Project Report On

Efficient Educational Recommendation System Using Transfer Learning

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Sona Sebastian (U2103200)

Sreya S (U2103203)

Susan Sara Joby (U2103205)

Vaishnavi M (U2103212)

Under the guidance of

Ms. Mehbooba P Shareef

Department of Computer Science
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report titled ***"Efficient Educational Recommendation System Using Transfer Learning"*** *is a bonafide record of the work done by* ***Sona Sebastian(U2103200), Sreya S(U2103203), Susan Sara Joby(U2103205), Vaishnavi M(U2103212)****, submitted to Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2021-2025.*

Ms. Mehbooba P Shareef
Assistant Professor
Dept of Computer Science
RSET

Dr. Sangeetha Jamal
Associate Professor
Dept of Computer Science
RSET

Dr. Preetha K G
Professor & HoD
Dept of Computer Science
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Dr. Jaison Paul Mulerikkal**, Principal of RSET, and **Dr. Preetha K G**, Head of the Department of Computer Science for providing us with the opportunity to undertake our project, "Efficient Educational Recommendation System Using Transfer Learning".

We are highly indebted to our project coordinator, **Dr. Anita John**, Associate Professor, Department of Computer Science, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Mehbooba P Shareef** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Sona Sebastian
Sreya S
Susan Sara Joby
Vaishnavi M

Abstract

This project brings an advanced machine learning-based system that is particularly created to enhance and enrich the learning experience of Massive Open Online Courses (MOOCs). With such widespread variety of courses available, users find it some kind of painful task to come up with courses that best suit their needs for learning. This approach mitigates the problem by offering highly individualized course recommendations based on individual preferences, past trends, and user-specific learning goals. One of the major features of this approach is its innovative application in the field of transfer learning, which aims at enhancing the precision and applicability of the suggestions by making use of pre-trained models from related areas. By integrating algorithms like collaborative filtering, content-based filtering, and transfer learning, the system does not only provide personalized recommendations to the users but will be constantly updating on the basis of actions and reviews from the users. That would mean that the course given to the user would be at maximum appropriateness for engaging activities, retaining the concepts learned, and all-inclusive learning. On the whole, the idea of transferring knowledge from related areas considerably reduces the requirements for big datasets for training and, thus, makes it very efficient and scalable for a large variety of educational environments.

Contents

List of Abbreviations

MOOC - Massive Open Online Courses

NLP - Natural Language Processing

BERT - Bidirectional Encoder Representations from Transformers

XLNet - eXtreme Language Model

MLP - Multi-Layer Perceptrons

CNN - Convolutional Neural Networks

RNN - Recurrent Neural Networks

RMSE - Root Mean Square Error

TL - Transfer Learning

ECG - Electrocardiogram

EEG - Electroencephalogram

HAR - Human Activity Recognition

DAL - Domain Adversarial Learning

LSTM - Long Short-Term Memory

GNN - Graph Neural Networks

GRU - Gated Recurrent Unit

DGSR - Dynamic Graph Neural Network for Sequential Recommendation

DGRN - Dynamic Graph Recommendation Network

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The infusion of technology in the educational arenas has changed the typical learning cycles thus offering new avenues for individualized learning experiences in light of user preferences. Due to the increasing development of content on the internet and accessibility to learning resources on the internet, students are now able to acquire multiple learning materials pertaining to their requirements. However, excess content has made it painful for students to determine which resources were most appropriate for addressing their needs for individual learning. This pain point translates to the need for educational recommender systems, which can guide the student from the right materials, courses, and study plans, aligned with his or her preferences and normal learning patterns. Generally, a recommender system uses algorithms to predict and suggest contents that are most likely to strike the interest of a user. In education, these systems can be optimal for recommending personalized learning resources, courses tailored to a student's learning style, progress, goals, and general taste. By personalizing the learning process, recommender systems can help in improving engagement, efficiency, and academic results. One of the key challenges in developing effective educational recommender systems is the requirement of large amounts of labeled data. The traditional ML models do very badly in scenarios of sparse and/or too-domain-specific data. That is, where transfer learning has its benefits. Transfer learning enables the models to optimize the knowledge gained from one domain/area and apply it to another, often reducing the need for large amounts of domain-specific data. This project hence focuses on building an efficient educational recommender system that leverages transfer learning to enhance the accuracy of recommendations.

1.2 Problem Definition

Challenge of recommending accurate personalized educational content to the students from a vast array of online resources, ensuring that the suggested recommendations match with each student's unique learning needs, preferences, and goals. By using transfer learning, the project aims to improve the accuracy and efficiency of the recommender system, especially in cases with bounded domain-specific data.

1.3 Scope and Motivation

Scope

The scope of this project is to develop a recommendation system that increases the accuracy of recommendation so as to provide personalized educational content to the students. Utilization of transfer learning techniques, enables the system to generate more accurate recommendations. This project excludes the creation of a user interface, concentrating instead on testing the recommendation model using existing user data. the goals comprise the fine-tuning of pre-trained models,improving the system's performance, and testing the possibility of scaling up for offering individualized course recommendations in diverse learning contexts.

Motivation

Educational resources online are multiplying rapidly that makes it a challenge for successfully recommending the relevant content to the students. Notably, with such an enormous volume of information and heterogeneity of needs, it poses a problem for students searching for the right resources that match their individual tastes and objectives of learning. This work is inspired by the imperative to overcome this challenge by utilizing transfer learning to improve the precision and efficiency of recommendations. Fine-tuning pre-trained models on educational data will help to enhance personalized learning experiences, particularly in low-resource scenarios, in order to increase access to education and make learning resources better suited to individual students' needs.

1.4 Objectives

1. **Leverage Transfer Learning**: Fine-tune pre-trained models on domain-specific educational data for improved performance in predicting student preferences using fewer training samples.

2. **Personalized Course Recommendations**: Implement a model that recommends personalized educational material to students through analysis of the unique learning patterns and needs, using user data available.

3. **Measure Effectiveness of the Model**:Compare the efficiency of the fine-tuned models based on comparing model with baselines models with respect to various metrics, like recommendation accuracy and relevance.

4. **Handling Sparse Data**: Obviate sparse domain-specific data by leveraging the utility of pre-existing data while being in the domain-specific data space for accuracy in recommendations.

5. **Real-World Data Testing**: Build and test the recommender system on a smaller dataset of users, which then simulates a real-world setup to determine practical performance.

6. **Model Adaptation and Scalability**: Investigate how to adapt and scale the model for other educational domains so that the system could be generalized for different subject areas or resources than those in the original dataset.

1.5 Challenges

Implementing transfer learning in educational recommender systems is challenging as aligning pre-trained models with domain-specific features of educational data, which often varies in structure and complexity can pose problems. Furthermore, handling time series data demands careful handling of temporal dependencies and right recommendations across diverse learning paths is computationally demanding. Another important barrier is fine-tuning the model to achieve generalization without overfitting.

1.6 Assumptions

- Transfer learning will enable better accuracy on recommendations than when the models were trained solely with educational data.
- Source domain data for pre-trained models are similar enough to educational data so that the knowledge transfer works effectively.
- Time series patterns in student learning behavior are predictable and can be exploited to increase the relevance of recommendations.
- The available educational data is sufficient to fine-tune the model without causing overfitting or introducing bias.

1.7 Societal / Industrial Relevance

A MOOC course recommender system that utilizes transfer learning holds immense value for both society and industry by creating personalized educational pathways in today's rapidly digitalizing world. By guiding users toward courses that best match their skills and goals, it enhances engagement and improves overall learning outcomes.It's a more accessible and inclusive form of education. For industries, this system provides a useful instrument for workforce development. Organizations may design learning experiences aligned with certain skill requirements; companies can stream training, minimize costs, and support continuous development of skills and a more competitive and innovative workforce. In personal and professional contexts, a MOOC recommender system can change education and skill building, bridging the learning gaps, and increasing productivity across society.

1.8 Organization of the Report

The report provides an in depth detailing on the evolution MOOC courses recommender system using transfer learning. Chapter 1 presents the introduction to the project, including the motivations for the project, the problem statement, and the societal and industrial significance of the work. This also details the objectives, scope of the study, and the structure of the report. Chapter 2 discusses the literature review of previous research in recommender systems, transfer learning, and particular issues of recommending online educational content. This chapter provides a basic understanding of the existing methodologies and also discusses how our approach aims to improve upon them. Chapter 3 covers the system design and architecture which includes system's architecture, component design, hardware and software requirements, functional requirements, and the project timeline, thereby providing a detailed understanding of the system's implementation.

Chapter 2

Literature Survey

2.1 Making Recommendations Using Transfer Learning [1]

This paper presents a novel approach to recommender systems by leveraging transfer learning with transformer-based models. Traditional deep learning models require extensive data and training time, which can be reduced by using pre-trained models from Natural Language Processing (NLP) to improve recommendation accuracy and efficiency.

Key Contributions

1. **Transfer Learning in Recommender Systems**:
   - The paper introduces transformer-based models, BERT and XLNet, to encode text-based information (e.g., user reviews, item descriptions) for recommendations.
   - Transfer learning allows for fine-tuning these models on specific recommendation tasks, reducing training time and improving performance over models trained from scratch.
2. **User Vector Embedding**:
   - A new user vector embedding algorithm is proposed to generate user representations when user-specific content is unavailable. This method groups users by similarity in ratings and fine-tunes the randomly initialized user vectors to improve recommendation accuracy.
3. **Recommendation Tasks**:

- The system tackles three recommendation tasks: matrix factorization (predicting user-item ratings), binary recommendation (classifying items as recommended or not), and multi-level recommendation (categorizing user-item interactions into rating levels).

4. **Model Comparison and Results**:

   - The suggested model performs better than other deep learning-based recommendation techniques, such as Multi-Layer Perceptrons (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), according to experimental results on the Amazon and IMDb datasets.
   - XLNet achieved the best performance, surpassing BERT across tasks, likely due to its permutation-based training, which captures more dependencies in the data.

Experimental Findings

1. **Dataset Performance**:

   - Using user vector embedding improved performance in all tasks, indicating the model's ability to represent user similarity effectively.
   - XLNet achieved the lowest RMSE (root mean square error) in matrix factorization and the highest accuracy in binary and multi-level recommendations, showing significant improvements over baseline models.

2. **Attention Mechanisms**:

   - Analysis of attention scores showed that XLNet's ability to focus on more relevant parts of item descriptions contributed to its superior performance compared to BERT.

Conclusion

The paper demonstrates how transfer learning improves accuracy and decreases training time in recommender systems, particularly when data are scarce. The proposed user vector embedding algorithm is a notable advancement for cases lacking user-specific content.

