INSYST
Journal of Intelligent System and Computation

Intelligent Pilot Training Recommender System and Electronic Flight Bag Handover Application

Jeremy Mart V Nainggolan[1], Meilyna Silvia A Hutajulu[1], and Samuel Situmeang[1]
[1] Information Systems Study Program, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, Toba, Indonesia

Corresponding author: Samuel Situmeang (e-mail: samuel.situmeang@del.ac.id).

ABSTRACT The aviation industry requires efficient and reliable pilot management tools to address the challenges posed by traditional methods for managing training documentation and Electronic Flight Bag (EFB) logistics, which often result in data inconsistencies and operational delays. This study presents the development and optimization of a comprehensive mobile application designed to enhance pilot training management and Electronic Flight Bag (EFB) handovers. While digitizing these processes using Flutter and GetX state management streamlines operational logistics, manually assigning mandatory remedial training remains a complex, error-prone task. To address this, the application was empowered with a machine learning-based recommendation engine. A synthetic aviation activity dataset was developed to benchmark traditional algorithms against state-of-the-art sequential transformer models, including eSASRec and HSTU. The frontend implementation revealed limitations in GetX's state synchronization, necessitating manual view updates to maintain data consistency. Conversely, the intelligent backend integration yielded highly robust predictive results. Evaluation metrics showed that sequential transformers (HSTU and eSASRec) achieved peak performance, with NDCG@3 of 0.521 and a perfect Partial AUC + Precision (PAP@3) of 1.0. These models also achieved a Serendipity@3 score of 0.130, indicating their ability to uncover latent remedial training requirements from granular assessment data reliably. Ultimately, embedding a sequential recommender system elevates the application from a conventional digital ledger into a proactive, intelligent platform for aviation safety compliance.

KEYWORDS Aviation Safety Compliance, Electronic Flight Bag, Pilot Training Management, Sequential Recommender Systems

I. INTRODUCTION

Rigorous pilot training and the efficient management of flight resources, such as the Electronic Flight Bag (EFB), are strictly required for aviation safety [1], [2], [3], [4]. Traditionally, extensive manual documentation and administrative oversight are heavily utilized to manage these operational processes. However, significant inefficiencies and a high susceptibility to human error are often encountered [5]. During the manual assignment of mandatory remedial training and the tracking of EFB handovers. Granular performance drops in pilot assessments are frequently overlooked when vast amounts of training data are manually reviewed by administrators. Furthermore, while mobile digitization is implemented as a partial solution to eliminate paper-based workflows, the cognitive capacity to proactively analyze historical assessment sequences and prescribe necessary compliance interventions is not provided by conventional static applications [6], [7]. The efficacy of analyzing historical performance logs to formulate predictive interventions, alongside the robust capability of transformer architectures to model complex sequential data, has been strongly established in recent computational research across various domains [8], [9].

To address these critical operational gaps, a comprehensive intelligent mobile application was developed. The digital tracking of EFB handovers and the logging of training card assessments were facilitated utilizing the Flutter framework and GetX state management [10], [11], [12]. Crucially, to mitigate the safety risks associated with manual training scheduling, a machine learning-based sequential recommender system was integrated into the backend architecture. In this study, the optimization of aviation training management is primarily aimed for through the deployment of this intelligent system, which is designed to dynamically prescribe remedial modules based on latent chronological patterns in pilot assessment histories. Additionally, to ensure optimal system accuracy, the predictive performance of traditional collaborative filtering algorithms and state-of-the-art sequential transformer models is rigorously benchmarked using a synthetic aviation activity dataset.

II. METHOD

The research was conducted utilizing a multi-method approach encompassing literature study [13], [14], direct observation [15], [16], software development prototyping [17], [18], and intelligent system simulation [19], [20]. This comprehensive framework ensures that both the operational digitization and the cognitive automation of the aviation training processes are rigorously evaluated.

A. DATA COLLECTION AND REQUIREMENT ANALYSIS

Initial requirements were gathered through direct observation of existing aviation business processes, specifically focusing on the manual administrative workflows associated with Electronic Flight Bag (EFB) handovers and pilot training card assessments. This was supplemented by a literature study reviewing current cross-platform mobile development frameworks and the application of machine learning in safety-critical compliance systems. These techniques established baseline operational constraints and identified the high susceptibility to human error during manual remedial training assignments.

