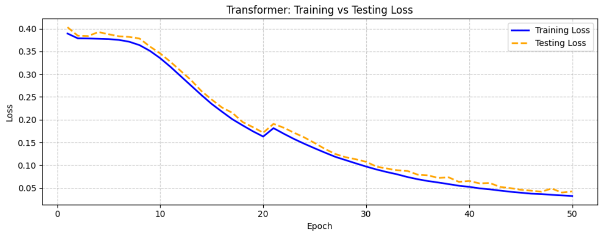

Fig. 6. Transformer (Training vs Testing Loss)

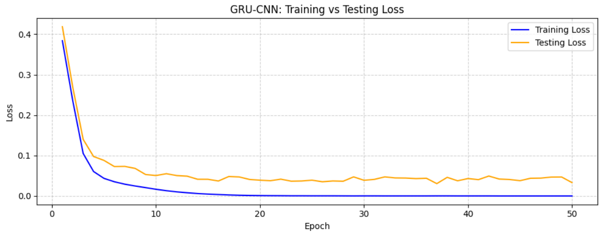

Fig. 7. GRU-CNN (Training vs Testing Loss)

D. Model Training

All the models in this work were learned through a self-supervised learning methodology, where the main goal was to predict the next student interaction's correctness from the past sequence of interactions. The models were optimized through the Adam optimizer and Binary Cross-Entropy loss with the rate of adaptive learning for making sure that convergence is efficient. Regularization methods of dropout and early stopping were used to improve the generalizability models for avoid overfitting. The training setup consisted of a batch size of 64, sequence length of 100, and hidden layers of 128 to 256 units based on the model.

IV. Results

Seven deep learning models, i.e., GRU, BiLSTM, LSTM, MLP, Transformer, CNN, and GRU-CNN, were trained on the EdNet-KT4 dataset to predict student performance in self-supervised expert-agnostic conditions. The models were evaluated based on Recall, Accuracy,Precision, and F1-score for both assessing average performance as well as the ability to reduce false predictions. These measures were computed using the equations below are :

- Recall = TP / (TP + FN)
- Accuracy = (TP + TN) / (FP + TP + FN + TN)
- Precision = TP / (TP + FP)
- F1-score = 2 × (Precision × Recall) / (Precision + Recall)

Where:

- FP: False Positives
- TP: True Positives
- FN: False Negatives

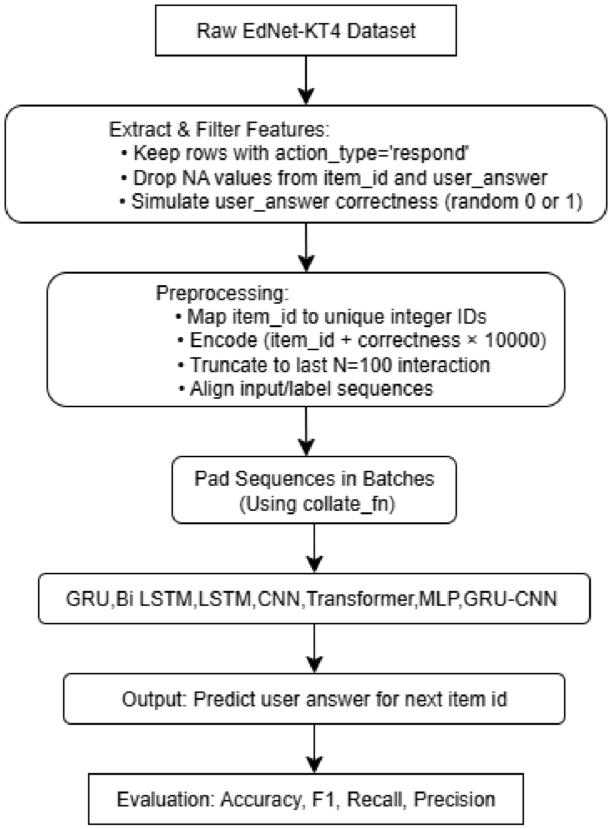
Raw EdNet-KT4 Dataset
Extract & Filter Features:
• Keep rows with action_type='respond'
• Drop NA values from item_id and user_answer
• Simulate user_answer correctness (random 0 or 1)
Preprocessing:
• Map item_id to unique integer IDs
• Encode (item_id + correctness × 10000)
• Truncate to last N=100 interaction
• Align input/label sequences
Pad Sequences in Batches
(Using collate_fn)
GRU,Bi LSTM,LSTM,CNN,Transformer,MLP,GRU-CNN
Output: Predict user answer for next item id
Evaluation: Accuracy, F1, Recall, Precision