Overall, the transformer-based models used in this work demonstrate a promising direction for enhancing recommendation performance, setting a foundation for future work in recommendation systems utilizing transformer models.

2.2 Novel online Recommendation algorithm for Massive Open Online Courses(NoR-MOOCs) [2]

There are a lot of overwhelming choices when it comes to educational online courses. Most of the time, students get confused about how to choose the right courses. To tackle this problem, we have developed a novel online recommender system for MOOC courses. This system is based on the users' interests, preferences, and the ratings given to courses. Users are named u1, u2, ..., un, and the items are the courses, denoted as i1, i2, i3, ..., in.

The recommender system works based on the concept of hyperspheres. Hyperspheres are basically clusters where similar items or users are grouped together. For example, a group of students with similar marks can be considered a cluster. Each cluster will be referred to as a hypersphere. In this case, the students are the users, and the hyperspheres represent courses such as Machine Learning, Cybersecurity, or Web Development.

Two algorithms are used: the Noor MOOC Training Algorithm and the Noor MOOC Recommendation Algorithm. H is an empty set that stores hyperspheres, while dh is used to track the density. A loop counter, x, is set for iterating through each user. A flag pointer is also used. If the distance between a point and the center of the hypersphere is less than the radius, the point lies within the hypersphere. If the distance is greater, it is outside the hypersphere. If the distance is equal to the radius, the point lies on the circumference of the hypersphere. If a point doesn't belong to any hypersphere, a new hypersphere must be created. The point will become the center, and the radius will be initialized with a default value. This default value can be chosen based on the average radii of all other existing hyperspheres.

We use the Coco dataset for experimental evaluation. Learners who have rated at least 10 courses are considered. The data is split into 80 percent for testing and 20 percent for validation. Root Mean Square Error (RMSE) and coverage are the primary metrics for predictive accuracy. The value predicted by the model is the predicted rating, while the actual rating given by the user is the actual rating. RMSE measures the error between the predicted rating and the actual rating. Coverage refers to the range of courses that can be recommended by the recommender system. Higher coverage means more recommendations. The higher the coverage, the higher the efficiency of the recommender system.

There are many similarity measurement methods we can use for comparison. These methods include PCC, PCCDV, and VCDV. PCCDV is similar to PCC, with the only difference being that missing values are replaced. VCDV refers to the cosine similarity between x and y, where x and y are the user-item pairs. VCCDV is the most accurate method, providing the maximum accuracy. Therefore, we also suggest using VCCDV. The RMSE value should be kept as low as possible for maximum accuracy. The generation size increases as more points are added to the hypersphere. When the generation size increases, the RMSE value decreases. The RMSE value is minimized at a generation size of 3000, so the generation size will be kept close to 3000. The complexity of the algorithm is linear with the number of observations, and the time complexity is O(mn), where m is the average number of ratings by a learner.

The algorithm does have its disadvantages, such as data sparsity and cold start problems. However, its advantages are also commendable. The formation of clusters helps in providing proper recommendations, as similarities can be easily identified. Each cluster represents a similar group. Based on these similarities, different users or students can be assigned different subjects or courses. Thus, it becomes easier for students to choose courses. Additionally, it is superior to many traditional algorithms that have been used for the recommendation process.

—

2.3 Transfer Learning With Time Series Data: A Systematic Mapping Study[3]

2.3.1 Introduction

- Background: Transfer Learning is crucial in fields like computer vision and NLP, where models that are trained on one task are adapted to new , related tasks with sparse data. Recently, TL has been applied to time series data, such as stock prices and medical records. Time series data presents challenges, being sequential in nature and due to temporal dependencies, making TL a valuable tool for handling these challenges effectively.

- Importance in Time Series: Time-series data are often faced with limited labeled datasets, which is either due to the high cost of manually labeling data or the challenges of acquiring consistent data across different environments. TL solves this by allowing models to transfer knowledge from large, labeled datasets in related domains, thus reducing the reliance on labeled data present in the target field. This is highly beneficial in real-time applications, like anomaly detection, healthcare, and financial forecasting, where labeled data may be sparse or difficult to acquire.

- Applications Overview: TL is applied across various fields that are associated with time series data:

  Industrial Fault Diagnosis: In fields like manufacturing and machine maintenance, TL models that are trained on one machine can be adjusted so as to detect faults in others, reducing the need for labeling of fault data.
  Healthcare: TL methods have been optimized to personalize patient monitoring systems, specifically in analyzing ECG or EEG data. This allows models to adapt in accordance with individual patients needs with minimal retraining.
  Financial Forecasting: By using models trained on historical market data, TL can predict trends in new markets, even when the available data for the new environment is limited.
  Human Activity Recognition (HAR): TL helps in transferring knowledge from the source devices or users to the target users, thus simplifying the process of building activity recognition systems across different users and environments.

- Objective: To provide a systematic overview of the TL methods that are applied to time series data, analyze the different methodologies, compare their effectiveness, and identify inevitable challenges such as domain shifts and data scarcity. The main aim of this is to showcase how TL can solve various challenges in time series analysis and can improve the efficiency of models to a large extent.

2.3.2 Methods

Model-Based TL

Approaches like pre training and fine-tuning, that train on the source data and adapt to the target domain with very minimal retraining.

- Pre Training and Fine-tuning: The most commonly used transfer learning approach in time series data. It involves initializing a target model with parameters from a pre-trained model on a source dataset to improve accuracy.

- Partial Freezing: This approach involves freezing certain layers of the pre-trained model while only fine-tuning some of the layers. The advantage is that it saves computational resources and reduces the risk of overfitting when the target domain has sparse data.

- Architecture Modification: This methodology involves a modification of the target model architecture. The aim in this methodology is to adjust some characteristics of the target model. This might entail adding some more layers or readjusting some neurons' quantity to make way for some pertinent features needed by the target. These modifications have the potential of making the models adapt quickly towards new and more complex data.

- Domain Adversarial Learning: DAL trains the model to learn domain-invariant features and aims to minimize the differences between domains. A feature encoder is trained to produce representations that are useful for the predicting task to confuse a domain discriminator, which tries to distinguish between source and target data. Thus, this approach is more robust to domain shifts since the adversarial objective generally forces the feature encoder to focus on shared characteristics between two domains.

Feature-Based TL

Feature Transformation and Learning: This method focuses on transforming the source data and modifying them to adjust well with the target domain, allowing the model to generalize across the domains. Neural networks, like autoencoders, are commonly used to capture relevant features from the source data, which are then used with the target domain.

- Autoencoder-Based Method: Autoencoders compress input data into a latent representation and then reconstructs the original input. In TL, autoencoders are trained to transform time series data from the source domain into representations that can be used for prediction purposes in the target domain. Training strategies like parallel or sequential training are used, depending on whether the source and target data can be shared or need separate training.

Instance-Based TL

This approach involves selecting and re-weighting features from the source domain that are significant to the target task. Instance-based TL can be useful when the source and target domains have similar features but different label spaces. By identifying and using the most relevant instances from the source, the model can achieve better performance with minimal labeled data in the target domain.

2.3.3 Conclusion

Model-based transfer learning approaches are generally most effective for tasks with substantial domain shifts, while feature-based methods work well when domain differences are primarily in feature spaces, and instance-based methods suit tasks where data instances remain relevant across domains. Transfer learning for time series data is applied in fields like fault diagnosis, healthcare monitoring, human activity recognition, and financial forecasting. In each, TL enables adaptation across different machines, patients, devices, or market instruments. Key challenges include managing domain shifts, computational demands for real-time processing, and ensuring interpretability in sensitive areas like healthcare. Addressing these issues requires robust, adaptable TL methods for evolving data streams.

2.4 Dynamic Graph Neural Networks For Sequential Recommendation [4]

2.4.1 Introduction

As the information available online increases, the need for recommender systems plays crucial role in managing information overload in various filelds like e-commerce, search engines, and social media. The traditional method of collaborative filtering captures only the static user-item interactions but thereby neglecting the progressing nature of user preferences. Thus the use of sequential recommendation methods, which utilizes historical interactions, focusses on making the predictions accurate.

There are various approaches for implementing sequential recommendation. Markov chain models which basically rely on limited past interactions, while RNN-based models, such as LSTM and GRU, identifies dependencies within the user sequences. More recent methods like CNNs and attention mechanisms like Caser looks into the sequential order of user interactions. Graph Neural Networks (GNNs) captures complex item relationships but in many cases miss out collaborative signals between the users.

The existing models are bounded in using dynamic collaboration across the users, focusing mostly on encoding individual sequences and ignoring connections between user sequences. Additionally, they struggle to look into the dynamic, high-order relationships that evolve over time which influences the user preferences. To address these limitations, the Dynamic Graph Neural Network for Sequential Recommendation (DGSR) is introduced. DGSR constructs a dynamic graph that integrates time-stamped interactions and links sequences based on shared items. It samples sub-graphs to model long and short-term user preferences and reevaluate next-item prediction as a link prediction task. Experiments preformed on multiple datasets validates the effectiveness of DGSR, thereby emphasizing the importance of using dynamic collaborative signals in sequential recommendation.

2.4.2 Methodology

The DGSR model comprises of several key components that are aimed at predicting user-item interactions in sequential recommendation. Each component plays a distinct role in capturing the user behavior, resulting in a model that can adapts to the dynamic real-world scenarios.

Dynamic Graph Construction

The DGSR model begins by converting user interaction sequences into a dynamic graph. In this graph, interactions of each user with an item is represented as an edge that connects the user node with item node. Each edge is labeled with interaction-specific details such as the exact time of the interaction and the sequence of the interaction between the items. This graph representation captures the evolution of user preferences over time. Unlike a static graph, where the relationships are fixed, the dynamic graph continuously updates when a new interaction occurs, making it suited for time-sensitive recommendations. Example: If a user u_1 has interacted with an item i_1 at time t_1 and later with an item i_2 at time t_2, then there will be edges in the dynamic graph representing both the interactions, with timestamps t_1 and t_2 to maintain the temporal sequences.

Sub-Graph Sampling

To manage the complexity of a large dynamic graph, DGSR applies the sub-graph sampling method. Instead of working with the entire graph, the model selects the relevant sub-graphs for each user sequence. This reduces the computational cost and minimizes the unnecessary noise present allowing the model to focus on meaningful data of each user.

Sub-graph sampling involves selecting the multi-hop neighbors around a particular user's interaction sequence. For example, the model focuses on the user's recent interactions and the limited number of connections to capture the context without overwhelming the network.

This process also incorporates neighbors that has direct connections in the graph and records nodes that have already been used, to avoid redundancy, ensuring that each sampled sub-graph is unique and relevant to the specific user's preferences.