B. APPLICATION DEVELOPMENT AND SYSTEM TESTING

The digitization of the EFB and training card processes was executed utilizing an agile prototyping methodology. To establish a robust foundation, the high-level system structure was mapped out prior to implementation. The holistic structure of the platform, including the separation of the client interface, data storage, and the newly integrated cognitive processing layer, is illustrated in Figure 1. Within this architecture, the Recommender Engine acts as an independent backend microservice, retrieving historical assessment data from the NoSQL database to compute required compliance training before serving the predictions back to the client application.

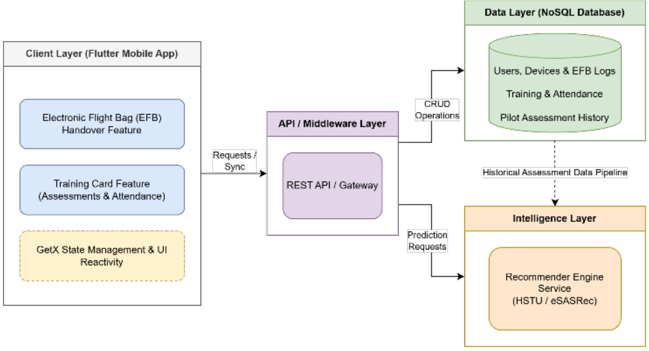

Figure 1. **Application Architecture Diagram**

The mobile application interface was developed utilizing the cross-platform Flutter framework. To manage the complex flow of screens—particularly during multi-step EFB handovers and detailed pilot assessment reviews, a streamlined routing protocol was required. The implementation of the navigation structure, which defines the hierarchical transitions between the training ledger and EFB modules, is depicted in Figure 2.

```
GetPage(
        name: AppRoutes.EFB_ANALYTICS,
        page: () => const EFB_Analytics(),
        binding: EFB_Analytics_Binding(),
        transition: Transition.native,
      ),
```

Figure 2. **Implementation of Navigation Structure**

To handle dynamic data inputs, such as real-time instructor grading and attendance logging, the GetX state management library was utilized. Reactive programming principles were applied to ensure the user interface automatically reflected database changes without requiring manual screen refreshes. The specific utilization of reactivity `Rx<T>` variables from the GetX library to bind the frontend UI to the underlying data models is detailed in Figure 3.

```
class EFB_Analytics extends GetView<EFB_Analytics_Controller> {
  const EFB_Analytics({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: ColorConstants.backgroundColor,
      body: Obx(() {
        // Check if the controller is loading data
        if (controller.isLoading.value) {
          // Show a loading animation while data is being fetched
          return Center(
            child: LoadingAnimationWidget.hexagonDots(
              color: ColorConstants.primaryColor,
              size: 48,
            ),
          );
        }

        // If data is loaded, display the analytics report
        return RefreshIndicator(
          onRefresh: controller.refreshData,
          child: SingleChildScrollView(
```

Figure 3. **Utilization of Reactivity Rx<T> from GetX**

Furthermore, performance optimization was prioritized during the development of the user interface. Because rendering complex assessment variable tables can cause frame drops on mobile devices, strategies to minimize unnecessary UI rebuilds were implemented. The architectural adjustments made to isolate state changes and achieve widget frequency reduction are presented in Figure 4.

```
class EFB_Analytics_Controller extends GetxController {
  final fromDateText = ''.obs; // Initial date for filtering
  final toDateText = ''.obs; // End date for filtering

  // Controllers for date input fields
  final fromDate = TextEditingController(
    text: DateFormatter.convertDateTimeDisplay(
      DateFormatter.getStartOfYear().toString(),
      'dd MMMM yyyy',
    ),
  );
  final toDate = TextEditingController(
    text: DateFormatter.convertDateTimeDisplay(
      DateFormatter.getCurrentDateWithoutTime().toString(),
      'dd MMMM yyyy',
    ),
  );
```

Figure 4. **Widget Frequency Reduction**

Finally, to evaluate the software's functional reliability, systematic system testing was performed. Analysis at this stage focused on identifying state synchronization limitations, specifically evaluating how data consistency across views occasionally necessitated manual overrides within the GetX architecture, despite the implementations outlined in Figures 3 and 4.

C. INTELLIGENT SYSTEM SIMULATION

Because real-world pilot assessment data is strictly confidential and heavily regulated, the machine learning methodology relied on highly controlled system simulation. A synthetic aviation activity dataset was generated programmatically to mirror the exact NoSQL JSON document structures of the production environment. To train the recommendation engine effectively, specific statistical correlations were injected into the simulation:

- Degradations in technical assessment variables (e.g., "Aircraft System Procedures") were programmed to trigger a higher probability of Recurrent Ground Training (RGT) assignments.
- "Unsatisfactory" soft-skill evaluations (e.g., "Threat/Error Management") were configured to force a Crew Resource Management (CRM) module assignment within a 30-day window.

