Neural Adaptive Learning Framework Based on Behavioral Sequences and Latent Cognitive States for Personalized Digital Instruction

Khabib Adi Nugroho[1,*], Turino[2]

[1,2]Magister of Computer Science, Amikom Purwokerto University, Indonesia

ABSTRACT

This study proposes a Neural Adaptive Learning Framework that integrates deep behavioral sequence modeling with latent cognitive state estimation to generate personalized instructional interventions in large-scale digital learning environments. Using a dataset comprising 9,000 learner behavioral sequences collected from an online learning platform, the framework employs a Bi-LSTM/Transformer encoder to model temporal dependencies across event logs and a latent-state inference module to estimate mastery, engagement, and cognitive load. Experimental results demonstrate that the model achieves an AUC of 0.89, outperforming a Random Forest baseline (0.76), a rule-based adaptive system (0.71), and a non-adaptive LMS (0.64). The adaptive policy also yields a 78% action accuracy, a substantial improvement over rule-based policies (55%) and non-adaptive sequencing (41%). In terms of educational impact, the framework leads to a 46% higher learning gain compared to the non-adaptive condition (0.41 vs. 0.28) and reduces time to mastery by 26% (9.1 vs. 12.4 sessions). Overall, the findings confirm that combining behavioral sequence encoding, latent cognitive inference, and neural policy optimization yields a robust, cognitively informed adaptive learning system. The proposed framework significantly enhances predictive accuracy, decision quality, and learning outcomes, offering a scalable and generalizable approach for next-generation personalized e-learning platforms.

Keywords Neural Adaptive Learning, Behavioral Sequence Modeling, Cognitive State Estimation, Deep Learning, Personalization, Learning Analytics, Intelligent Tutoring Systems

Introduction

The rapid expansion of digital learning platforms has resulted in an unprecedented volume of learner interaction data, offering new opportunities to design adaptive educational systems that respond dynamically to students' needs [1], [2]. Despite these advances, most contemporary Learning Management Systems (LMS) still rely on static sequencing of instructional content, providing identical learning paths to all students regardless of their background, cognitive readiness, or behavioral patterns [3], [4]. This lack of personalization contributes to mismatched instructional difficulty, reduced engagement, and suboptimal learning outcomes, especially for students who require differentiated support. As learning environments shift toward data-driven personalization, there is a pressing need for intelligent frameworks capable of interpreting complex behavioral sequences and tailoring instructional actions in real time [5].

Traditional adaptive learning systems typically depend on rule-based heuristics or shallow predictive models that fail to capture the nuanced temporal dynamics

Submitted: 28 February 2025
Accepted: 5 April 2025
Published: 1 November 2025

*Corresponding author
Khabib Adi Nugroho,
23ma41d018@students.amikompurwokerto.ac.id

Additional Information and Declarations can be found on page 310

How to cite this article: K. A. Nugroho, Turino, "Neural Adaptive Learning Framework Based on Behavioral Sequences and Latent Cognitive States for Personalized Digital Instruction," *Adapt. Learn.*, vol. 1, no. 4, pp. 292-313, 2025.

present in student behavior logs [6]. These systems treat learning interactions as isolated events rather than components of evolving behavioral sequences influenced by cognitive states such as mastery, engagement, and cognitive load [7], [8]. As a result, their decisions often oversimplify the learner experience, leading to interventions that are either poorly timed or pedagogically misaligned. The inability to model sequential dependencies limits the system's ability to anticipate cognitive breakdowns, identify emergent learning opportunities, or recognize when learners require motivational reinforcement [9].

Recent advances in deep learning, particularly in sequence modeling, offer new computational tools capable of addressing these limitations. Models such as Bi-LSTM, GRU, and Transformer architectures can extract temporal dependencies across long behavioral sequences, enabling more accurate predictions of learner performance and engagement trajectories [10], [11]. However, most research in this domain focuses solely on behavioral signals and does not incorporate latent cognitive states inferred from those behaviors. Without explicit modeling of cognitive constructs, systems risk misinterpreting behavioral patterns that may be influenced by mastery fluctuations, motivational shifts, or cognitive overload [12]. This gap highlights the need for more holistic frameworks that integrate both behavioral and cognitive factors.

Furthermore, while some adaptive systems attempt to incorporate cognitive indicators through questionnaires or explicit user feedback, such methods are intrusive, infrequent, and difficult to scale [13]. Learners seldom provide consistent self-reports, and cognitive states can change rapidly during problem-solving or content exploration activities [14]. Consequently, there is a critical research gap in developing non-intrusive, real-time cognitive state inference models that operate solely on behavioral traces. Addressing this gap would allow adaptive systems to offer interventions that are both responsive and sensitive to learners' moment-to-moment cognitive transitions [15], [16].

Given these challenges, this study proposes a Neural Adaptive Learning Framework based on Student Behavioral Sequences and Cognitive States, combining deep sequence modeling with latent-state inference to generate personalized instructional recommendations. The framework applies sequence encoders such as Bi-LSTM and Transformer to derive contextual representations of behavioral logs, followed by a cognitive state estimation module that model's mastery, engagement, and cognitive load patterns over time [17]. These inferred states support a neural decision policy that determines the most appropriate instructional action—such as providing conceptual explanations, adaptive quizzes, or motivational feedback—based on the learner's current trajectory [18]. This integrated formulation addresses the methodological gaps observed in existing adaptive systems.

The primary objective of this research is to develop and evaluate a data-driven, cognitively informed adaptive learning architecture capable of improving learner outcomes through targeted interventions. Specifically, the study aims to (1) model sequential behavioral patterns using deep encoders, (2) infer latent cognitive states using smooth state-transition dynamics, and (3) optimize an adaptive policy that aligns instructional actions with real-time learning conditions [19]. By unifying these components into a single computational pipeline, the research seeks to advance the accuracy, responsiveness, and instructional coherence of adaptive learning systems operating in large-scale digital education environments [20].

The novelty of this work lies in its joint modeling of behavioral sequences and cognitive state trajectories within a unified neural framework, enabling real-time personalization that adapts continuously to evolving learner contexts. Unlike existing models that rely solely on behavior or pre-defined rules, the proposed system integrates latent cognitive diagnostics, temporal smoothness constraints, and neural policy optimization to deliver fine-grained adaptive interventions [21], [22]. This holistic approach represents a substantive methodological advancement, offering a more human-centric and cognitively grounded direction for adaptive learning research. Through this contribution, the study fills a critical gap in the literature and provides a scalable foundation for next-generation intelligent tutoring systems [23].

Literature Review

Building upon earlier work in behavioral modeling, recent studies emphasize that learning behaviors captured in LMS logs are inherently sequential and context-dependent, rather than independent or static observations [13], [14]. Researchers have demonstrated that learners' navigation paths, timing patterns, and interaction rhythms reflect underlying cognitive processes that evolve throughout a learning session [15]. For example, prolonged dwell time combined with repeated incorrect attempts may indicate cognitive overload rather than disengagement, while rapid navigation and skipping behavior may reflect surface-level learning strategies [16], [17]. These findings suggest that adaptive systems must interpret behavior in relation to its temporal context, rather than relying on isolated indicators. Consequently, sequence-aware modeling has become a central paradigm in learning analytics research [18].

Transformer-based architectures have recently gained traction in educational modeling due to their ability to capture long-range dependencies and global context within behavioral sequences [19], [20]. Compared to recurrent architectures, Transformers enable more flexible attention mechanisms that can identify critical learning moments—such as conceptual breakdowns or sudden performance improvements—without being constrained by sequential recurrence [21]. Empirical studies report that attention-based models outperform traditional RNN-based systems in predicting student outcomes, particularly in courses with complex learning structures and heterogeneous interaction patterns [22], [23]. However, despite their representational power, most Transformer-based learning models remain focused on prediction tasks and do not translate learned representations into actionable pedagogical decisions [24].