Dynamic Graph Recommendation Network (DGRN)

The DGRN constitutes the primary recommendation engine of DGSR. It fuses long-term and short-term information to obtain user preference through a series of message-passing and aggregation mechanisms. It is these highlighted user preferences that reflect both historical and recent behavior.

Long-Term Information Aggregation

Long-term information is the core component for the overall user preferences. The method combines historical data from multiple interactions over the time, thereby capturing the patterns and trends that can influence the user decisions. The DGRN applies a dynamic attention mechanism that emphasizes the interactions based on their order and significance. For instance, the interactions that has occurred in current time may carry more weight than those in the distant past. The sequence order information helps the network to prioritize the recent interactions.

Short-Term Information Aggregation

The short-term information captures the most recent interests of the user and thus reflects immediate needs and preferences. To build short-term interactions, we aggregate interactions with a special mechanism that directs focus only on the few most recent interactions and their importance for predicting the next action of the user. It is a method of aggregation that relies on self-attention mechanisms, as it compares each such interaction with the previous ones within a recent sequence. For instance, if a user has previously interacted with many items in the same category, the model can detect a trend and exploit it to forecast similar items in the future.

2.4.3 Prediction Layer

As the DGRN encodes the user preferences into long-term and short-term embeddings, these embeddings are then passed on to the prediction layer. This layer is accountable for generating a prediction for the next interaction that is likely to occur, thereby considering the users preferences. The user embedding is generated by concatenating the embeddings from the multiple DGRN layers obtained. Each layer captures different facets of the user's preferences, thereby allowing the prediction layer to have a well-defined view of the user's interests. This final embedding serves as a representation of the user's historical and recent preferences. A link prediction function then scores the candidate items, thereby determining the likelihood that the user will interact with each of the item. The model then rank the items based on these scores and recommends those with the highest likelihood, thus providing recommendations according to the user's unique sequence of interactions.

2.5 Summary and Gaps Identified

2.5.1 Summary

Table 2.1: Comparison of Papers

Paper	Advantages	Disadvantages
[1] Making recommendations using transfer learning Xing Fang. (2020)	Bidirectional context understanding. Pretrained on large corpora, easy to fine-tune.	Slower fine-tuning compared to unidirectional models
[2] Novel online Recommendation algorithm for Massive Open Online Courses (NoR-MOOCs) (2021)	Able to provide personalized course recommendations to users.	Parameter Sensitivity: as the generation size, radius of hyper-spheres. Cold start problems.
[3] Transfer Learning With Time Series Data: A Systematic Mapping Study. (2021)	Simple to implement; leverages knowledge from the source domain; widely applicable.	May struggle with significant domain shifts;

Paper	Advantages	Disadvantages
[4] Dynamic Graph Neural Network for Sequential Recommendation. (2022)	Captures Temporal User Behavior. Efficiently handle dynamic changes.	The complexity of processing large, evolving graph structures and training DGNNs.

2.5.2 Gaps in the Current State of the Art

While several approaches have been developed for educational recommender systems, there are key gaps that limit their effectiveness. These include:

1. **Limited Cross-Domain Knowledge Transfer**: Current models struggle to effectively transfer knowledge across different domains or subjects in educational recommender systems, limiting their ability to adapt to new topics or courses.

2. **Failure to Incorporate Temporal Evolution of Preferences**: Most existing methods overlook how students' preferences evolve over time, especially as they progress through their studies, leading to less accurate recommendations for long-term learning.

3. **Lack of Personalization in Contextual Learning**: While many systems consider student profiles, they fail to capture dynamic, context-specific factors such as real-time performance and immediate learning needs, which are essential for personalized recommendations.

4. **Inability to Handle Data Scarcity in Niche Areas**: In educational settings with limited data on certain subjects or learners, traditional models suffer from data sparsity. Transfer learning could mitigate this by leveraging knowledge from related domains.

5. **Poor Integration of Collaborative Filtering with Sequential Learning**: Existing systems often rely solely on either collaborative filtering or sequential recom-

mendation methods, but fail to integrate both. An effective system should combine collaborative signals with the sequential nature of learning.

2.5.3 Conclusion

Transfer learning has been shown to significantly enhance recommendation systems by enabling the transfer of knowledge from related domains, addressing data scarcity issues, and improving recommendation accuracy. By using preexisting knowledge from source domains, transfer learning allows for the development of personalized recommendation models, even when limited data is available in the target domain. The ability to adapt to dynamic and evolving user preferences is further strengthened, as long-term and short-term preferences can be integrated effectively. Transfer learning helps to overcome challenges such as the cold start problem, where new users lack adequate historical data. Through these capabilities, transfer learning is positioned to provide more robust and scalable solutions for recommendation systems across a wide range of applications.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

- Processor - Intel i7:

  A processor with sufficient computational power is essential for handling machine learning workloads efficiently. An Intel Core i7 processor with at least 8 cores ensures smooth data processing, model training, and inference tasks.

- RAM - 16GB:

  A minimum of 16GB RAM is necessary to handle large datasets and complex machine learning models. This ensures efficient data handling, reduces processing delays, and optimizes performance.

- Storage - 500GB SSD:

  A minimum of 500GB SSD storage ensures quick data retrieval and faster execution of machine learning tasks. SSDs improve read/write speeds, enhancing system responsiveness.

- NVIDIA GPU:

  An NVIDIA Graphics Processing Unit (GPU) is beneficial for accelerating deep learning computations. CUDA-enabled GPUs significantly reduce training time for models by performing parallel processing.

- Internet Connectivity:

  A high-speed internet connection is necessary to ensure smooth data transfer, cloud access, and real-time updates.

- Operating System - Ubuntu:

  The system should run on Ubuntu, ensuring compatibility with required hardware and software configurations.

- Python:

  Python is the primary programming language used for developing machine learning models. Ensure compatibility with required libraries and frameworks.

- Machine Learning Frameworks:

  TensorFlow, Keras, PyTorch, and Scikit-learn are required for building, training, and optimizing recommendation models.

- Database Management:

  A suitable database system should be used for storing user data, course metadata, and interaction logs efficiently.

3.2 Functional Requirements

1. Data Ingestion and Preprocessing

   Load and preprocess datasets by normalizing ratings, handling missing values, and encoding categorical variables.

2. Pre-trained Model Selection and Transfer Learning

   Utilize pre-trained models (LSTM, FNN, MLP) and fine-tune them by freezing all but the last layer for domain adaptation.

3. Model Training and Evaluation

   Train models using collaborative filtering and deep learning, evaluating performance with RMSE, MAE, and computation time.

4. Course Recommendation Generation

   Generate personalized course recommendations based on learner interactions to enhance adaptive learning experiences.

5. System Scalability and Adaptability

   Ensure cross-domain adaptability and allow integration of additional datasets for future scalability.

Chapter 4

System Architecture

4.1 System Overview

Recommender systems are very essential in personalized learning as they facilitate learners to identify relevant courses of interest based on their interests and past interactions. Most conventional recommendation models suffer from limited domain adaptation, which entails intensive training per dataset. In an effort to overcome this weakness, we suggest an effective education recommendation system using transfer learning for enhancing recommendation performance. By utilizing pre-trained models and fine-tuning them across a variety of datasets—Netflix, Goodreads, ML1M, COCO, and Beauty—our model performs more improved generalization as well as cross-domain flexibility and adaptability.

4.2 Architectural Design

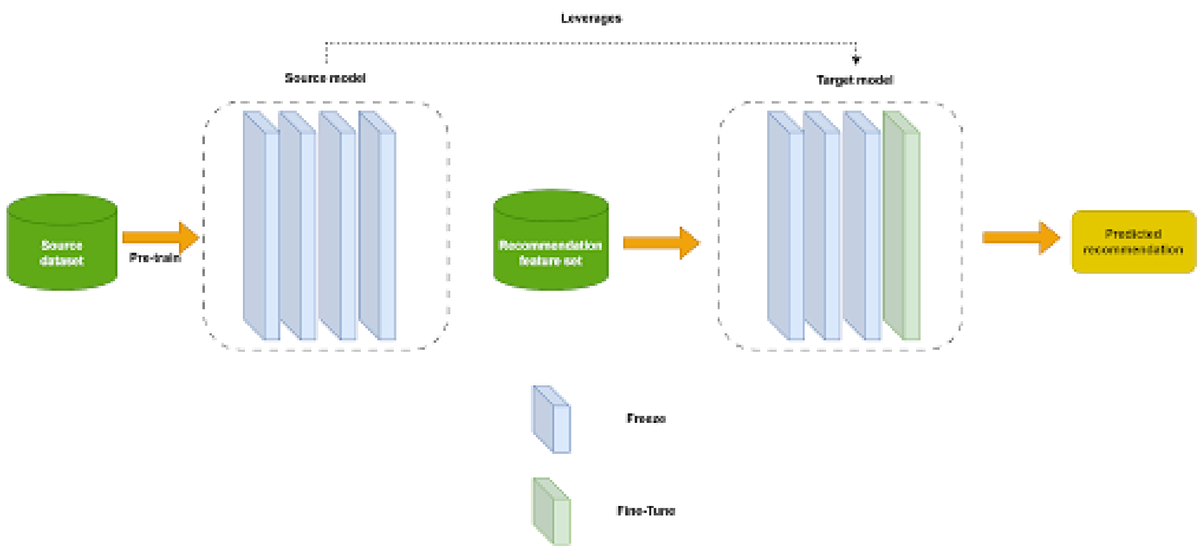


Figure 4.1: System Architecture

Transfer learning is a powerful technique in deep learning where knowledge gained from training on a large dataset is transferred to a related task with a smaller dataset. The given system architecture outlines how transfer learning can be applied to a recommendation system, leveraging pre-trained models to improve recommendation quality and efficiency.

4.2.1 Source Dataset

The process begins with a source dataset, represented as a green cylinder in the diagram. This dataset contains a large volume of labeled data, typically comprising user-item interactions, product details, or any domain-specific information. The dataset serves as the foundation for training an initial deep learning model.

4.2.2 Pre-trained Model

A deep learning model, represented by multiple stacked layers, is pre-trained on the source dataset. This model learns feature representations from the raw input data, capturing meaningful patterns that can be leveraged for recommendation tasks. The output of this model is stored in a recommendation feature set, which serves as an intermediate representation.

4.2.3 Feature Transfer and Fine-Tuning

The extracted features are then transferred to a secondary deep learning network. This stage involves two key processes: All layers of the pre-trained model except the last layer are frozen to retain previously learned knowledge. Other layers are fine-tuned on a smaller, domain-specific dataset to adapt to the new recommendation task. This fine-tuning process ensures that the model not only retains general features but also learns task-specific patterns to enhance performance.