D. RECOMMENDER SYSTEM BENCHMARKING AND ANALYSIS TECHNIQUES

To determine the optimal intelligent architecture, a comparative benchmark was conducted using the `RecTools` framework. The predictive capabilities of baseline algorithms and collaborative filtering models [21], [22], [23], [24] (Random, Popular, Implicit ALS, LightFM) were tested against state-of-the-art sequential transformer architectures, specifically the Hierarchical Sequential Transformer Unit (HSTU) [25] and eSASRec [26].

The analysis techniques utilized a TimeRangeSplitter to simulate real-world, chronological pilot training assignments across 30-day validation folds. The predictive accuracy and operational safety of the models were evaluated using the following specific metrics:

- **NDCG@K:** Analyzed to measure the fundamental accuracy and ranking quality of the prescribed mandatory trainings.
- **PAP (Partial AUC + Precision):** Utilized to evaluate the models at the absolute highest-ranked recommendations, simultaneously penalizing dangerous false negatives (missing critical recurrent training) and minimizing costly false positives (unnecessary simulator sessions).
- **Serendipity@K:** Calculated to measure the system's ability to uncover non-obvious remedial training requirements hidden within granular pilot assessment histories that would typically be overlooked during manual observation**.**

III. RESULT AND DISCUSSION

A. APPLICATION INTERFACE AND STATE MANAGEMENT IMPLEMENTATION RESULTS

The digitization of the Electronic Flight Bag (EFB) handover and pilot training assessment workflows was successfully deployed via the Flutter mobile application. The user interface effectively captured real-time flight logs, device conditions, and granular assessment variables. To rigorously evaluate the technical efficiency of this implementation, hardware performance profiling was conducted utilizing a mid-range mobile test device equipped with a Snapdragon 732G (8 nm) processor and 8GB of RAM. The evaluation focused on CPU usage, memory consumption, and rendering latency, comparing the application's baseline performance against the optimized GetX state management architecture (see Table 1) [27], [12].

TABLE I
MOBILE APPLICATION PERFORMANCE SUMMARY

Test Condition	Idle CPU (%)	Average Active CPU (%)	Peak CPU (%)
Without GetX Optimization	9.17	50.99	109.83
With GetX Optimization	2.83	49.20	95.00

The performance metrics reveal a critical technical trade-off inherent to the GetX architecture. The optimization successfully minimized core hardware resource consumption, reducing Average Active CPU from 50.99% to 49.20% and significantly decreasing total memory allocation from 435.8 MB to 270.0 MB. However, the rendering performance metrics exposed specific operational constraints. The introduction of `Rx<T>` reactive variables increased the percentage of Janky Frames from 0.44% to 1.68% and elevated the 90th percentile UI Thread rendering time from 8.50 ms to 11.17 ms.

This rendering latency quantitatively corroborates the state synchronization limitations observed during development. While GetX efficiently manages primitive state changes on isolated screens, its reactive overhead in deeply nested widget trees—such as updating a pilot's overall grading status from complex assessment sub-menus—causes frame drops and rendering delays. Consequently, data consistency across views could not rely entirely on automatic reactive bindings, necessitating the implementation of manual update() lifecycle methods to force widget rebuilds and ensure accurate data representation without further degrading the UI thread.

B. INTELLIGENT SYSTEM BENCHMARKING RESULTS

The integration of the recommendation engine was evaluated using the simulated aviation activity dataset across a 30-day TimeRangeSplitter fold. The comparative performance of baseline algorithms, collaborative filtering models, and sequential transformers is quantified and visually summarized in **Figure 5**.

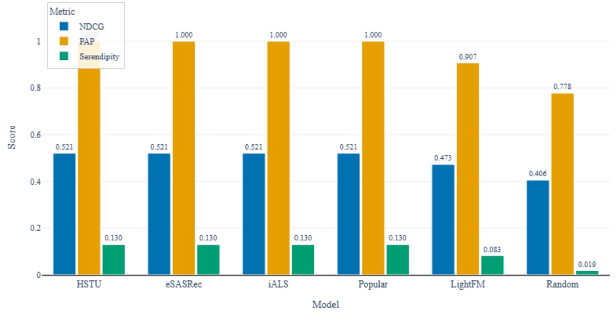

Figure 5. **Performance comparison of baseline, collaborative filtering, and sequential transformer recommender models evaluated across a 30-day TimeRangeSplitter fold**

The benchmark results demonstrate that the sequential transformer models—specifically the Hierarchical Sequential Transformer Unit (HSTU) and eSASRec—achieved peak predictive performance. As illustrated, both models recorded an NDCG@3 of 0.521 and a perfect Partial AUC + Precision (PAP@3) score of 1.000. In comparison, traditional collaborative filtering architectures such as LightFM yielded a lower PAP@3 score of 0.907.