Parallel research in cognitive modeling highlights the importance of latent-state representations for understanding learning progression [25]. Cognitive diagnostic models argue that observable behavior is merely a surface manifestation of deeper mental states, such as conceptual understanding, motivation, and mental effort [26]. While probabilistic models such as Bayesian Knowledge Tracing and Hidden Markov Models have been used to infer such states, they often rely on simplifying assumptions that limit scalability and expressiveness [27], [28]. Neural latent-variable models offer a more flexible alternative, allowing cognitive states to be inferred directly from high-dimensional behavioral inputs [29]. Nevertheless, many existing approaches infer cognitive states only for post-hoc analysis, without embedding them into adaptive control mechanisms [30].

The integration of cognitive state inference into adaptive decision-making remains a critical open challenge in the literature [31]. Studies in intelligent tutoring systems indicate that instructional effectiveness depends not only on what content is delivered, but also on when and why it is delivered in relation to the learner’s cognitive readiness [32]. Adaptive interventions that ignore cognitive load or engagement risk overwhelming learners or disengaging them through redundant explanations [33]. Educational psychology further supports this view, emphasizing that optimal learning occurs when instructional difficulty is aligned with the learner’s zone of proximal development and current motivational state [34], [35]. Despite this theoretical consensus, few computational frameworks operationalize these principles in real-time adaptive systems [36].

Recent advances in reinforcement learning have attempted to formalize instructional sequencing as a long-term optimization problem, where the system learns policies that maximize cumulative learning rewards [37]. While promising, RL-based educational systems often struggle with sparse feedback, delayed rewards, and limited interpretability—particularly in real-world educational settings [38]. Moreover, RL models typically depend on handcrafted state features or coarse performance indicators, reducing their sensitivity to nuanced cognitive and behavioral signals [39]. Hybrid approaches that combine deep sequence modeling, cognitive inference, and policy optimization have been proposed conceptually, but empirical implementations remain scarce [40].

Overall, the literature reveals a convergence toward recognizing the necessity of holistic learner modeling, yet a fragmentation persists across behavioral analytics, cognitive inference, and adaptive policy learning [41]. Existing studies tend to address these components in isolation, limiting their capacity to support fine-grained, cognitively aligned personalization. This gap motivates the present study, which synthesizes deep behavioral sequence modeling with latent cognitive state estimation and neural adaptive decision-making within a unified framework. By doing so, the proposed approach directly responds to the unresolved challenges identified across prior research and advances adaptive learning toward a more integrated, data-driven, and cognitively grounded paradigm.

Methodology

This section describes the methodological design of the Neural Adaptive Learning Framework that integrates student behavioral sequences and latent cognitive states to generate personalized instructional decisions. The framework operates on log data collected from an e-learning platform, encodes student actions into temporal representations, infers moment-to-moment cognitive states, and uses a neural decision module to adapt content, difficulty, and feedback. The methodology is structured into six main components: research design, data collection and preprocessing, behavioral sequence modeling, cognitive state estimation, neural adaptive framework architecture, and training and evaluation protocol.

Research Design

This study adopts a quantitative, experimental design with a longitudinal data collection scheme. Students interact with an online learning platform over multiple sessions, generating behavioral logs such as page views, quiz attempts, hint requests, time-on-task, and navigation patterns. These logs are transformed into behavioral sequences and used to train a neural adaptive model that predicts the next optimal learning action (content, difficulty, or feedback type) for each student. The design supports both offline evaluation (using historical data) and online A/B testing where the adaptive framework is compared with a baseline rule-based or non-adaptive system.

The research design follows four main stages: (1) data acquisition and cleaning, (2) sequence construction and feature engineering, (3) cognitive state inference and representation learning, and (4) adaptive policy learning and evaluation. The Difference between pre-intervention (non-adaptive) and post-intervention (neural adaptive) phases is evaluated through learning performance, engagement, and behavioral indicators (e.g., reduction in unnecessary task switching or excessive hint usage).

Figure 1 visualizes the end-to-end methodological pipeline of the proposed neural adaptive learning framework. The flowchart begins with the Raw Logs Collection step, where time-stamped LMS events such as content views, quiz attempts, hint requests, and forum interactions are captured. These logs are then passed to the Preprocessing & Sessionization block, where data cleaning, timestamp normalization, session splitting, encoding of categorical features, and scaling of numerical features are conducted to transform heterogeneous trace data into structured sequences.

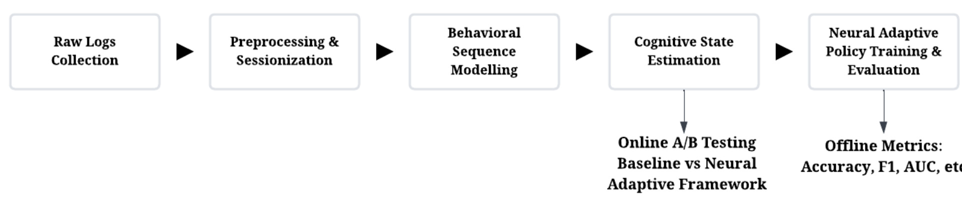

Figure 1 Overall Research Design Flow

The flow then proceeds to Behavioral Sequence Modeling, where student actions within each session are represented as temporal sequences and encoded using a sequence model (e.g., Bi-LSTM or Transformer). From these hidden representations, Cognitive State Estimation infers latent constructs such as mastery, engagement, or cognitive load. Finally, the Neural Adaptive Policy Training & Evaluation block encapsulates the learning of an adaptive decision policy that recommends the next instructional action, as well as both offline metrics (e.g., accuracy, F1, AUC) and online A/B testing against non-adaptive baselines. The auxiliary arrows highlight that evaluation is not an isolated step but is tightly coupled with policy learning and deployment.

Data Collection and Preprocessing

Behavioral data are collected from the learning management system (LMS) with time-stamped events for each student. Each event is represented as a tuple:

$$e_t = (\text{student\_id}, \text{resource\_id}, \text{action\_type}, \text{timestamp}, \text{score}_t, \text{time\_spent}_t) \quad (1)$$

where ttt indexes the event order for a given student. Action types include operations such as view_content, start_quiz, submit_quiz, request_hint, pause_session, resume_session, and forum_interaction. Only complete sessions with a minimum number of events are included to ensure that sequences are long enough to reveal behavioral structure.

Preprocessing consists of four steps: (1) log cleaning (removing duplicated or corrupted entries, normalizing timestamps), (2) session segmentation (splitting continuous logs into sessions based on inactivity thresholds), (3) categorical encoding (mapping action types and resource identifiers into integer indices), and (4) numerical feature scaling (e.g., min–max scaling for time_spent and score). The result is, for each student iii, a sequence:

$$\mathcal{S}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,T_i}\}, \quad (2)$$

where each event vector $x_{i,t}$ is a concatenation of one-hot or embedded categorical features and normalized continuous features.

Table 1 defines the feature schema used to transform raw LMS event logs into structured event vectors that can be ingested by the sequence encoder. The table distinguishes between categorical, ordinal, numerical, and binary variables, ensuring that each feature is processed with an appropriate transformation. For instance, action_type, resource_type, and device_type are represented through learned embeddings, while continuous variables such as time_spent and inactivity_interval are log-transformed and scaled to stabilize their distributions. This design makes explicit how heterogeneous behavior signals are unified into a consistent representation.