4.2.4 Fine-Tuned Model and Prediction

The second deep learning module, depicted as another stack of layers, processes the refined features to generate recommendations. The final output, shown as a yellow box, represents the predicted recommendations. These could be a ranked list of items, personalized content, or product suggestions tailored to user preferences.

4.2.5 Sequence Diagram

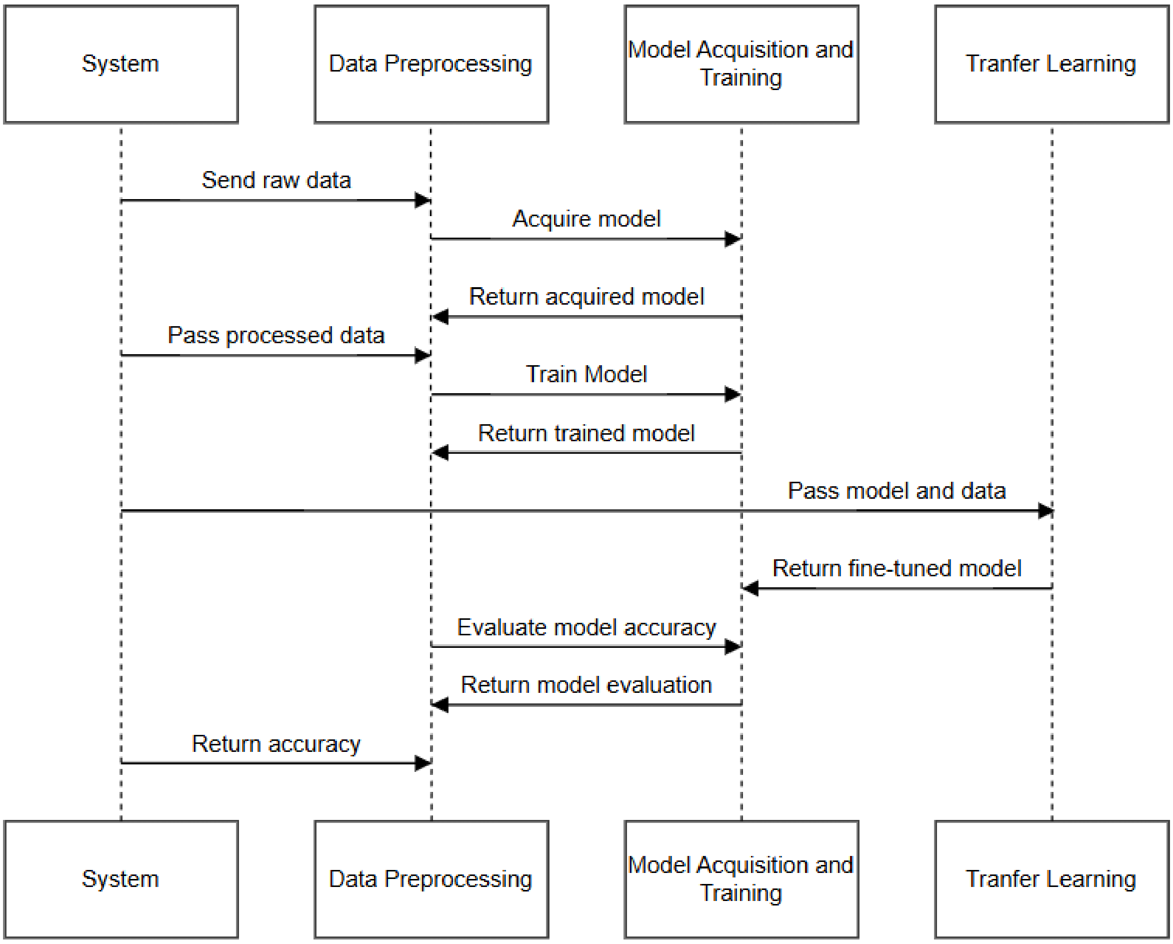


Figure 4.2: Sequence Diagram

4.3 Module Division

1. **Data Preprocessing**

   This module focuses on preparing the raw datasets for model training. It includes normalization of rating values to ensure consistency across datasets, handling of missing values using imputation techniques, and encoding of categorical attributes like course type and genre into numerical format. These preprocessing steps standardize the input data, making it suitable for feeding into machine learning models.

2. **Model Acquisition**

   In this module, suitable pre-trained models are selected for transfer learning. The system uses three models—LSTM, Fully Connected Neural Network (FNN), and Multilayer Perceptron (MLP)—each originally trained on different tasks. These models bring domain-general knowledge that can be adapted for educational recommendations, reducing the need for large amounts of training data in the target domain.

3. **Model Training Using Base Dataset**

   This module involves training the selected models from scratch using the educational datasets without applying transfer learning. Collaborative filtering and deep learning methods are used to predict learner preferences. The models are evaluated using RMSE, MAE, and Time/Step metrics, which establish the baseline performance before applying transfer learning.

4. **Transfer Learning and Fine-Tuning**

   In this module, transfer learning is applied to the pre-trained models by freezing the initial layers and fine-tuning the final layers using the target educational datasets. This enables the models to retain their learned representations while adapting to the specific task of course recommendation. This process enhances model performance while significantly reducing training time and resource usage.

5. **Performance Comparison and Analysis**

   This module performs a comparative analysis of model performance before and after applying transfer learning. Using evaluation metrics like RMSE, MAE, and Time/Step, improvements in accuracy and computational efficiency are highlighted. The effectiveness of transfer learning is validated through this comparison.