Furthermore, the sequential transformers and the iALS model achieved a Serendipity@3 score of 0.130. This significantly outperformed both the Random baseline (0.018) and LightFM (0.083), quantitatively validating the advanced models' superior capability in identifying latent, non-obvious training requirements from complex pilot assessment histories.

C. DISCUSSION

The development and integration of this system yield significant implications for both mobile software engineering and aviation informatics. From a software architecture perspective, the observed limitations of GetX indicate that while it accelerates rapid prototyping, its reactive overhead in deeply nested widget trees makes it sub-optimal for highly complex, multi-layered aviation compliance forms. Future developments in this domain may require migrating to more robust, immutable state management solutions like standard BLoC (Business Logic Component) to guarantee determinism in safety-critical user interfaces.

More profoundly, the recommender system benchmarking confirms that embedding a cognitive processing layer completely transforms the operational utility of the application. The predictive supremacy of sequential transformers (HSTU and eSASRec) over static collaborative filtering is highly significant in the context of aviation. Static models evaluate a pilot's history as an unordered collection of past training. Conversely, sequential transformers process the chronological escalation of events, successfully mapping a consecutive degradation in specific assessment variables (e.g., "Decision Making") directly to the necessity of remedial interventions (e.g., Crew Resource Management).

The achievement of a 1.000 PAP@3 score by the sequential models is the most critical operational finding. In aviation training, false positives (recommending unnecessary full-flight simulator sessions) cause severe financial and scheduling inefficiencies, while false negatives (failing to recommend mandatory Recurrent Ground Training) lead to grounded pilots and regulatory violations. A perfect PAP metric at the top of the recommendation list guarantees that the highest-priority, safety-critical modules are ranked accurately without exception.

Furthermore, the high Serendipity score validates the system's capability as an intelligent administrative assistant. By successfully uncovering latent, non-obvious training requirements hidden within granular soft-skill assessments, often overlooked during manual human review—the application effectively transitions from a reactive digital ledger into a proactive, intelligent platform for optimizing pilot safety compliance.

IV. CONCLUSION

The development and optimization of the aviation training and Electronic Flight Bag handover application successfully addressed critical operational inefficiencies through a combination of architectural digitization and cognitive automation. From a software engineering perspective, the implementation of GetX state management demonstrated substantial proficiency in hardware resource optimization. Performance profiling indicated considerable reductions in average active CPU load and total memory allocation, signifying a highly efficient baseline state update mechanism. However, this resource efficiency was counterbalanced by slight rendering latency and an increase in janky frames when navigating deeply nested assessment interfaces. These synchronization constraints occasionally necessitated manual lifecycle overrides, indicating that while GetX is highly optimal for memory-efficient development, its reactive overhead presents distinct architectural limitations when handling complex, multi-layered aviation compliance forms.

Beyond interface digitization, the most significant operational advancement presented in this study is the integration of the intelligent recommendation engine. By embedding a machine learning backend, the application transcended its role as a static digital ledger to become a proactive safety management platform. Simulation benchmarking validated that sequential transformer architectures, specifically HSTU and eSASRec, are exceptionally suited for the aviation domain due to their capacity to process pilot assessment histories chronologically. Their demonstrated ability to achieve perfect predictive precision ensures that mandatory remedial modules—such as Crew Resource Management or Recurrent Ground Training—are safely and automatically prescribed based on latent patterns of technical or soft-skill degradation that manual administrative reviews might overlook.

Moving forward, it is recommended that subsequent iterations of safety-critical mobile interfaces explore strictly deterministic state management solutions, such as the Business Logic Component (BLoC) pattern, to entirely mitigate UI reactivity latency. Furthermore, future research should focus on expanding the intelligent system's feature matrix. Integrating real-world flight telemetry data alongside subjective instructor assessments would allow the sequential recommender models to continuously refine and personalize pilot training schedules based on live operational performance, further advancing the paradigm of intelligent aviation informatics.