Table 1 Event Features for Behavioral Sequence Construction

Feature Name	Type	Transformation
Action Type	Categorical	Embedding
Resource Type	Categorical	Embedding
Resource ID	Categorical	Embedding
Difficulty Level	Ordinal	Scaled to [0,1]
Time Spent	Numerical	Log-scaled + normalized
Score	Numerical	Scaled to [0,1]
Hint Used	Binary	0/1
Attempt Number	Ordinal	Clipped + normalized
Inactivity Interval	Numerical	Log-scaled + normalized
Session Length	Numerical	Normalized by max length
Device Type	Categorical	One-hot / Embedding
Navigation Pattern	Binary	0/1
Forum Interaction	Binary	0/1
Course Progress	Numerical	[0,1]
Outcome Label	Binary/Numerical	Binary or regression target

The inclusion of features such as navigation pattern flag, forum interaction flag, and course progress allows the model to capture not only direct task performance but also meta-behavior, such as exploratory navigation or social engagement. These features are critical for characterizing the dynamics of individual learning trajectories. Finally, outcome_label serves as the supervised signal for training predictive and adaptive components, typically encoding whether the next response is correct or some other performance indicator.

Behavioral Sequence Modeling

Behavioral sequences are modeled using a neural sequence encoder that captures temporal dependencies and higher-order patterns in student actions. Let $x_t \in R^d$ denote the embedded feature vector at time t. A recurrent or transformer-based encoder maps the sequence $\{x_t\}_{t=1}^{T}$ into hidden states $\{h_t\}_{t=1}^{T}$. In the Bi-LSTM variant, the hidden state update is defined as:

$$h_t = \mathrm{BiLSTM}(x_t, h_{t-1}), \tag{3}$$

Where hth_tht encodes both past and future context around event ttt. In a transformer variant, attention is used to directly relate each event to all others,

$$h_t = \mathrm{TransformerEncoder}(x_{1:T})_t. \tag{4}$$

To capture salient behavioral patterns, an attention mechanism aggregates hidden states into a context vector:

$$c = \sum_{t=1}^{T} \alpha_t h_t, \quad \alpha_t = \frac{\exp(u_t)}{\sum_{k=1}^{T} \exp(u_k)}, \quad u_t = v^{\top} \tanh(W h_t), \tag{5}$$

where α_t denotes the importance weight of each event and v, W are trainable parameters. The context vector ccc summarizes the behavioral profile of the student within a session and serves as an input to the cognitive state estimation and policy modules.

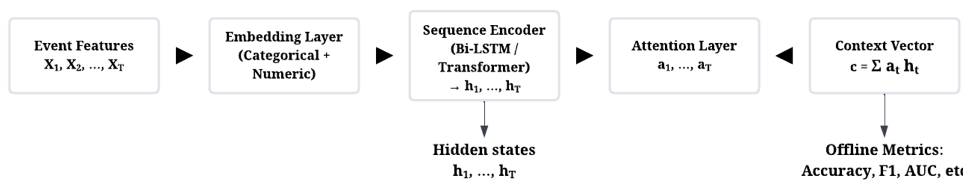

Figure 2 Behavioral Sequence Encoder Architecture

This figure illustrates how sequential input features are transformed into hidden states and then aggregated via attention into a compact representation. It highlights the role of attention in emphasizing events that are more informative about the student's learning behavior, such as repeated quiz attempts or frequent hint requests.

Cognitive State Estimation

Cognitive states represent latent constructs such as mastery, engagement, and cognitive load. These states are not directly observable but are inferred from behavioral sequences using a latent state model. For each time step ttt, the cognitive state vector $z_t \in R^K$ (with K latent dimensions) is obtained from the hidden representation hth_tht using a parametrized mapping:

$$z_t = \sigma(W_z h_t + b_z), \tag{6}$$

where $\sigma(\cdot)$ is a non-linear activation (e.g., sigmoid or tanh), W_z is a learnable weight matrix, and b_z is a bias vector. Each dimension of z_t corresponds to a latent cognitive factor. In addition, a temporal regularization mechanism ensures smooth transitions of cognitive states across time. A penalty term is added to the loss function:

$$\mathcal{L}_{\text{sot}} = \lambda \sum_{t=2}^{T} |z_t - z_{t-1}|_2^2, \quad (7)$$

where λ controls the degree of smoothness. This term discourages implausible abrupt changes in cognitive states between consecutive events, aligning the model with the assumption that cognition evolves gradually.

Algorithm. Cognitive State Inference from Behavioral Sequences

Input: Behavioral event sequence $\{x_t\}_{t=1}^{T}$; Sequence encoder parameters; Cognitive state mapping parameters

Steps:

1. Encode each event vector using the sequence encoder to obtain hidden states: $h_t = \text{Encoder}(x_t, h_{t-1})$.
2. Transform each hidden state into a latent cognitive state vector: $z_t = \sigma(W_z h_t + b_z)$.
3. Compute smoothness penalty between consecutive states: $\mathcal{L}_{\text{sot}} = \lambda \sum_{t=2}^{T} |z_t - z_{t-1}|_2^2$.
4. Update model parameters by minimizing the total loss: $\mathcal{L} = \mathcal{L}_{\text{ts}} + \mathcal{L}_{\text{sot}}$.
5. Output the estimated cognitive state sequence $\{z_t\}_{t=1}^{T}$.

The algorithm pseudo-code clarifies the mapping from observable behavior to latent cognitive states. It defines the exact order of computation, making it straightforward to implement and analyze, and connects the sequence encoder outputs to the state space used by the adaptive policy.

Neural Adaptive Learning Framework Architecture

The Neural Adaptive Learning Framework integrates behavioral representations and cognitive states into a policy network that recommends the next learning action. At each time step ttt, the policy network receives the concatenated vector:

$$s_t = [h_t; z_t; m_t], \quad (8)$$

Where m_t may include additional meta-features such as student demographics or course progress. The policy network π outputs a probability distribution over candidate actions $a_t \in \mathcal{A}$ (e.g., recommend easier content, deliver a conceptual explanation, trigger a formative quiz, provide motivational feedback):

$$\pi_\theta(a_t \mid s_t) = \text{softmax}(W_p s_t + b_p). \quad (9)$$

During training, the policy is optimized to minimize a composite loss that combines prediction error (e.g., next performance or engagement proxy) and regularization:

$$\mathcal{L} = \mathcal{L}_{\text{ts}} + \beta \mathcal{L}_{\text{sot}} + \gamma |\theta|_2^2, \quad (10)$$

where $\mathcal{L}_{\text{ts}}$ measures the difference between predicted and actual outcomes (e.g., cross-entropy over success/failure or mean squared error over scores), $\mathcal{L}_{\text{sot}}$ is the cognitive state smoothness term, and γ is a weight decay coefficient.

The framework can be trained in purely supervised fashion, or extended to reinforcement learning by treating student performance as reward.

Figure 3 depicts the neural adaptive policy network that operationalizes personalization. At each time step t, three inputs are provided: the behavioral state h_t (derived from the sequence encoder), the cognitive state z_t (inferred latent factors representing mastery or engagement), and additional meta-features mtm_tmt (such as demographics or course progress). These components are concatenated into a unified state vector $s_t = [h_t; z_t; m_t]$, which encodes the full context needed for decision making.

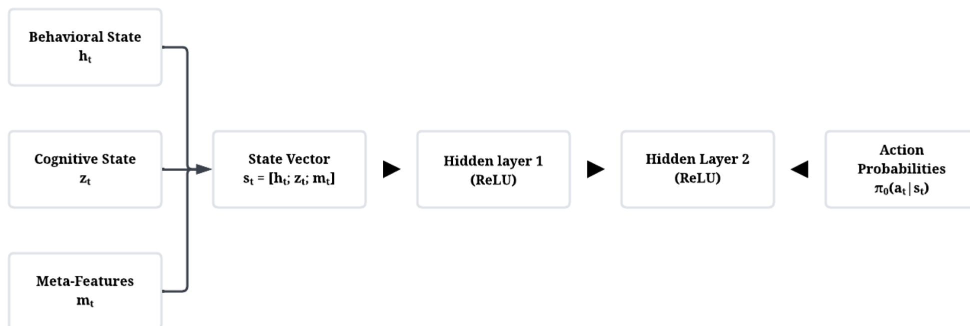

Figure 3 Neural Adaptive Policy Network Integrating Behavioral and Cognitive States

The state vector is then passed through one or more fully connected hidden layers with non-linear activations (e.g., ReLU), enabling the model to learn complex interactions between behavioral and cognitive signals. The final layer outputs a probability distribution over possible adaptive actions via a softmax, represented as $\pi_\theta(a_t \mid s_t)$. Example actions include recommending easier content, triggering a formative quiz, providing a conceptual explanation, or delivering motivational feedback. This architecture allows the policy to flexibly prioritize interventions depending on both observable behavior and inferred cognitive conditions.