4.4 Work Schedule - Gantt Chart

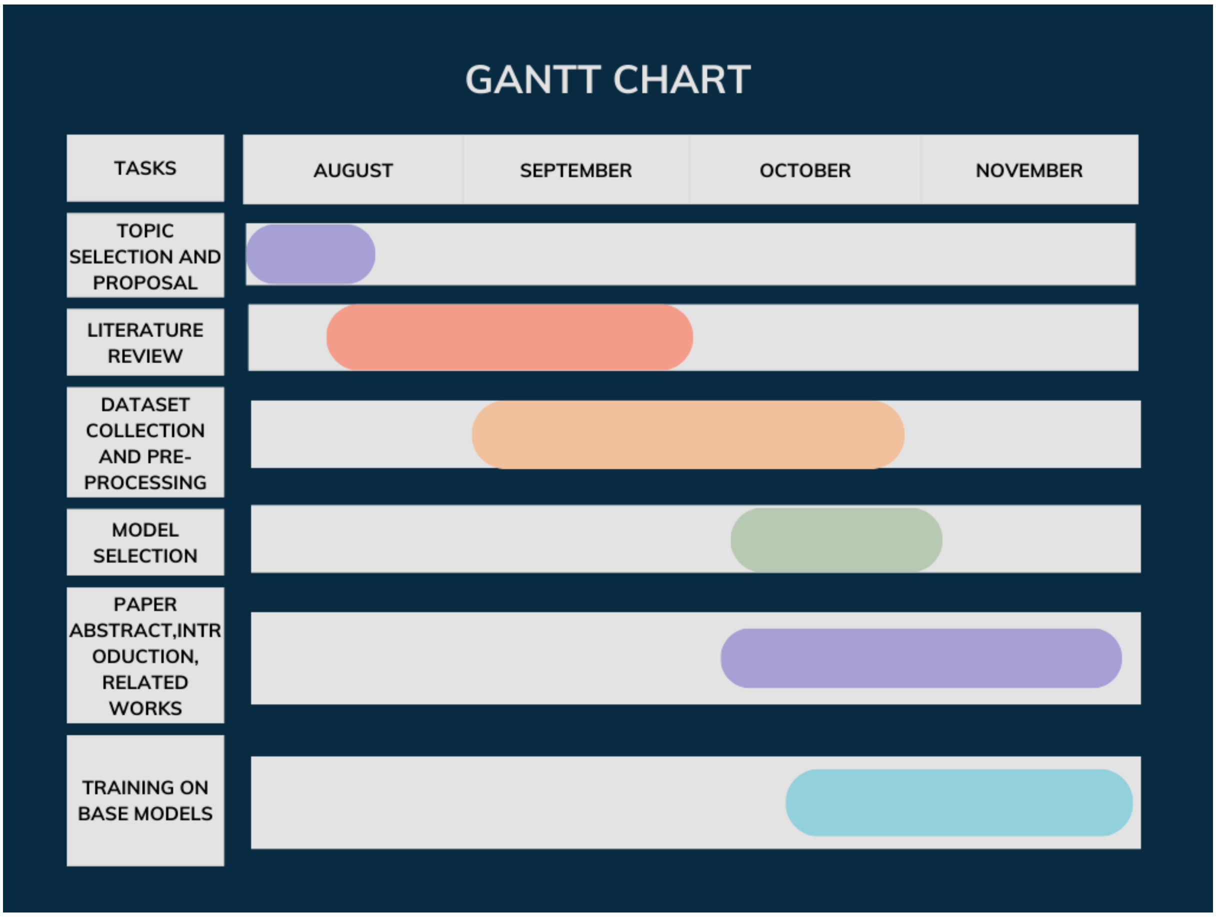


Figure 4.3: Gantt chart

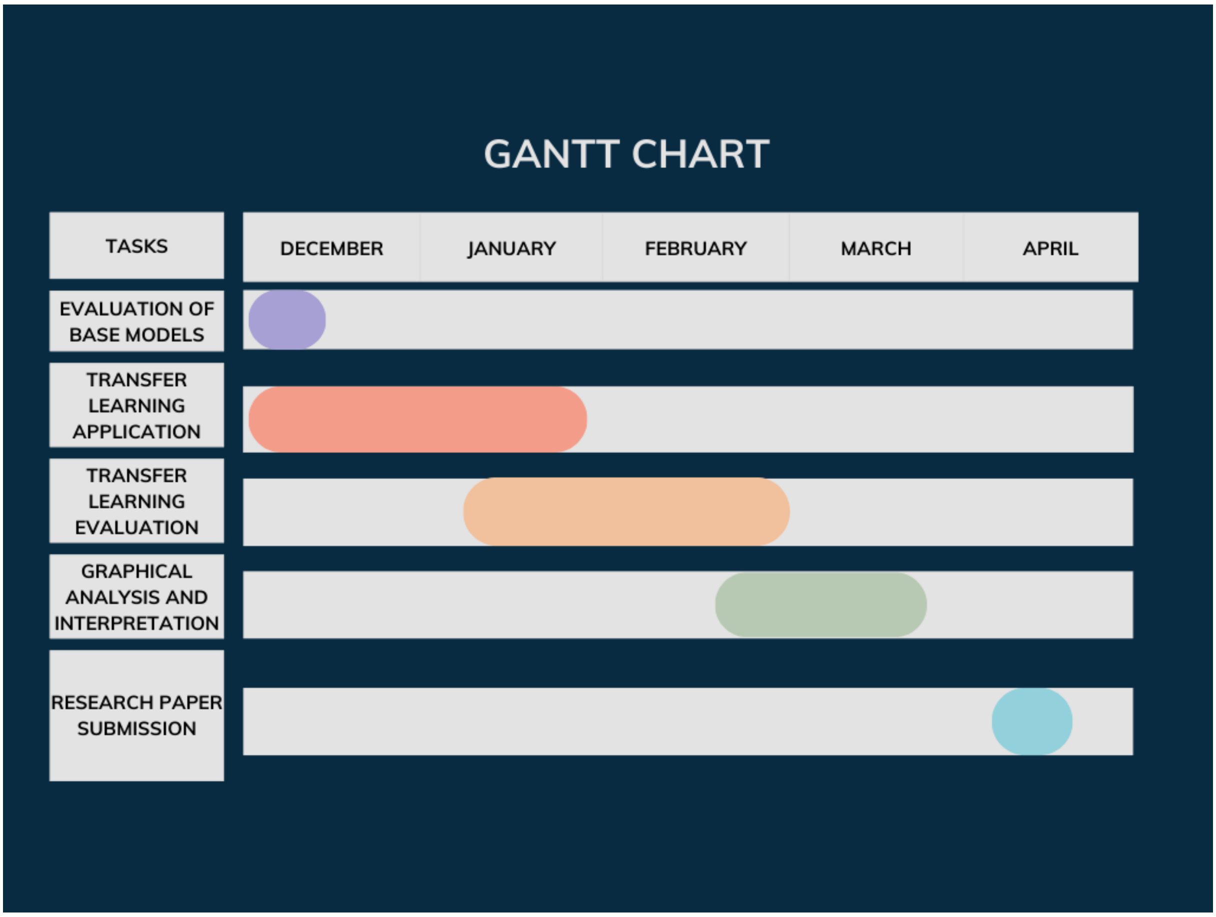


Figure 4.4: Gantt chart

Chapter 5

System Implementation

5.1 Datasets Identified

The foundation of any recommendation system lies in the dataset, which is used to train and evaluate the model. In this system, five distinct datasets from different domains—Netflix, Goodreads, ML1M, COCO, and Beauty—are utilized to capture the diverse nature of learner behavior across various contexts. These datasets contain in- formation about user interactions with items (courses in this case), which includes:

• Learner ID: A unique identifier for learners, allowing the system to track and personalize recommendations based on individual learner behavior.

• Course ID: A unique identifier for courses that learners have interacted with, helping the system to make recommendations based on available content.

• Learner Rating: A numerical value reflecting the learner's feedback on a course, which is essential for evaluating and predicting user preferences.

5.2 Proposed Methodology/Algorithms

The suggested methodology is aimed at creating an educational recommendation system using transfer learning, which improves the performance of recommendations by leveraging pre-trained models on different sets of data. The proposed system is intended to offer personalized course recommendations to the learners based on their previous experiences, interest, and learning scenarios. An step-by-step self-explanatory description of different steps of the proposed system follows below.

5.2.1 Dataset Processing

The different datasets which are being used has to be preprocessed.Under data preprocessing,the below given are the steps to be done in order:

- Normalization: All rating values across the datasets are standardized to a common scale to maintain consistency in how learner preferences are evaluated.
- Handling Missing Data: Missing or incomplete values are addressed through imputation or interpolation techniques, ensuring that no essential information is lost.
- Categorical Encoding: Non-numerical data, such as genre, course type, and other categorical variables, are encoded into numerical representations to ensure compatibility across datasets.

5.2.2 Pre-trained Model Selection and Transfer Learning

Once the dataset is pre-processed and prepared, the next step is the selection and application of pre-trained models through transfer learning. Transfer learning leverages models that have already been trained on large, diverse datasets, enabling them to be adapted to the target task (in this case, educational recommendations) with less data and computation than training a model from scratch.

The system utilizes three key pre-trained models:

- LSTM (Long Short-Term Memory): Suitable for making predictions based on time-series data, useful for modeling sequences of learner interactions with courses.
- Fully Connected Neural Network (FNN): A feedforward network used for tasks like air quality index prediction, which can be adapted to make predictions on educational data.
- Multilayer Perceptron (MLP): A deep learning model used for tasks like house price prediction, adaptable for predicting ratings or course preferences in this case.

Transfer Learning: In transfer learning, most of the pre-trained model layers are frozen (i.e., their weights are fixed), except for the last few layers. This allows the model to retain general knowledge learned from the original task, while adapting to the new task by only fine-tuning the final layers. Fine-tuning involves retraining the last layers of the pre-trained models using the target dataset (educational data), allowing the system to adjust to the specific domain of recommendation. By using this approach, the system benefits from reduced training time and the ability to perform well with less data, making it both time-efficient and resource-efficient.

5.2.3 Model Training and Evaluation

The models are trained using a combination of collaborative filtering and deep learning techniques, which are the core method used in recommendation systems. The model training involves going through the training data where the model is being trained to forecast learner preference based on previous interactions.

Performance Metrics: To gauge the quality of the models, the following performance metrics are used:

- Root Mean Squared Error (RMSE): A widely used metric that calculates the difference between the actual ratings and the predicted ratings, which can be used in calculating how good the model is at predicting.

- Mean Absolute Error (MAE): The metric compares the average absolute difference in actual and predicted ratings, which provides information on the magnitude of errors.

- Time/Step (Computation Efficiency): Average time per training step. This is important in determining the computational efficiency of the model, especially when real-time output is an issue.

Baseline Evaluation: The models are initially tested on the target datasets without transfer learning. RMSE, MAE, and time/step are recorded to compare against to see the performance of models trained from scratch.

Transfer Learning Application: The previously trained models are re-tuned after applying transfer learning. The models are tested again using the same metrics to locate the change resulted by transfer learning

Comparison and Analysis: Comparison of models' performance before and after applying transfer learning, on the basis of improvement in RMSE, MAE, and time/step is supported by the system.

5.2.4 Advantages of the Proposed System

The system proposed has various merits which make it a robust and effective solution for educational recommendations:

- Increased Accuracy: Utilizing transfer learning as well as pre-trained models, the system will be able to get more accurate results on various datasets than conventional recommendation systems.

- Decreased Training Time: Utilizing pre-trained models decreases training time and computational resources needed for training substantially, making it an efficient solution for real-time recommendations.

- Cross-Domain Generalizability: The framework is able to generalize over domains, i.e., Netflix, Goodreads, and ML1M, so that the system may find applicability across a broad range of learning environments.

- Extensibility: The framework itself is easily extendable, with room for adding new data sets and domains in the future without radical changes to architecture.

This approach delivers a robust and flexible framework for constructing an efficient and reliable educational recommendation framework that provides personalized and interactive learning to all users.

This methodology provides a comprehensive and adaptable framework for building an educational recommendation system that is both accurate and efficient, ensuring a personalized and engaging learning experience for all users.