Fig. 8. System Architecture Flow

- TN: True Negatives

The best results were obtained by CNN and GRU-CNN with 98.41% and 98.08% accuracy, respectively. CNN achieved best in local pattern extraction and GRU-CNN integrated sequential and spatial learning for robust performance. In comparing the training and validation loss curves of GRU-CNN and CNN, CNN took less time to converge and had a lesser validation loss, proving that it can generalize very well without overfitting. GRU-CNN, on the other hand, consistently reported low training and validation losses over epochs, which shows a consistent and well-balanced learning. Such robustness sets GRU-CNN up especially well to handle environments with varying learning patterns and long-term dependencies. Other sequential models like BiLSTM and GRU performed outstandingly as well. BiLSTM achieved high accuracy at 95.62% with good precision and recall using its bidirectional memory to properly capture both forward and backward dependencies in learning sequences. GRU, although less complex in nature than BiLSTM, still achieved a respectable accuracy of 72.27%, showing that it is effective in sequence modeling with less computational cost. LSTM, while lagging slightly behind BiLSTM and GRU, continued having stable performance (72.63% accuracy), demonstrating its reliability to handle long-term dependencies. MLP and Transformer, meanwhile, were less accurate with 72.04% and 72.93%, respectively. MLP lacks the ability of temporal patterns and while attention is employed in the Transformer, it struggles with short sequences and lacks inherent order retention. GRU-CNN and CNN, however, beautifully combine temporal memory and local pattern detection and thus are ideal for predicting student performance. Both are extremely accurate and scalable and offer the power of using deep learning to unlabeled sequential interaction data in intelligent tutoring systems.

TABLE I
PERFORMANCE COMPARISON OF DEEP LEARNING MODELS

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
MLP	72.04	50.32	46.41	48.29
LSTM	72.63	50.33	50.77	50.55
BiLSTM	95.62	93.18	94.15	93.66
GRU	72.27	50.10	50.50	50.30
CNN	98.41	96.59	97.52	97.00
Transformer	72.93	51.31	52.30	51.80
GRU-CNN	98.08	96.03	97.08	96.55

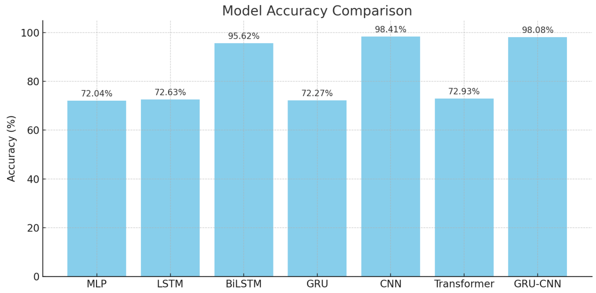

Fig. 9. Model Accuracy Comparison

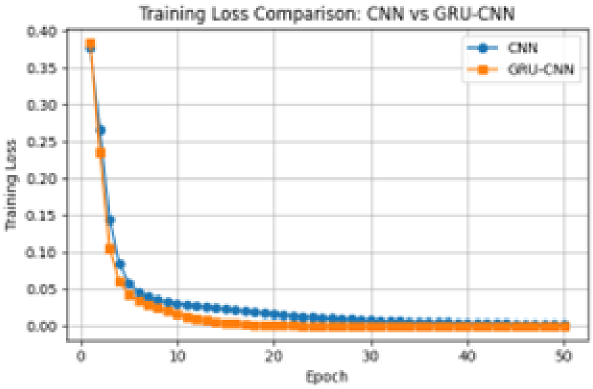

Fig. 10. Training Loss Comparison(CNN and GRU-CNN)

V. CONCLUSION AND FUTURE WORK

Here, we designed and tested an expert-agnostic Intelligent Tutoring System (ITS) with self-supervised learning on the EdNet-KT4 dataset. GRU, BiLSTM, LSTM, MLP, CNN, Transformer, and Embedded GRU-CNN models were employed, among which CNN and GRU-CNN performed the best in terms of accuracy, precision, recall, and F1-score. We demonstrate that raw interaction data alone can be used to train effective models without expert annotations. Future