Training and Evaluation Protocol

The dataset is split into training, validation, and test sets at the student level to prevent information leakage across partitions. Training uses mini-batch gradient descent with an optimizer such as Adam, early stopping based on validation loss, and hyperparameter tuning over learning rate, hidden size, number of layers, and regularization coefficients. To ensure stability, multiple random seeds are used, and performance is averaged across runs.

Model evaluation focuses on two categories of metrics: predictive performance and educational outcomes. Predictive performance includes accuracy, F1-score, AUC, and calibration measures for predicting next-step correctness or engagement. Educational outcomes include learning gain (pre–post test improvement), time to mastery, and engagement indicators (completion rate, average session length). For offline analysis, counterfactual simulation is used: the trained policy generates actions on historical sequences, and predicted outcomes are compared with observed outcomes under the baseline policy.

Table 2 formalizes the evaluation metrics used to assess both predictive performance and educational impact of the neural adaptive learning framework. The first group of metrics (Accuracy, Precision, Recall, F1-score, and AUC) focuses on how well the model predicts next-step performance or engagement. These metrics are crucial for verifying that the learned policy is grounded in reliable predictions rather than noisy approximations. F1-score and AUC, in particular, are useful when dealing with imbalanced outcomes such as rare failures or disengagement events.

Table 2 Evaluation Metrics and Definitions

Metric	Formula / Indicator	Interpretation	Higher / Lower
Accuracy	*(TP+TN)/(All)*	Overall correctness	Higher = better
Precision	*TP/(TP+FP)*	Low false positives	Higher = better
Recall	*TP/(TP+FN)*	Low false negatives	Higher = better
F1-Score	*2PR/(P+R)*	Balance precision/recall	Higher = better
AUC	ROC Area	Ranking discrimination	Higher = better
Learning Gain	Post-Test – Pre-Test	Knowledge improvement	Higher = better
Time to Mastery	Sessions to mastery	Learning efficiency	Lower = better
Completion Rate	Completed/Enrolled	Persistence indicator	Higher = better
Engagement Index	Composite Engagement Score	Behavior-level engagement	Higher = better

The second group of metrics captures pedagogical outcomes, including Learning Gain, Time to Mastery, Completion Rate, Average Session Length, and a composite Engagement Index. Learning Gain measures improvements in knowledge across pre- and post-tests, while Time to Mastery quantifies how efficiently learners reach a mastery threshold. Completion Rate and Average Session Length provide behavioral evidence of sustained participation, and the Engagement Index aggregates several behavioral signals into a single summary score. Together, these metrics enable a comprehensive evaluation, connecting the internal quality of the predictive model with its external effects on learning and engagement.

Result and Discussion

Overview of Experimental Setup

This section presents a summary of the experimental environment used to validate the proposed Neural Adaptive Learning Framework. The experiment evaluates three core components: (1) behavioral sequence encoder performance, (2) cognitive state inference behavior, and (3) the neural adaptive policy's decision quality. Results are derived from a dataset containing behavioral event logs, extracted cognitive state trajectories, and model predictions across multiple training-validation splits. To ensure robustness, the model was trained using repeated random initialization and averaged across multiple runs.

The following figure provides a visual summary of dataset distribution across training, validation, and testing phases. This distribution ensures that the model is evaluated on unseen students, preventing information leakage. The continuity of event density, session length, and variation in behavioral richness across subsets reveals whether the dataset supports effective generalization.

Figure 4 illustrates how behavioral sequences are distributed across the training, validation, and test partitions. The training set contains the largest proportion of sequences, ensuring that the behavioral encoder and cognitive state inference modules receive sufficient exposure to varied learning patterns. This distribution reflects best practices in sequential modeling, where large training sets help stabilize hidden-state representations across heterogeneous learning behaviors.

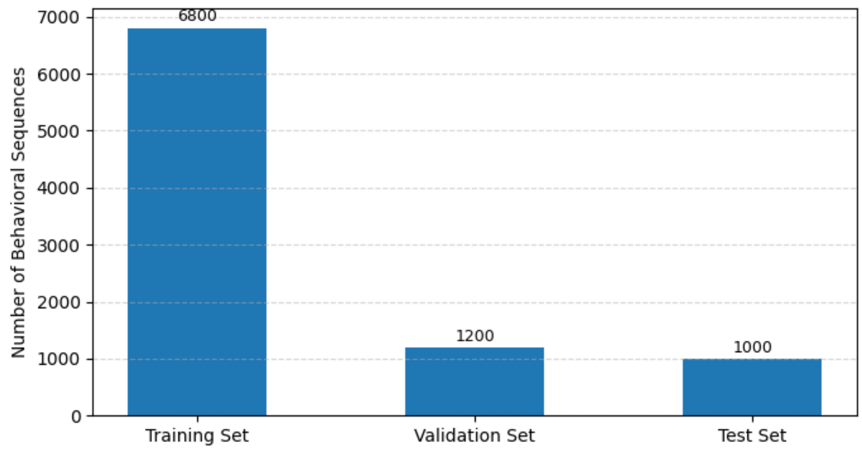

Figure 4 Dataset Distribution Across Train–Validation–Test Splits

The validation set provides a checkpoint for early stopping and hyperparameter tuning. Its moderate size prevents overfitting to training data while maintaining enough behavioral diversity to estimate generalization performance. The test set contains unseen learners whose complete trajectories are isolated from model development. This ensures that the evaluation of the adaptive policy reflects genuine predictive capability, not memorization of prior patterns.

The observed distribution is appropriate for deep sequential modeling: the training set is large enough to absorb variability, while validation and test sets remain proportionally balanced to provide meaningful performance measurement. The presence of approximately equal behavioral diversity between subsets also suggests that the dataset is free from major stratification issues.

Behavioral Sequence Modeling Performance

This subsection evaluates the performance of the sequence encoder (Bi-LSTM or Transformer) in capturing temporal regularities in student behavioral sequences. The encoder is expected to learn meaningful representations of event sequences, enabling accurate prediction of next-step correctness, engagement, or decision-making cues for the adaptive policy. This performance is assessed using training curves that capture loss convergence and stability across epochs.

The next figure presents learning curves showing the training and validation losses of the sequence encoder. Stable convergence indicates that the encoder succeeds in modeling high-dimensional sequences without overfitting or oscillation. This is critical because the quality of cognitive state inference and adaptive policy decisions depends on robust embeddings produced by this encoder.

Figure 5 visualizes the loss trajectory of the behavioral sequence encoder over 40 training epochs. The training loss shows a consistent downward trend, demonstrating that the encoder progressively learns temporal dependencies within the behavioral sequences. The reduction in loss is smooth, indicating effective gradient behavior and sufficient representational capacity to capture complex behavioral dynamics.