5.3 Description of Implementation Strategies

5.3.1 Experimental Setup

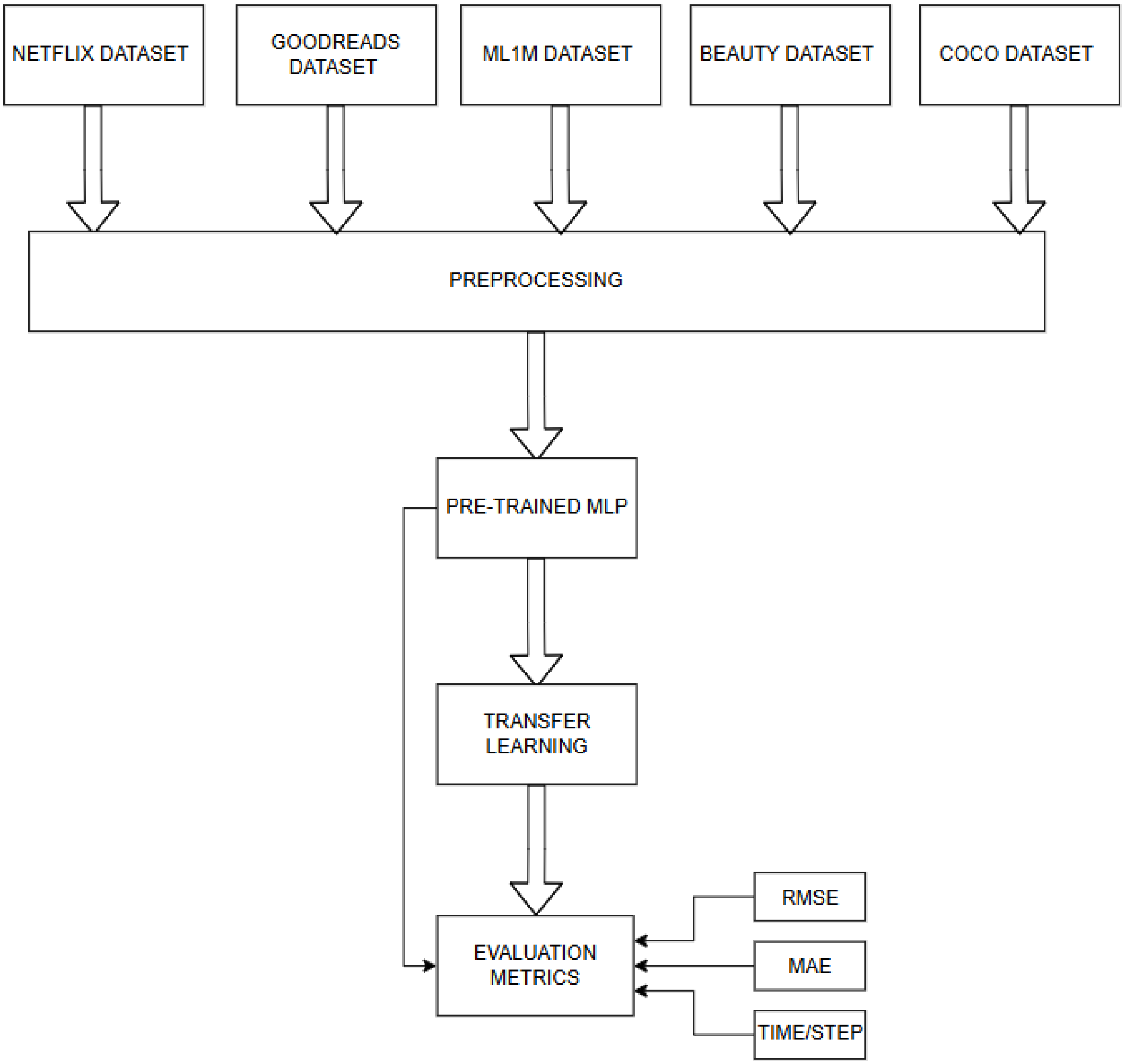


Figure 5.1: MLP Architecture

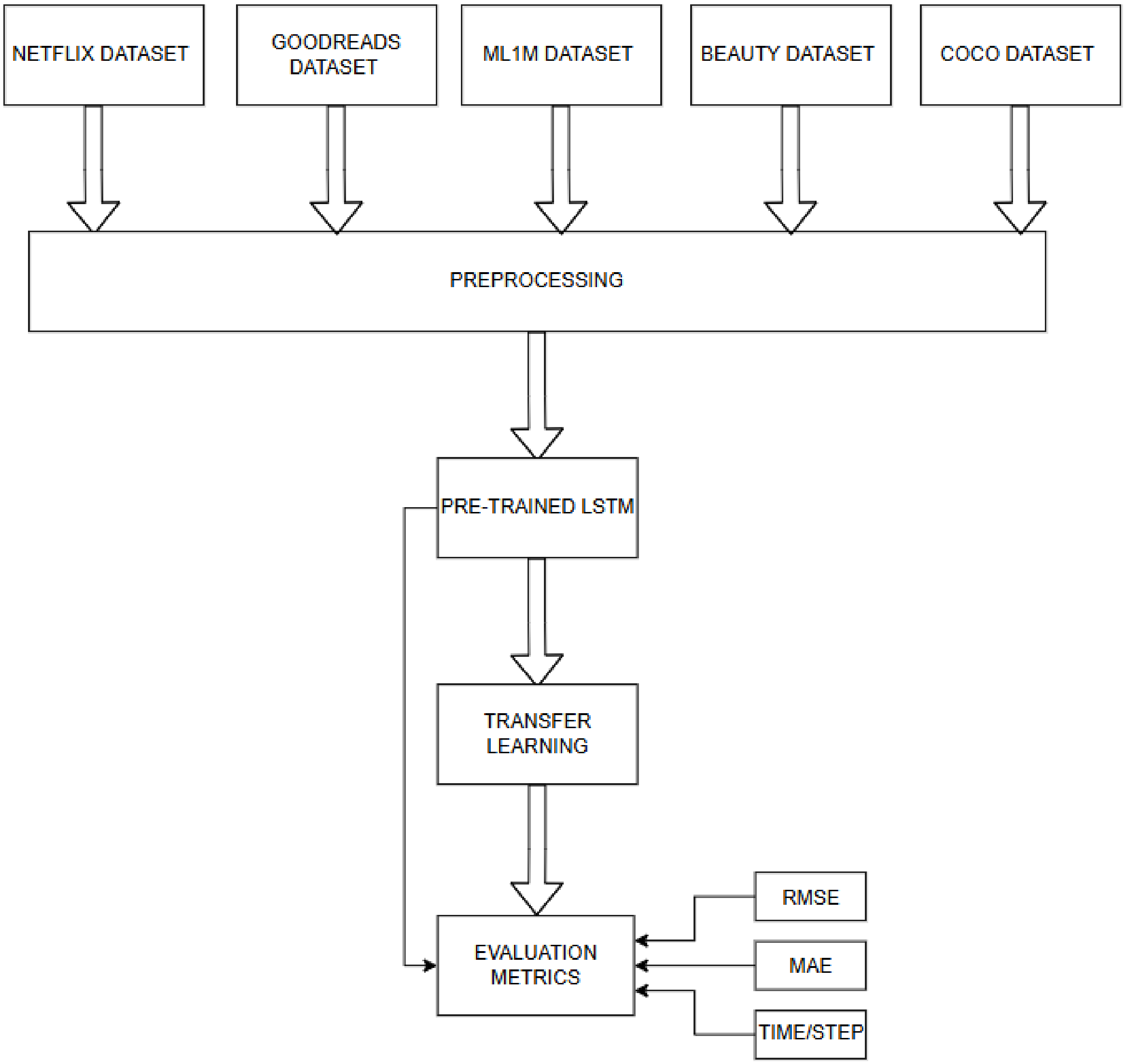


Figure 5.2: LSTM Architecture

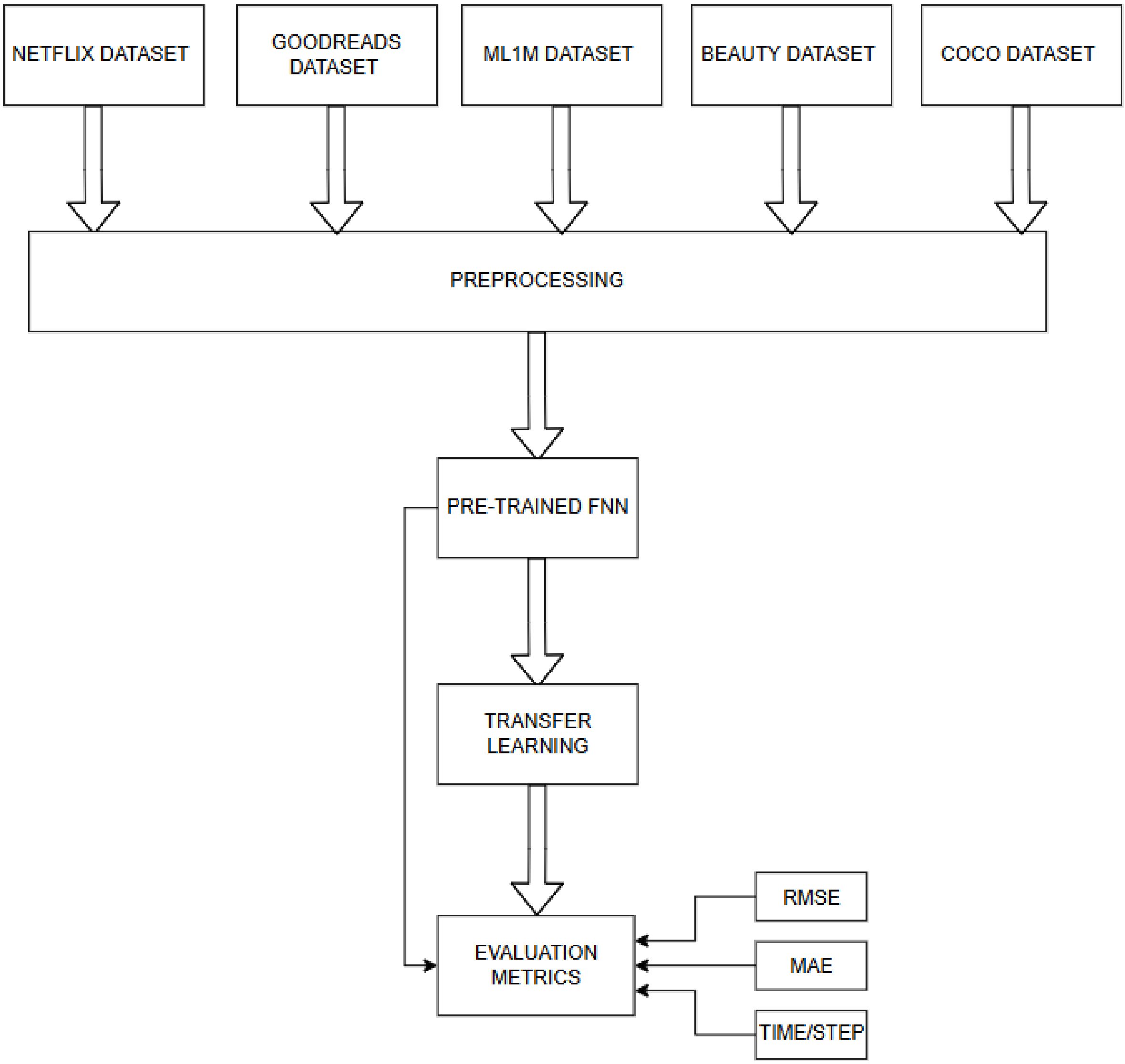


Figure 5.3: FNN Architecture

5.3.2 Implementation

The application of this research is done in a sequential and systematic process to assess the effect of transfer learning on pre-trained models across five varying datasets. The main aim is to identify if transfer learning enhances model performance as well as saves time computationally. The process is structured such that all testing processes are implemented in a sequential process. The steps involved, including dataset preparation, model selection, training processes, transfer learning application, and performance testing.

5.3.3 Selection of Pre-Trained Models

Pre-trained models are the foundation of this work because they have been trained on abundant data and have learned useful representations. Choosing the models is also important in the sense that it is used to guarantee that the experiments cover a broad range of applications. Models employed in this research are:

- Long Short-Term Memory (LSTM): Form of recurrent neural network (RNN) structure useful for handling sequence and time-series data. It remembers long-term dependencies in data, and therefore appropriate for uses such as stock price prediction and text-based analysis.

- Multi-Layer Perceptron (MLP): Feedforward network with full connectivity suitable for structured data. It performs best with tabular data and is frequently used in classification and regression tasks.

- Feedforward Neural Network (FNN): A broad deep learning architecture that is widely employed in application involving dense feature representations. Each model is compared before and after using transfer learning to see its effect on computational efficiency and predictive accuracy.

5.3.4 Dataset Preparation and Preprocessing

Five datasets are selected for experimentation, with variation in data types. The datasets are chosen to span multiple domains, such as time-series, structured numerical data, and categorical features. Before training, every dataset is preprocessed to ensure compatibility with the models selected.

5.3.5 Preprocessing Steps

- Data Cleaning: Missing values are treated with imputation methods, duplicate entries are deleted, and data inconsistencies are resolved.

- Feature Engineering: Appropriate features are derived, and categorical information is translated into numerical values with the help of one-hot encoding or embedding methods.

- Standardization and Normalization: Number values are converted to a comparable range to avoid bias caused by different scales.
- Train-Test Split: Each dataset is partitioned into training and testing subsets (typically 80-20 split) to evaluate model generalization.

These steps ensure that the data is well-structured and prepared for input into the selected models.

5.3.6 Model Training

To establish baseline performance, each pre-trained model is initially trained on the datasets without transfer learning. This step allows for the collection of initial evaluation metrics, such as RMSE, MAE, and computational time per step.

5.3.7 Transfer Learning Application

Transfer learning is a technique that allows models to use previously learned knowledge from large-scale datasets to improve their performance on new, smaller datasets. Instead of training a deep learning model from scratch, a pre-trained model is utilized as a starting point. This reduces computational costs and speeds up training while maintaining high accuracy.