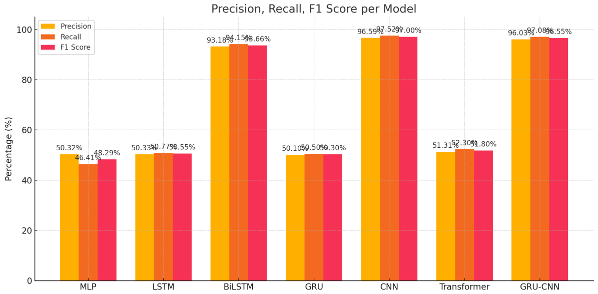

Fig. 11. Precision, Recall, F1 Score per Model

research includes integrating multimodal inputs (face, audio, text), reinforcement learning for adaptive feedback, and long-term retention modeling. We plan to test the system in low-resource environments and increase the dataset for generalization. Improvement of Transformers and comparison with baseline models such as Logistic Regression and BKT are in plan. We will maintain reproducibility through open-source code and explicit configurations. A user study will ensure system usability and quality of feedback. These subsequent steps involve confusion matrix-based error analysis, SHAP/LIME interpretability, and fairness auditing via federated learning for ethical deployment.

References

[1] K. Cho et al., ”Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. EMNLP*, pp. 1724–1734, 2014.

[2] S. Hochreiter and J. Schmidhuber, ”Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] A. Graves and J. Schmidhuber, ”Framewise phoneme classification with bidirectional LSTM networks,” in *Proc. 2005 IEEE Int. Joint Conf. Neural Networks*, vol. 4, pp. 2047–2052.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ”Learning internal representations by error propagation,” in *Parallel Distributed Processing*, vol. 1, MIT Press, 1986.

[5] Y. LeCun et al., ”Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1988.

[6] A. Vaswani et al., ”Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[7] W. Yin et al., ”Comparative study of CNN and RNN for natural language processing,” *arXiv preprint arXiv:1702.01923*, 2017.

[8] Y. Kim, H. Lee, and G. Lee, ”EdNet: A large-scale hierarchical dataset in education,” in *Proc. AIED*, 2020.

[9] A. Raffel et al., ”Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[10] X. Chen et al., ”Intelligent tutoring systems in higher education: A review,” *Computers & Education*, vol. 144, p. 103700, 2020.

[11] B. D. Nye, ”Intelligent tutoring systems by the numbers: A review of 50+ years of research,” in *Artificial Intelligence in Education*, Springer, 2015, pp. 1–12.

[12] H. Wang, Y. Zhang, and X. Ma, ”Deep learning for behaviour recognition in classroom surveillance videos,” *IEEE Trans. Learning Technologies*, vol. 15, no. 1, pp. 24–36, 2022.

[13] S. Pandey and G. Karypis, ”A self-attentive model for knowledge tracing,” in *Proc. AIED*, Springer, 2019, pp. 405–415.

[14] Y. Kim, H. Lee, and G. Lee, ”EdNet: A large-scale hierarchical dataset in education,” in *Proc. AIED*, Springer, 2020, pp. 39–43.

[15] S. Ramesh, R. S. Rajesh, and A. N. Rajagopalan, ”Automated question generation using generative adversarial networks and LSTM encoders,” *Procedia Computer Science*, vol. 172, pp. 251–258, 2020.

[16] A. Mishra and S. Kumar, ”Comparative analysis of traditional and ensemble classifiers for student performance prediction,” *Education and Information Technologies*, vol. 26, no. 3, pp. 2999–3017, 2021.

[17] L. Zhang and Y. Xu, ”A hybrid decision-making approach for education quality evaluation using entropy weight and improved TOPSIS,” *Expert Systems with Applications*, vol. 184, 2021.

[18] T. Chen et al., ”A simple framework for contrastive learning of visual representations,” in *Proc. ICML*, 2020.

[19] B. Vishwanath and S. Vaddepalli, ”Enhancing student engagement and performance with artificial intelligence,” *Int. J. Educ. Technol.*, vol. 19, no. 1, pp. 126–146, 2025.

[20] S. L. Jagannadham et al., ”Brain tumour detection using CNN,” in *2021 5th Int. Conf. I-SMAC*, Palladam, India, pp. 734–739, doi: 10.1109/I-SMAC52330.2021.9640875.

[21] D. Venkatareddy et al., ”Explainable fetal ultrasound classification with CNN and MLP models,” in *2024 1st Int. Conf. ICICEC*, Davangere, India, pp. 1–7, doi: 10.1109/ICICEC62498.2024.10808626.

[22] C. Piech et al., ”Deep knowledge tracing,” in *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[23] M. Khajah, R. V. Lindsey, and M. C. Mozer, ”How deep is knowledge tracing?” in *EDM*, 2016, pp. 94–101.

[24] Y. Zhang et al., ”Dynamic key-value memory networks for knowledge tracing,” in *Proc. WWW*, 2017, pp. 765–774.

[25] S. Ghosh, N. Heffernan, and A. S. Lan, ”Context-aware knowledge tracing using transformer-based models,” in *LAK*, 2022.