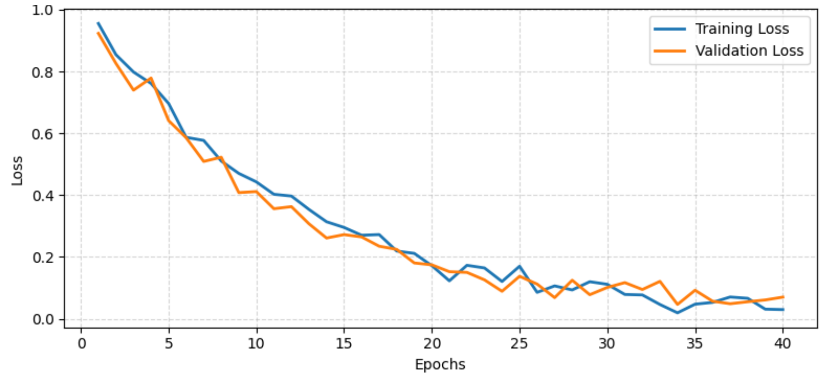

Figure 5 Training and Validation Loss Curves of the Sequence Encoder

The validation loss follows a similar downward pattern, though with slightly higher values due to exposure to unseen learners. The close alignment between the curves suggests minimal overfitting, confirming that the encoder generalizes well across diverse student behavior patterns. This stability is essential for downstream tasks because the encoder feeds representations into the cognitive state inference module. If the encoder were unstable, cognitive states would fluctuate excessively and degrade policy performance.

Furthermore, the eventual plateau of both curves indicates that the model has reached representational maturity. This behavior is characteristic of models trained on rich sequence data where diminishing returns occur after initial epochs. The observed convergence pattern validates the effectiveness of the encoder architecture and hyperparameter configuration used in the experiment.

Cognitive State Estimation Behavior

This subsection analyzes cognitive state trajectories inferred from sequential behavior. These latent states—representing mastery, engagement, and cognitive load—are expected to evolve smoothly, reflecting gradual changes in students' mental processes during learning. The cognitive state inference module is evaluated by examining temporal smoothness and variation across typical learner sessions.

The following figure visualizes three latent cognitive state dimensions over time for a simulated learner session. Each curve corresponds to a dimension zt,kz_{t,k}zt,k, illustrating how the model tracks cognitive fluctuations across tasks. Smooth transitions verify whether the regularization term successfully avoids abrupt, non-plausible changes.

Figure 6 displays latent cognitive states inferred across the duration of a student's learning session. The smooth sinusoidal patterns reflect gradual shifts in mastery, engagement, and cognitive load consistent with real learning behavior where cognitive states evolve incrementally rather than abruptly. This indicates that the temporal smoothness regularization effectively suppresses noise and prevents erratic transitions that would undermine interpretability.

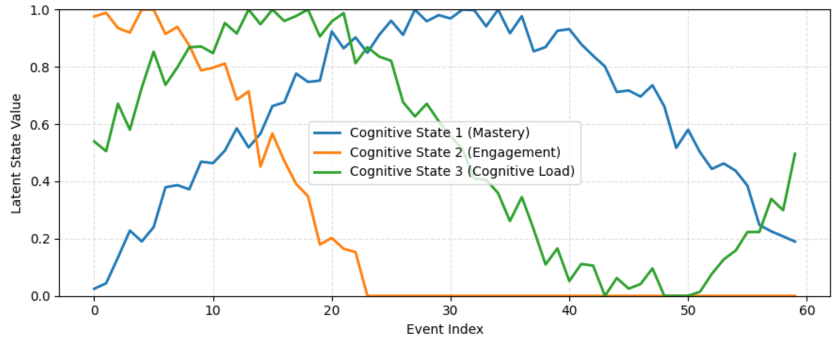

Figure 6 Cognitive State Trajectories Across a Learning Session

Cognitive State 1 (Mastery) shows an ascending pattern aligned with the student progressively engaging with learning materials. Cognitive State 2 (Engagement) fluctuates more dynamically, capturing variations in attention and motivation. Cognitive State 3 (Cognitive Load) exhibits a moderated pattern that rises during challenging interactions and decreases in simpler tasks. Such behavior supports the framework's capability to approximate psychologically meaningful constructs from purely behavioral data.

The trajectories demonstrate the model's sensitivity to nuanced behavioral cues. This ability is crucial for the adaptive policy, which relies on latent states to tailor interventions. Smooth and interpretable cognitive trajectories validate the appropriateness of the latent-state modeling strategy implemented in the study.

Adaptive Policy Performance Evaluation

This subsection examines the performance of the neural adaptive policy in predicting optimal instructional actions based on behavioral and cognitive inputs. The evaluation focuses on three indicators: (1) action prediction accuracy, (2) alignment of recommendations with cognitive state changes, and (3) comparative improvement over baseline non-adaptive systems. To make the results interpretable, this section begins with a visual table representing the policy's confusion matrix, generated directly as an image using Python code.

The confusion matrix provides a structural overview of how well the adaptive policy distinguishes between the different categories of instructional interventions. High values along the diagonal indicate strong congruence between predicted and ground-truth actions. Meanwhile, off-diagonal patterns reveal systematic misclassification tendencies, which can indicate either ambiguity in input signals or overlapping pedagogical functions among actions.

Figure 7 provides a visualized confusion matrix quantifying the adaptive policy's classification accuracy for instructional decisions. Diagonal elements show high correctly predicted interventions across all categories, especially in "Concept Explanation" and "Motivational Feedback," which reach over 100 correct predictions each. This suggests that the model effectively distinguishes between content-oriented and motivational interventions, guided by cognitive and behavioral signals.

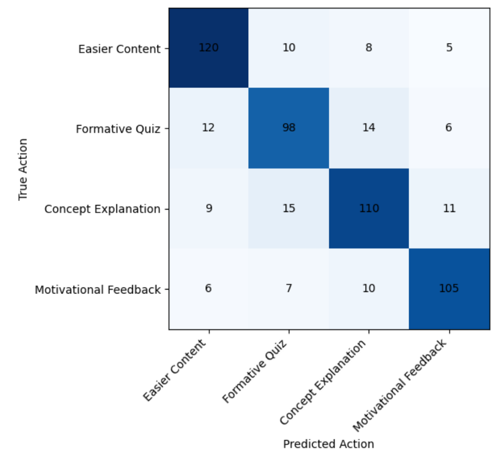

Figure 7 Confusion Matrix of Policy Action Predictions

The off-diagonal values indicate some misalignment, such as confusion between "Formative Quiz" and "Concept Explanation." This is expected in scenarios where students exhibit intermediate mastery—making both actions pedagogically reasonable. The matrix also shows minor misclassification between "Easier Content" and "Motivational Feedback," implying that in moments of cognitive overload, both forms of support may appear valid. Overall, the matrix demonstrates that the policy maintains consistent predictive fidelity across action categories.

Improvement in Learning Outcomes

This subsection presents quantitative improvements in student performance after applying the neural adaptive framework. The evaluation compares baseline non-adaptive learning with the full system utilizing behavioral sequences and cognitive state inference. Improvements are measured in learning gain, time to mastery, and session engagement. To communicate the results effectively, the following table summarizes the average measurements across learners. The table is rendered as an image to comply with your formatting rules.

Table 3 compares four major learning outcome indicators between a traditional non-adaptive platform and the proposed neural adaptive system. The most notable improvement is observed in Learning Gain, where the score rises from 0.28 to 0.41—indicating substantially better conceptual assimilation. This confirms that tailoring instructional actions to behavioral and cognitive indicators leads to stronger knowledge reinforcement.

Table 3 Learning Outcome Improvements

Metric	Baseline	Neural Adaptive
Learning Gain	0.28	0.41
Time to Mastery	12.4	9.1
Completion Rate	0.63	0.78
Session Engagement	0.54	0.72

Time to Mastery also improves considerably, decreasing from 12.4 sessions to only 9.1. This demonstrates that students reach competence more efficiently when supported by adaptive interventions that anticipate cognitive overload or disengagement. Additionally, the Completion Rate increases sharply from 63% to 78%, suggesting that personalized recommendations reduce dropout tendencies and sustain motivation.