Feature Extraction

Feature extraction is a crucial step in transfer learning, where the lower layers of a pre-trained model are frozen, meaning their parameters remain unchanged. The rationale behind this approach is that these lower layers have already learned useful generic features such as the basic patterns (in the case of text or tabular models). By retaining these pre-trained representations, the model avoids the need to relearn fundamental data properties, thereby reducing training time and computational requirements. When performing feature extraction, the pre-trained model is first loaded, and all its layers except the final classification or regression layers are locked. These frozen layers act as a feature extractor, transforming raw input data into a structured format suitable for higher-level decision-making. The extracted features are then passed to the newly added layers, which will be trained specifically for the new task.

Fine-Tuning the Last Layer

Fine-tuning involves replacing and retraining the last layer of the model with new ones tailored to the target dataset. Since the final layer of a pre-trained model is typically specific to the original dataset it was trained on, it must be replaced with a new layer that aligns with the number of categories or output types in the new dataset. During this step, the new layers are initialized randomly and trained on the target dataset. Unlike the frozen layers, which remain unchanged, these final layers are adjusted through backpropagation to adapt the model to the new dataset. This step allows the model to specialize in the task at hand while still benefiting from the pre-trained features extracted by the earlier layers.

Optimization Adjustments

When applying transfer learning, optimization parameters such as the learning rate must be carefully adjusted to ensure effective fine-tuning. Since the early layers contain valuable knowledge that should not be significantly altered, using a small learning rate prevents drastic changes to pre-trained weights while still allowing incremental improvements. A too-high learning rate could cause the model to forget the previously learned information, a phenomenon known as catastrophic forgetting, while a too-low learning rate may result in slow convergence. Therefore, it is common practice to use different learning rates for different layers—keeping a very low learning rate for the frozen layers and a slightly higher learning rate for the newly added layers. In some cases, after initial training, some of the previously frozen layers can be gradually unfrozen and trained with a very low learning rate. This further fine-tunes the model without losing essential pre-learned features.

5.3.8 Conclusion

By implementing transfer learning, the model leverages previously learned representations, reducing training time and improving performance. Feature extraction ensures that foundational knowledge is retained, fine-tuning enables adaptation to new tasks, and optimization adjustments refine the learning process. This structured approach ensures that models trained with transfer learning outperform models trained from scratch in both efficiency and accuracy.

This approach allows the model to generalize effectively to new datasets while significantly reducing training time. The primary objective of this project is to demonstrate how transfer learning can reduce the overall training time and the time required per step compared to the original pre-trained models.

5.4 Performance Evaluation Metrics

To quantify the impact of transfer learning, model performance is assessed using three key evaluation metrics:

5.4.1 Root Mean Squared Error (RMSE)

Measures the average deviation between predicted and actual values. Lower RMSE indicates better predictive accuracy.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.1}$$

5.4.2 Mean Absolute Error (MAE)

Evaluates the average magnitude of prediction errors, providing insight into model consistency.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.2}$$

5.4.3 Time per Step

Tracks computational efficiency by measuring the time taken for each training iteration.

These metrics are recorded for both the baseline models and transfer learning models to assess performance improvements.

5.5 Comparative Analysis

A comparative analysis is performed to evaluate the advantages of applying transfer learning. The key aspects of comparison include:

- Reduction in RMSE and MAE: Assessing whether transfer learning improves prediction accuracy.
- Reduction in Training Time: Identifying the decrease in computation time per step.
- Impact on Different Datasets: Understanding the effectiveness of transfer learning across various dataset domains.

Graphical representations, such as line charts and bar graphs, are used to illustrate the improvements observed in performance metrics.

5.6 Conclusion

The implementation strategy follows a structured workflow designed to evaluate the effects of transfer learning. The key steps include:

1. The execution plan is in the form of a structured process that is coded to analyze the effect of transfer learning. The major steps are:
    - Selecting the appropriate pre-trained models that span across a broad range of domains.
    - Pre-processing and aligning the datasets into a compatible shape.
    - Establishing the baseline performance through initial training.
    - Using transfer learning by freezing the bottom layers and fine-tuning the last layer.
    - Comparing the performance of the model in terms of RMSE, MAE, and computation per step.
    - Analyzing the results to identify improvements in precision and computation.

Transfer learning has been pursued in this study to demonstrate how the performance of the model is improved by realizing excellent computational cost and time per step savings compared to pre-trained models. The results give important insights into real world advantages of transfer learning in machine learning tasks.

Chapter 6

Results and Discussions

6.1 Overview

This part offers comparative performance of LSTM-1, LSTM-2, FNN, and MLP models before and after transfer learning. The testing was performed on five data sets with the help of three principal measures: Root Mean Squared Error(RMSE), Mean Absolute Error (MAE), and inference time per step (TIME/STEP). Low values of RMSE and MAE represent high accuracy in prediction,whereas low TIME/STEP represents faster execution efficiency.

- **Dataset 1:** LSTM-2 produced the best RMSE (0.93), but was competitive with MLP and FNN, which had faster inference times.
- **Dataset 2:** MLP produced best performance with RMSE of 0.41 and MAE of 0.25, outperforming all models.
- **Dataset 3**: MLP once again generated the lowest RMSE (0.54) and MAE (0.39), demonstrating good generalization. •
- **Dataset 4:** LSTM-2 just surpassed others in terms of accuracy (RMSE: 1.00), though it was slower for inference.
- **Dataset 5:** FNN had the best accuracy (RMSE: 0.78), while MLP and LSTM-2 followed closely.

Overall, both MLP and LSTM-2 were outstanding, but the MLP model had the edge. It was the highest accuracy out of the main datasets, had the lowest error margins in all the tests, and consistently beat its previous performances when it came to execution speed. This renders MLP the most effective and efficient model overall for the task.

6.2 Quantitative Results

Table 6.1: Performance Metrics for LSTM-1, LSTM-2, FNN, and MLP Models Across Datasets

Dataset	Before Transfer Learning	After Transfer Learning
Dataset 1	**LSTM-1:** RMSE: 0.8851, MAE: 0.7301, TIME/STEP: 11ms **LSTM-2:** RMSE: 0.84, MAE: 0.65, TIME/STEP: 10ms **FNN:** RMSE: 0.97, MAE: 0.77, TIME/STEP: 656us **MLP:** RMSE: 0.96, MAE: 0.79, TIME/STEP: 847us	**LSTM-1:** RMSE: 0.9876, MAE: 0.7754, TIME/STEP: 3ms **LSTM-2:** RMSE: 0.93, MAE: 0.74, TIME/STEP: 2ms **FNN:** RMSE: 0.97, MAE: 0.77, TIME/STEP: 509us **MLP:** RMSE: 0.96, MAE: 0.78, TIME/STEP: 491us
Dataset 2	**LSTM-1:** RMSE: 0.3491, MAE: 0.2456, TIME/STEP: 10ms **LSTM-2:** RMSE: 0.63, MAE: 0.35, TIME/STEP: 5ms **FNN:** RMSE: 0.68, MAE: 0.34, TIME/STEP: 658us **MLP:** RMSE: 0.37, MAE: 0.17, TIME/STEP: 526us	**LSTM-1:** RMSE: 0.5373, MAE: 0.4293, TIME/STEP: 3ms **LSTM-2:** RMSE: 0.73, MAE: 0.57, TIME/STEP: 2ms **FNN:** RMSE: 0.96, MAE: 0.70, TIME/STEP: 437us **MLP:** RMSE: 0.41, MAE: 0.25, TIME/STEP: 536us
Dataset 3	**LSTM-1:** RMSE: 1.721, MAE: 1.42966, TIME/STEP: 10ms **LSTM-2:** RMSE: 0.97, MAE: 0.86, TIME/STEP: 7ms **FNN:** RMSE: 0.89, MAE: 0.76, TIME/STEP: 742us **MLP:** RMSE: 1.68, MAE: 1.48, TIME/STEP: 860us	**LSTM-1:** RMSE: 2.0208, MAE: 1.9150, TIME/STEP: 3ms **LSTM-2:** RMSE: 0.98, MAE: 0.92, TIME/STEP: 2ms **FNN:** RMSE: 0.93, MAE: 0.83, TIME/STEP: 669us **MLP:** RMSE: 0.54, MAE: 0.39, TIME/STEP: 417us
Dataset 4	**LSTM-1:** RMSE: 1.228, MAE: 1.021, TIME/STEP: 11ms **LSTM-2:** RMSE: 1.15, MAE: 0.93, TIME/STEP: 6ms **FNN:** RMSE: 0.97, MAE: 0.79, TIME/STEP: 617us **MLP:** RMSE: 1.13, MAE: 0.95, TIME/STEP: 692us	**LSTM-1:** RMSE: 1.0461, MAE: 0.8397, TIME/STEP: 3ms **LSTM-2:** RMSE: 1.00, MAE: 0.83, TIME/STEP: 2ms **FNN:** RMSE: 1.02, MAE: 0.85, TIME/STEP: 556us **MLP:** RMSE: 1.14, MAE: 0.95, TIME/STEP: 506us
Dataset 5	**LSTM-1:** RMSE: 1.0269, MAE: 0.822, TIME/STEP: 10ms **LSTM-2:** RMSE: 0.82, MAE: 0.66, TIME/STEP: 7ms **FNN:** RMSE: 0.78, MAE: 0.62, TIME/STEP: 737us **MLP:** RMSE: 0.94, MAE: 0.71, TIME/STEP: 802us	**LSTM-1:** RMSE: 1.0865, MAE: 0.8981, TIME/STEP: 3ms **LSTM-2:** RMSE: 1.2, MAE: 1.06, TIME/STEP: 2ms **FNN:** RMSE: 0.78, MAE: 0.62, TIME/STEP: 659us **MLP:** RMSE: 1.28, MAE: 1.02, TIME/STEP: 564us