AUTHORS CONTRIBUTION

Jeremy Mart V Nainggolan: Conceptualization (equal); Methodology (equal); Formal analysis (equal); Investigation (equal); Project administration (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

Meilyna Silvia A Hutajulu: Conceptualization (equal); Methodology (equal); Formal analysis (equal); Investigation (equal); Project administration (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

Samuel Situmeang: Supervision (lead); Project administration (supporting); Validation (supporting); Writing – review & editing (equal).

COPYRIGHT

REFERENCES

[1] M. Carroll and N. Dahlstrom, “Human Computer Interaction on the Modern Flight Deck,” *International Journal of Human–Computer Interaction*, vol. 37, no. 7, pp. 585–587, Apr. 2021, doi: 10.1080/10447318.2021.1890495.

[2] N. Dolzhenko, I. Assilbekova, A. Abzhapbarova, G. Mussayeva, and T. Sarzhanov, “Unification of Training Programs for Aviation Professionals as a Flight Safety Criterion,” *Journal of Aerospace Technology and Management*, vol. 16, 2024, doi: 10.1590/jatm.v16.1320.

[3] N. M. Lopes, M. Aparicio, and F. T. Neves, “Supporting Situational Awareness on Aviation Pilots: Key Insights Affecting the Use of Electronic Flight Bags Devices,” 2022, pp. 93–101. doi: 10.1007/978-3-031-04819-7_10.

[4] D. Yu and A. R. Dattel, “The Impact of Electronic Flight Bags on General Aviation Pilots’ Situation Awareness,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 69, no. 1, pp. 52–57, Sep. 2025, doi: 10.1177/10711813251370740.

[5] S. Suppiah, D. Liu, S.-A. Lee, A. Dattel, and D. Vincenzi, “Impact of Electronic Flight bag (EFB) on Single Pilot Performance and Workload,” *International Journal of Aviation, Aeronautics, and Aerospace*, 2020, doi: 10.15394/ijaaa.2020.1531.

[6] M. D. Briscoe, “The Paperless Office Twenty Years Later: Still A Myth?,” *Sustainability: Science, Practice and Policy*, vol. 18, no. 1, pp. 837–845, Dec. 2022, doi: 10.1080/15487733.2022.2146370.

[7] R. Othen, J. Padberg, C. Möbitz, and T. Gries, “Digital Transformation in the Paper Industry: Assessing Maturity, Challenges, and Opportunities,” *Sustainability*, vol. 17, no. 2, p. 770, Jan. 2025, doi: 10.3390/su17020770.

[8] J. C. Bonar Sirait, P. R. Togatorop, P. M. L. P. Tambunan, Y. F. Yanti Marpaung, S. I. G. Situmeang, and H. T. A. Simanjuntak, “Predicting Students Performance Using Data Mining Approach (Case Study: IT Del),” in *Proceedings of 2023 IEEE International Conference on Data and Software Engineering, ICoDSE 2023*, 2023. doi: 10.1109/ICoDSE59534.2023.10291748.

[9] S. Situmeang, S. R. Tambunan, Jevania, M. F. Simanjuntak, and S. Sinaga, “Transformer and Text Augmentation for Tourism Aspect-Based Sentiment Analysis,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 14, no. 6, pp. 4614–4622, 2025, doi: http://doi.org/10.11591/ijai.v14.i6.pp4614-4622.

[10] O. M. A. AL-atraqchi, “A Proposed Model for Build a Secure Restful API to Connect between Server Side and Mobile Application Using Laravel Framework with Flutter Toolkits,” *Cihan University-Erbil Scientific Journal*, vol. 6, no. 2, pp. 28–35, Aug. 2022, doi: 10.24086/cuesj.v6n2y2022.pp28-35.

[11] A. A. Al Ayubbi, S. Widowati, and M. Adrian, “Comparative Performance Analysis of GetX and MobX in Flutter-Based Job Finder Application,” in *2025 3rd International Conference on Software Engineering and Information Technology (ICoSEIT)*, IEEE, Oct. 2025, pp. 1–6. doi: 10.1109/ICoSEIT67010.2025.11290872.

[12] M. Zulistiyan, M. Adrian, and Y. F. A. Wibowo, “Performance Analysis of BLoC and GetX State Management Library on Flutter,” *Journal of Information System Research (JOSH)*, vol. 5, no. 2, pp. 583–591, Jan. 2024, doi: 10.47065/josh.v5i2.4698.