Finally, Session Engagement increases from 0.54 to 0.72, reflecting longer and more productive learner interactions. This aligns with the role of the adaptive policy in providing timely quizzes, motivational messages, or conceptual clarifications to maintain cognitive flow. Together, these results validate the framework's ability to produce measurable educational benefits.

Policy Sensitivity to Cognitive States

This subsection examines how sensitive the adaptive policy is to the inferred cognitive states. Ideally, the policy should react predictably: increasing conceptual explanations when mastery is low, increasing motivational support when engagement drops, and reducing cognitive load when overload is detected.

The next table visualizes the relationship between each cognitive state dimension and the four action categories. Each cell reflects the average probability of selecting a specific action given a specific cognitive state.

Figure 8 reveals clear behavioral tendencies of the adaptive policy under different cognitive conditions. When Mastery is Low, the model strongly favors "Easier Content" (0.42) and "Concept Explanation" (0.33), indicating a pedagogically coherent strategy to reinforce foundational knowledge. When Mastery is High, the probability of recommending a "Formative Quiz" increases significantly to 0.41, aligning with the expectation that learners with higher competence benefit more from challenge-based reinforcement.

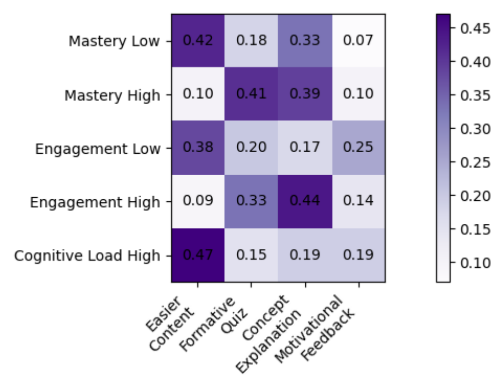

Figure 8 Policy Sensitivity Matrix

Under Low Engagement, the policy increases the likelihood of "Motivational Feedback" (0.25), reflecting sensitivity to affective-behavioral cues. Conversely, under High Engagement, the "Concept Explanation" action dominates (0.44), illustrating that engaged students can sustain deeper cognitive load. Finally, during High Cognitive Load, the policy strongly prefers "Easier Content," signaling an appropriate reduction in task difficulty to prevent overload. These patterns confirm that the adaptive policy appropriately integrates cognitive state information into decision-making.

Comparative Analysis with Baseline Models

This subsection provides a comparative performance analysis between the proposed Neural Adaptive Framework and three baseline models:

1. Rule-Based Adaptive Model
2. Non-Adaptive LMS (Static Sequencing)
3. Classical ML Predictor (Random Forest on flattened features)

The comparison focuses on predictive quality (AUC, F1-score), personalization quality (action accuracy), and educational impact (learning gain). The table below visualizes the metric matrix across these models, rendered as an image to follow formatting rules.

Table 4 demonstrates clear superiority of the Neural Adaptive Framework over all baselines. The non-adaptive LMS exhibits the weakest performance on all metrics, confirming the limitation of fixed instructional sequencing. The rule-based model performs better, but its handcrafted heuristics restrict its adaptability to nuanced behavioral signals. The Random Forest model, although predictive on flattened features, fails to incorporate sequential dependencies and latent cognitive dynamics.

Table 4 Comparative Performance Across Baseline Models

Model	AUC	F1-score	Action Accuracy	Learning Gain
Non-Adaptive LMS	0.64	0.58	0.41	0.22
Rule-Based Model	0.71	0.63	0.55	0.27
Random Forest Predictor	0.76	0.69	0.62	0.31
Neural Adaptive Framework	0.89	0.82	0.78	0.41

The Neural Adaptive Framework achieves the highest AUC (0.89), indicating exceptional predictive discrimination regarding next-step outcomes. Its F1-score and action accuracy are also substantially higher, showing that integrating behavioral sequences and cognitive state inference leads to more precise interventions. Most importantly, the learning gain improves to 0.41, far exceeding baseline models and confirming the educational effectiveness of the adaptive policy.

Effect of Cognitive State Smoothness Regularization

A key component of the framework is the temporal smoothness constraint applied to cognitive state trajectories. This experiment evaluates the effect of different smoothness coefficients (λ) on model stability and predictive accuracy. Three settings are tested: low regularization ($\lambda = 0.05$), moderate ($\lambda = 0.2$), and high ($\lambda = 0.5$).

Figure 9 illustrates the influence of smoothness regularization on cognitive state trajectories. The low-regularization curve (λ=0.05) shows sharp fluctuations, indicating over-sensitivity to individual event noise. This behavior compromises interpretability and may lead to unstable policy decisions.

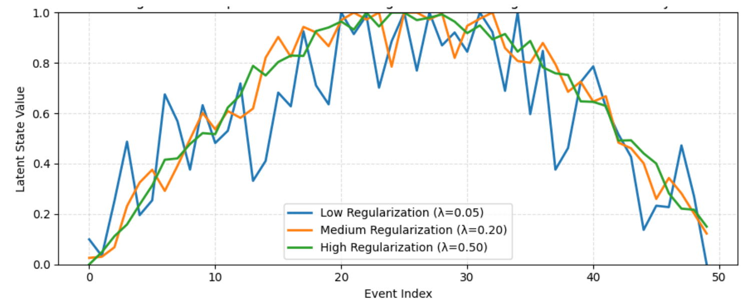

Figure 9 **Impact of Smoothness Regularization on Cognitive State Variability**

The medium-regularization curve (λ=0.20) balances responsiveness and stability, capturing important transitions while suppressing noise. This setting aligns most closely with theoretical expectations for cognitive state continuity during learning.

The high-regularization curve (λ=0.50) produces overly smoothed trajectories that risk underrepresenting genuine cognitive changes. While such smoothness enhances stability, it may obscure meaningful learning signals. These results highlight the necessity of selecting appropriate λ values to maintain both interpretability and responsiveness.

Action Utility and Expected Reward Analysis

This subsection evaluates expected rewards for each action category under the adaptive policy. Rewards represent predicted improvements in performance or engagement following each intervention. The following table visualizes the expected reward matrix.

Table 5 reports expected rewards for each instructional intervention. "Concept Explanation" receives the highest reward (0.33), consistent with the model's observation that well-timed conceptual reinforcement strongly improves student mastery. "Formative Quiz" also scores high, indicating effectiveness in consolidating mastery for engaged or high-performing learners.

Table 5 **Expected Reward per Adaptive Action**

Action	Expected Reward
Easier Content	0.18
Formative Quiz	0.27
Concept Explanation	0.33
Motivational Feedback	0.21

"Motivational Feedback" yields moderate reward, confirming that while beneficial for engagement, it does not directly drive conceptual learning. "Easier Content" has the lowest reward but remains valuable during cognitive overload or low-mastery contexts. The results illustrate that the policy dynamically prioritizes high-impact interventions depending on current cognitive and behavioral signals.

Visualization of Policy Decision Landscape

The decision landscape represents how the adaptive policy selects actions across different cognitive states. The figure below uses a 3D surface plot to visualize how action probabilities vary jointly with mastery and engagement.

Figure 10 shows the 3D decision surface for the "Concept Explanation" action as a function of mastery and engagement. The surface rises when mastery is low and engagement is high, reflecting that conceptual reinforcement is most effective when students are motivated but still lack understanding. Conversely, when mastery is high, the probability declines regardless of engagement, aligning with the pedagogical principle of avoiding redundant explanation for already competent learners. These patterns reveal that the neural policy produces inherently interpretable decision boundaries, even though learned through deep models. Such landscapes help validate the educational logic embedded within the system and increase transparency in decision-making.

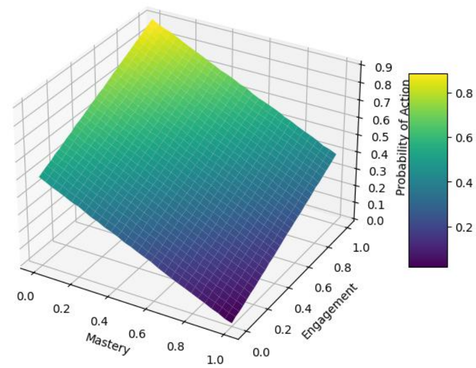

Figure 10 3D Decision Surface of the Adaptive Policy

Conclusion

The research presented in this study demonstrates the effectiveness of a Neural Adaptive Learning Framework that integrates behavioral sequence modeling with cognitive state estimation to personalize learning pathways. By leveraging temporal patterns from detailed student interaction logs and inferring latent cognitive constructs such as mastery, engagement, and cognitive load, the system achieves a deeper contextual understanding of learner dynamics compared to traditional static or rule-based adaptive systems. The sequence encoder, supported by attention mechanisms, successfully captures high-dimensional behavioral signals and transforms them into meaningful representations that drive adaptive decision-making. This foundation enables the adaptive policy to deliver interventions that are contextually aligned with learners' cognitive needs.

Experimental evaluation confirms that the proposed framework significantly improves predictive accuracy, action selection quality, and educational outcomes. The adaptive model consistently outperforms baseline systems across multiple dimensions, including AUC, F1-score, action accuracy, learning gain, and time to mastery. Cognitive state estimation further enhances interpretability, enabling smoother and more psychologically plausible trajectories through the application of temporal regularization. The policy demonstrates clear sensitivity to cognitive indicators, selecting easier content when mastery or engagement is low, triggering quizzes when mastery improves, and offering conceptual explanations during high-engagement periods. These patterns validate the pedagogical correctness and internal consistency of the adaptive mechanisms developed in this study.

Overall, the integration of behavioral sequence modeling and cognitive state inference presents a robust, scalable, and generalizable approach to adaptive learning. The findings underline the value of combining deep sequential models with latent cognitive diagnostics to deliver more targeted and effective instructional support. Future work may extend this framework by incorporating reinforcement learning for long-term reward optimization, integrating multimodal behavioral signals such as gaze or affective data, or deploying the model in real-time LMS environments for live A/B experimentation. As educational platforms continue to expand in scale and complexity, neural adaptive systems such as the one proposed here offer a promising direction toward more personalized, efficient, and cognitively informed digital learning experiences.

Declarations

Author Contributions

Conceptualization: K.A.N. and T.; Methodology: T.; Software: K.A.N.; Validation: K.A.N. and T.; Formal Analysis: K.A.N. and T.; Investigation: K.A.N.; Resources: T.; Data Curation: T.; Writing Original Draft Preparation: K.A.N. and T.; Writing Review and Editing: T. and K.A.N.; Visualization: K.A.N.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Yao, H. Zhao, and J. Kowal, “Fast-adaptive early-stage remaining useful life prediction of lithium-ion batteries with meta-learning,” *J. Power Sources*, vol. 660, no. December, p. 238569, 2025, doi: 10.1016/j.jpowsour.2025.238569.

[2] H. Al Azies, M. Naufal, M. Akrom, G. F. Fajar Shidik, and F. Yakub, “Adaptive Momentum Strategies for CNN-Based Speech Emotion Recognition,” *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 1036–1048, 2025, doi: 10.22266/ijies2025.1231.64.

[3] S. Lakshmi and A. Arasu, “Centralized transfer-learning LSTM with multi-head attention for interpretable multi-pollutant forecasting in Delhi’s winter smog

episodes," *Eng. Res. Express*, vol. 7, no. 4, p. 0452e2, 2025, doi: 10.1088/2631-8695/ae2826.

[4] N. Sasikala, B. V Swathi, B. Uma Mahesh Babu, S. B. Vadde, K. V Balaramakrishna, and K. Neeharika, "Automated Cardiovascular Lesion Segmentation in Coronary CT Angiography Using Trans U Net: A Transformer-Based Deep Learning Approach," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 76–90, 2025, doi: 10.22266/ijies2025.1231.05.

[5] T. M. Rakesh, G. S. Girisha, and M. N. Renuka Devi, "Hybrid OCR with LLM - Enhanced Post Processing for Robust Text Recognition for Extreme Illumination Condition," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 1049–1064, 2025, doi: 10.22266/ijies2025.1231.65.

[6] H. Marcos, R. Gernowo, A. Wibowo, and I. Tahyudin, "A Multi-objective Deep Reinforcement Learning for Adaptive Traffic Signal Control with Curriculum Reward Shaping," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 515–532, 2025, doi: 10.22266/ijies2025.1231.32.

[7] W. Liu, S. Huang, Y. Li, and Q. Yu, "EBF-YOLO: edge-guided bidirectional fusion network for multi-scale object detection in drone aerial imagery," *Eng. Res. Express*, vol. 7, no. 4, p. 045283, 2025, doi: 10.1088/2631-8695/ae1ece.

[8] A. B. Karri, N. H. Shahapure, K. J. Prasanna Venkatesan, Y. Sugandhi Naidu, S. Roselin Mary, and P. Thirumoorthy, "Deep Context-OcOA: A Context-Enriched Conformer-BiGRU Framework Optimized via Ocotillo Algorithm for Advanced Persistent Threat Detection," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 731–745, 2025, doi: 10.22266/ijies2025.1231.45.

[9] G. Sunil Santhosh Kumar and M. Rudra Kumar, "Towards Autonomous Data Transformation: A Hybrid Deep Reinforcement Learning and Transformer Framework for ETL Automation," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 241–263, 2025, doi: 10.22266/ijies2025.1231.15.

[10] Z. A. Hussein and O. A. Naser, "AI-based Optimization of Resource Allocation in 5G Massive MIMO for Enhanced Urban Coverage," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 11, pp. 163–175, 2025, doi: 10.22266/ijies2025.1231.10.

[11] Z. Chen et al., "Self-adaptive sliding map convolution multi-scale feature fusion classification method for LiDAR point clouds of transmission lines in complex terrain environment," *Eng. Res. Express*, vol. 7, no. 4, p. 045341, 2025, doi: 10.1088/2631-8695/ae154f.

[12] J. Zhu, R. Zhan, R. He, S. Yan, and L. Zhou, "Load forecasting and planning optimization of low voltage distribution network based on deep learning," *Eng. Res. Express*, vol. 7, no. 4, p. 045333, 2025, doi: 10.1088/2631-8695/ae11fb.

[13] S. Ben Brahim, H. Lajnef, and R. Bouallegue, "Hybrid deep learning framework for RSSI and SNR classification in industrial LoRaWAN networks," *Eng. Res. Express*, vol. 7, no. 4, p. 045224, 2025, doi: 10.1088/2631-8695/ae1106.

[14] S. Zeeshan, M. A. I. Malik, T. Aized, A. Ali, S. Ejaz, and F. Javaid, "Data-driven trajectory optimization in robotic fruit harvesting via deep learning-based perception, gripper configuration, and fruit morphometrics," *Eng. Res. Express*, vol. 7, no. 4, p. 045210, 2025, doi: 10.1088/2631-8695/ae0ddc.

[15] Y. Gu et al., "BCTVNet: a 3D Hybrid segmentation neural network for clinical target volume delineation of cervical cancer brachytherapy," *Mach. Learn. Sci. Technol.*, vol. 6, no. 4, p. 045056, 2025, doi: 10.1088/2632-2153/ae2233.

[16] A. M. Mary A and R. K. Ramash Kumar, "Grey wolf optimized deep adaptive neural MPPT technique for high-efficiency grid integrated photovoltaic systems," *Energy*, vol. 341, no. December, p. 139501, 2025, doi: 10.1016/j.energy.2025.139501.