[13] M. H. Anderson and R. K. Lemken, “Citation Context Analysis as a Method for Conducting Rigorous and Impactful Literature Reviews,” *Organizational Research Methods*, vol. 26, no. 1, pp. 77–106, Jan. 2023, doi: 10.1177/1094428120969905.

[14] R. Dekkers, L. Carey, and P. Langhorne, *Making Literature Reviews Work: A Multidisciplinary Guide to Systematic Approaches*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-90025-0.

[15] C. Dignath and M. V. J. Veenman, “The Role of Direct Strategy Instruction and Indirect Activation of Self-Regulated Learning—Evidence from Classroom Observation Studies,” *Educational Psychology Review*, vol. 33, no. 2, pp. 489–533, Jun. 2021, doi: 10.1007/s10648-020-09534-0.

[16] G. M. Fix, B. Kim, M. A. Ruben, and M. B. McCullough, “Direct Observation Methods: A Practical Guide for Health

Researchers," *PEC Innovation*, vol. 1, p. 100036, Dec. 2022, doi: 10.1016/j.pecinn.2022.100036.

[17] E. Bjarnason, F. Lang, and A. Mjöberg, "An Empirically Based Model of Software Prototyping: A Mapping Study and A Multi-Case Study," *Empirical Software Engineering*, vol. 28, no. 5, p. 115, Sep. 2023, doi: 10.1007/s10664-023-10331-w.

[18] A. Siegel, M. Santomauro, T. Dyer, T. Nelson, and S. Krishnamurthi, "Prototyping Formal Methods Tools: A Protocol Analysis Case Study," 2021, pp. 394–413. doi: 10.1007/978-3-030-91631-2_22.

[19] H. Smith, M. Sweeting, T. Morris, and M. J. Crowther, "A Scoping Methodological Review of Simulation Studies Comparing Statistical and Machine Learning Approaches to Risk Prediction for Time-To-Event Data," *Diagnostic and Prognostic Research*, vol. 6, no. 1, p. 10, Jun. 2022, doi: 10.1186/s41512-022-00124-y.

[20] M. Sosnowski, J. Krzywanski, and R. Ščurek, "Artificial Intelligence and Computational Methods in the Modeling of Complex Systems," *Entropy*, vol. 23, no. 5, p. 586, May 2021, doi: 10.3390/e23050586.

[21] O. Alshareet and A. Awasthi, "Collaborative Filtering in The Age of AI: Foundations, Innovations, and Emerging Trends," *Computing*, vol. 107, no. 11, p. 216, Nov. 2025, doi: 10.1007/s00607-025-01564-2.

[22] G.-N. Hu, X.-Y. Dai, F.-Y. Qiu, R. Xia, T. Li, S.-J. Huang, and J.-J. Chen, "Collaborative Filtering with Topic and Social Latent Factors Incorporating Implicit Feedback," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 2, pp. 1–30, Apr. 2018, doi: 10.1145/3127873.

[23] G. Jiang, H. Wang, J. Chen, H. Wang, D. Lian, and E. Chen, "xLightFM: Extremely Memory-Efficient Factorization Machine," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, in SIGIR '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 337–346. doi: 10.1145/3404835.3462941.

[24] Y. Qingmin and C. Xingyu, "Research on Collaborative Filtering Recommendation Algorithm," in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, IEEE, Jan. 2022, pp. 741–743. doi: 10.1109/ICPECA53709.2022.9719134.

[25] J. Zhai, L. Liao, X. Liu, Y. Wang, R. Li, X. Cao, L. Gao, Z. Gong, F. Gu, J. He, *et al.*, "Actions Speak Louder Than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations," in *Proceedings of the 41st International Conference on Machine Learning*, in ICML'24. JMLR.org, 2024.

[26] D. Tikhonovich, N. Zelinskiy, A. V. Petrov, M. Spirina, A. Semenov, A. V. Savchenko, and S. Kuliev, "eSASRec: Enhancing Transformer-based Recommendations in a Modular Fashion," in *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, New York, NY, USA: ACM, Sep. 2025, pp. 1175–1180. doi: 10.1145/3705328.3759317.

[27] A. Akhmad Dinan Jatnika, M. A. Akbar, and A. Pinandito, "Comparative Analysis of the Use of State Management in E-commerce Marketplace Applications Using the Flutter Framework," *Journal of Information Technology and Computer Science*, vol. 8, no. 2, pp. 111–124, Aug. 2023, doi: 10.25126/jitecs.202382557.