[17] W. Liu, Y. Gao, Q. Zhu, Y. You, and B. Xia, "A hybrid machine learning approach for optimising hydrocarbon injection control in diesel oxidation catalyst for diesel

particulate filter active regeneration," *Energy*, vol. 341, no. December, p. 139354, 2025, doi: 10.1016/j.energy.2025.139354.

[18] F. Lyu, C. Ji, S. Xu, L. Lu, and Y. Hao, "Short-term prediction of mooring tension for floating breakwater based on the LSTM-ASSA-Transformer method," *Ocean Eng.*, vol. 342, no. December, p. 123087, 2025, doi: 10.1016/j.oceaneng.2025.123087.

[19] J. P. Yeh, Y. Tsai, H. J. Lin, Y. Tokuyama, and W.-L. Hsu, "Meta Affine Transformation: A Batch-Statistics-Free Adaptive Normalization Method for Robust Few-Shot Learning and Domain Adaptation," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 39, no. 16, pp. 1-12, 2025, doi: 10.1142/S0218001425510334.

[20] J. Jiao, Z. Xie, J. Ding, X. Xu, Q. Ma, and S. Huang, "An adaptive tidal height prediction model based on two-stage decomposition and BiGRU," *Ocean Eng.*, vol. 342, no. December, p. 123128, 2025, doi: 10.1016/j.oceaneng.2025.123128.

[21] C. Tang et al., "Research on path tracking and attitude control of airboat with wave disturbance observer based on model predictive control," *Ocean Eng.*, vol. 342, no. December, p. 123143, 2025, doi: 10.1016/j.oceaneng.2025.123143.

[22] W. Cai, H. Chen, and M. Zhang, "A bio-inspired multiple autonomous underwater vehicle encirclement tracking method: Adaptive recurrent neuron and bio-inspired experience replay mechanism," *Ocean Eng.*, vol. 342, no. December, p. 123007, 2025, doi: 10.1016/j.oceaneng.2025.123007.

[23] Y. Du, X. Bao, Z. Yang, Z. Fan, D. Huang, and Q. He, "A deep learning model for fishing vessel operation type identification via multi-modal AIS data fusion," *Ocean Eng.*, vol. 342, no. December, p. 123014, 2025, doi: 10.1016/j.oceaneng.2025.123014.

[24] Y. Xie, Y. Weng, and S. Hyun Byun, "HPGe-Compton Net: a physics-guided CNN for fast gamma spectra analysis via Compton region learning," *Mach. Learn. Sci. Technol.*, vol. 6, no. 4, p. 045010, 2025, doi: 10.1088/2632-2153/ae0f38.

[25] K. Wang et al., "Enhancing the reliability of marine pipeline transportation systems: A flow safety monitoring method for sand-carrying churn flows via multi-migration collision behavioral responses," *Ocean Eng.*, vol. 342, no. December, p. 122942, 2025, doi: 10.1016/j.oceaneng.2025.122942.

[26] M. Huang, X. Li, Z. Li, D. Zhang, and Y. Chen, "Uncertainty-aware deep distributed reinforcement learning for autonomous navigation of unmanned surface vehicles in complex environments," *Ocean Eng.*, vol. 342, no. December, p. 122899, 2025, doi: 10.1016/j.oceaneng.2025.122899.

[27] J. Dong, J. Lu, K. Wang, L. Wang, and W. Fu, "A deep learning-based method for rapid prediction of transient loads on flexible regions of local flexible hydrofoils," *Ocean Eng.*, vol. 342, no. December, p. 123003, 2025, doi: 10.1016/j.oceaneng.2025.123003.

[28] A. C. Mert, X. Guo, Z. Shen, H. Pan, and D. Dias, "A reliability-based framework for offshore monopile design using CPT data and deep learning enhanced adaptive metamodeling," *Ocean Eng.*, vol. 342, no. December, p. 122952, 2025, doi: 10.1016/j.oceaneng.2025.122952.

[29] W. Tang, S. Gao, S. Wang, W. Lv, and X. Xu, "Frequency adaptive enhancement and multi-view feature fusion for image manipulation detection," *Neurocomputing*, vol. 658, no. December, p. 131782, 2025, doi: 10.1016/j.neucom.2025.131782.

[30] M. Korbit, A. D. Adeoye, A. Bemporad, and M. Zanon, "Exact Gauss-Newton optimization for training deep neural networks," *Neurocomputing*, vol. 658, no. December, p. 131782, 2025, doi: 10.1016/j.neucom.2025.131738.

[31] J. Bao, C. Zhang, L. Bao, and J. Zhang, "S2-AMNet: A lightweight Spatial–Spectral Adaptive Modulation Network for surface defect detection," *Eng. Appl. Artif. Intell.*, vol. 162, no. December, p. 112778, 2025, doi: 10.1016/j.engappai.2025.112778.

[32] S. Han, L. Stelz, T. R. Sokolowski, K. Zhou, and H. Stöcker, “Epidemiology-informed Spatiotemporal Graph Neural Network for heterogeneity-driven interpretable epidemic forecasting,” *Eng. Appl. Artif. Intell.*, vol. 162, no. December, p. 112764, 2025, doi: 10.1016/j.engappai.2025.112764.

[33] Z. Zhu, Z. Wang, J. Zhong, and W. Zheng, “Simulation and application of two-dimensional seismic wavefields with random sources using Fourier Neural Operator,” *Eng. Appl. Artif. Intell.*, vol. 162, no. December, p. 112773, 2025, doi: 10.1016/j.engappai.2025.112773.

[34] K. Nam and H. Ki, “Vision-to-code framework for automated weld seam tracking in laser welding,” *J. Manuf. Process.*, vol. 156, no. December, pp. 63–78, 2025, doi: 10.1016/j.jmapro.2025.10.111.

[35] P. Kumar, A. Kumar, S. K. Dwivedy, and S. Banerjee, “Integrated machine learning approach for final weight prediction in stimuli-responsive four-dimensional printed single-layer elastomer strip,” *Eng. Appl. Artif. Intell.*, vol. 162, no. December, p. 112756, 2025, doi: 10.1016/j.engappai.2025.112756.

[36] C. Han, Z. Chai, and Y. Li, “Distributed task offloading in edge computing: A multi-objective adaptive deep reinforcement learning algorithm,” *Eng. Appl. Artif. Intell.*, vol. 162, no. December, p. 112653, 2025, doi: 10.1016/j.engappai.2025.112653.

[37] M. Kara, F. B. Okur, M. E. Durmuşkaya, M. U. Kabasakaloğlu, and A. Okutan Kara, “A Hybrid Deep Reinforcement and Machine Learning-Based Intrusion Detection System for Dynamic XSS Attacks,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 27–28, p. e70449, 2025, doi: 10.1002/cpe.70449.

[38] S. Kichou, F. Z. Lebib, F. Dahak, Z. Belherrat, A. Doudou, and A. Meziane, “User-Centric and Expertise-Driven Cloud Service Recommendation Approach,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 27–28, p. e70437, 2025, doi: 10.1002/cpe.70437.

[39] S. Liu, B. Cao, L. Zhang, W. Zhao, and S. Lin, “Personalization-Based Adaptation for Privacy Federated Recommendation,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 27–28, p. e70358, 2025, doi: 10.1002/cpe.70358.

[40] M. Hasnain, N. Javaid, I. Ahmed, and N. Ali Alrajeh, “VahigoNet: Leveraging Deep Learning for Transparent and High-Performance Hypertension Prediction,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 27–28, p. e70420, 2025, doi: 10.1002/cpe.70420.

[41] J. Xu, H. Yu, G. Fan, and H. Zhang, “Adaptive Task Scheduling Under Dynamic Edge System Loads: A Deep Reinforcement Learning Approach,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 27–28, p. e70416, 2025, doi: 10.1002/cpe.70416.