6.3 Graphical Results

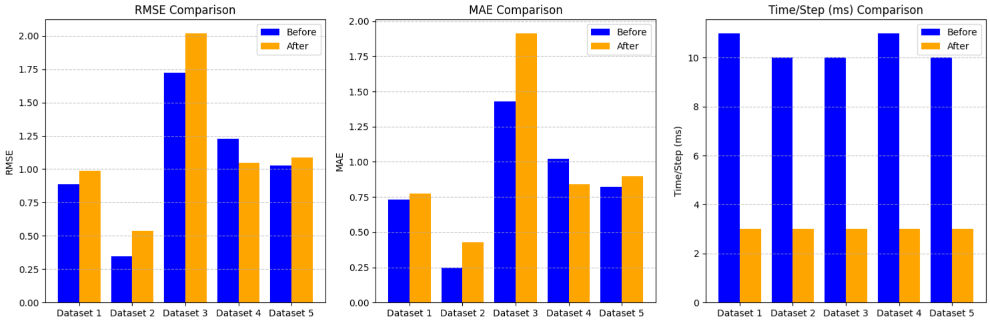


Figure 6.1: Graphical performance results of LSTM-1 model

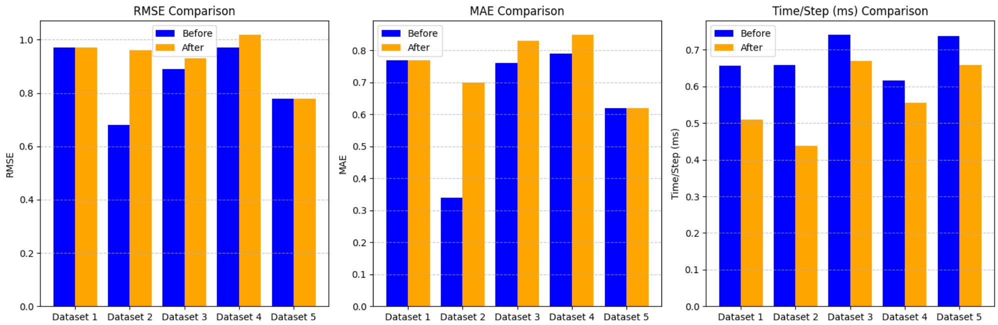


Figure 6.2: Graphical performance results of FNN model

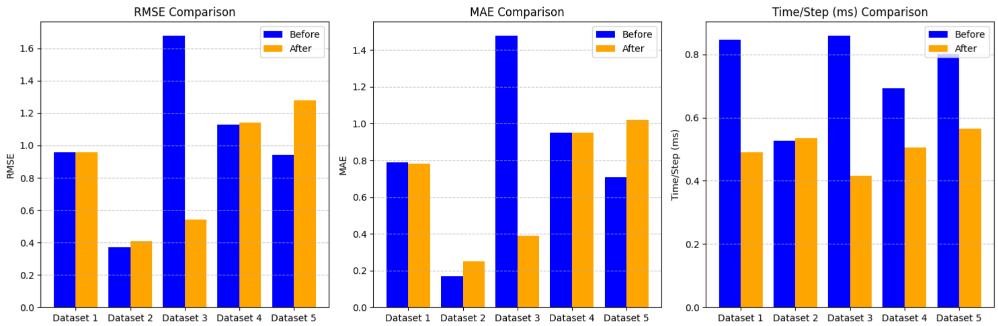


Figure 6.3: Graphical performance results of MLP model

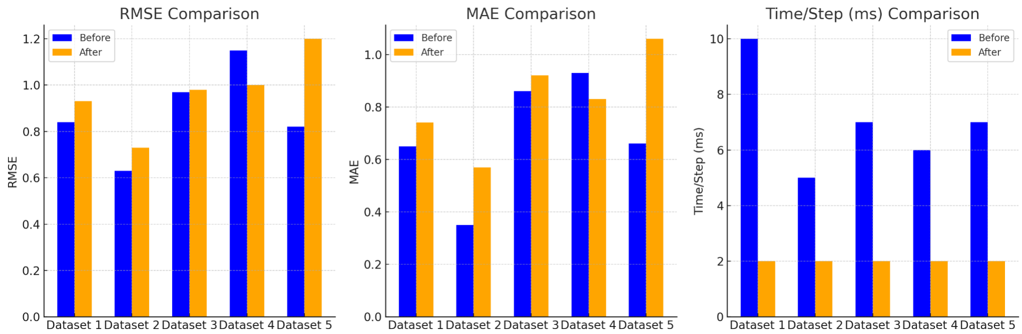


Figure 6.4: Graphical performance results of LSTM-2 model

6.4 Discussion

The performance results on the five datasets reveal some significant trends and observations. Transfer learning did not uniformly affect different models and datasets initially. It introduced improvements to certain metrics in certain instances (e.g., lower inference time for LSTM-2), but it introduced extra errors in certain instances, especially for LSTM-based models.

From the results of post-transfer learning, FNN model emerges as the most consistently strong performer on all datasets with the lowest RMSE and MAE values. Although MLP showed faster inference times in some instances, the difference was minor compared to the better accuracy performance of FNN.

FNN does a fine balance between providing consistent accuracy and efficient execution, particularly doing well in Dataset 5 with the best RMSE (1.02) and on-par speed. As for comparison, the LSTM variants performed inconsistently and always had higher inference time, which makes them less suitable for real-time systems. Overall, FNN is the top performing model in terms of generalization, accuracy, and computational practicability after transfer learning.

In general, the findings indicate that transfer learning is a useful solution but its performance is highly reliant on model architecture and data characteristics. Accuracy versus execution time trade-offs also need to be considered when choosing a model to deploy.

6.4.1 Limitations

One of the most severe deficiencies of this research is that transfer learning had a varying impact on different models. While MLP was improved with transfer learning, some of the other LSTM variations suffered from reduced accuracy performance even with improved inference speed. Another deficiency is that testing was limited to five data sets, and it is not clear whether these would be representative of all possible data patterns or complexities. In addition, very minimal hyperparameter tuning was undertaken to sustain consistency between experiments, which may have restricted the optimum capacity of some models.

6.4.2 Future Work

Subsequent studies can explore more advanced transfer learning methods, such as domain adaptation or fine-tuning in a layer-wise manner, in an effort to generalize model flexibility across different datasets. Combining other datasets with different time-series patterns would provide a better evaluation. Research on hybrid models that leverage the power of different architectures (e.g., ensemble of LSTM-MLP) can also result in improved accuracy and generalization. Finally, adding feedback loops or in-place retraining might provide additional performance gain in dynamic environments.

Chapter 7

Conclusions & Future Scope

The intention behind the project was to perform a comparison of various model for educational recommender system which would be capable of making accurate predictions using transfer learning. The system was capable of performing better in recommending suitable resources for learning and even better for domain specific resources by utilizing model training and fine tuning.The utilization of transfer learning addressed the issue of time management by utilizing pretrained model and utilize the resources effectively. The evidence pointed out the system's capability to assist learners in making well-informed decisions by suggesting relevant and suitable content, indicating promising directions for providing personalized education solutions.Further research and more work is required to be able to extend the system to encompass broader types of educational ranges and other learner types. Feedback and real time user interaction ought to assist in maximizing the recommendation as well.There is also a chance that adding additional features to the system, like using videos and audio, would enhance its functionality. Furthermore, utilization of approaches such as federated learning may enhance privacy problems that include having several institutions using the system without compromising data privacy.

References

[1] X. Fang, "Making recommendations using transfer learning," *Neural Comput. Appl.*, vol. 33, no. 15, p. 9663–9676, Aug 2021.

[2] A. Khalid, K. Lundqvist, A. Yates, and M. A. Ghzanfar, "Novel online recommendation algorithm for massive open online courses (nor-moocs)," *PLOS ONE*, vol. 16, no. 1, p. e0245485, 2021.

[3] M. Weber, M. Auch, C. Doblander, P. Mandl, and H.-A. Jacobsen, "Transfer learning with time series data: A systematic mapping study," *IEEE Access*, vol. 9, pp. 165 409–165 432, 2021.

[4] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, "Dynamic graph neural networks for sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022.

Appendix A: Presentation

EFFICIENT EDUCATIONAL RECOMMENDER SYSTEM USING TRANSFER LEARNING

GUIDED BY:
Ms. Mehbooba Shereef

PRESENTED BY:
Sona Sebastian-U2103200
Sreya S-U2103203
Susan Sara Joby-U2103205
Vaishnavi M-U2103212
S8 CSE GAMMA(2021-25)

CONTENTS

INTRODUCTION

Our project focuses on evaluating the effectiveness of **transfer learning** in improving the performance of deep learning models. We implement and compare four different models across multiple datasets, analyzing changes in **RMSE**, **MAE**, and **inference time per step**. The aim is to demonstrate how transferring knowledge from a source domain can enhance model accuracy and efficiency in a target domain with limited data.

PROBLEM STATEMENT

Current recommendation systems struggle with efficiently adapting to new user preferences, lack personalized recommendations tailored to diverse student needs, and face challenges in improving accuracy over time without extensive retraining.

PURPOSE AND NEED

In today's data-driven educational environment, students are often overwhelmed by the range of course options. A recommendation system, fine-tuned with course-specific data, can provide personalized suggestions in less time, helping students make informed decisions while improving their overall learning experience.

PROJECT OBJECTIVE

Fine-tune pre-trained models on course-specific data for accurate recommendations.

- Analyze and Compare Machine Learning Models for Educational Recommendation
- Investigate the Impact of Transfer Learning
- Optimize Computational Efficiency

NOVELTY AND INNOVATIVENESS

- **Comprehensive Multi-Model Evaluation**
Applied transfer learning across four different models to assess generalizability.
- **Cross-Dataset Performance Analysis**
Tested on multiple datasets to study domain transfer effectiveness.
- **Speed vs. Accuracy Trade-off Exploration**
Highlighted inference time reduction (up to 80%) with minimal accuracy loss.
- **Real-Time Application Feasibility**
Achieved inference speeds as low as 2ms — suitable for time-sensitive tasks.
- **Unified Benchmarking Approach**
Compared RMSE, MAE, and Time/Step before and after transfer learning.

SCOPE OF IMPLEMENATION

Model Adaptability
Transfer learning enables rapid adaptation of models to new but related datasets with minimal retraining.
•**Real-Time Applications**
Optimized inference speed (as low as 2ms) supports deployment in time-critical systems.
•**Resource-Constrained Environments**
Reduced computational cost makes the models suitable for edge devices and low-power platforms.
•**Scalable Evaluation Framework**
Methodology can be extended to additional models, datasets, or domains for broader research.

LITERATURE SURVEY

PAPER	DATASET	METHODOLOGY	RESULT	ADVANTAGES	DISADVANTAGES
[1] Dynamic Graph Neural Network for Sequential Recommendation. (2022)	Amazon: three categories,Amazon CDs, Amazon-Games, Amazon-Beauty. MovieLens2:MoveLens-1M	User sequences are converted into dynamic graph, which contains the chronological order, time stamp of user item interactions	User sequences converted into a dynamic graph thereby using short and long term embeddings.	Captures Temporal User Behavior, efficiently handle dynamic changes	The complexity of processing large, evolving graph structures and training DGNNs.
[2] Transfer Learning With Time Series Data:A Systematic Mapping Study. (2021)	UCI HAR Dataset, MIT-BIH Arrhythmia Database, PhysioNet datasets Energy forecasting datasets	Reuse a pretrained model on new data by adjusting its weights to adapt to the target task.	**fine-tuning** was the most commonly used transfer learning approach in time series problems.	Simple to implement; leverages knowledge from the source domain;widely applicable.	May struggle with significant domain shifts;

LITERATURE SURVEY

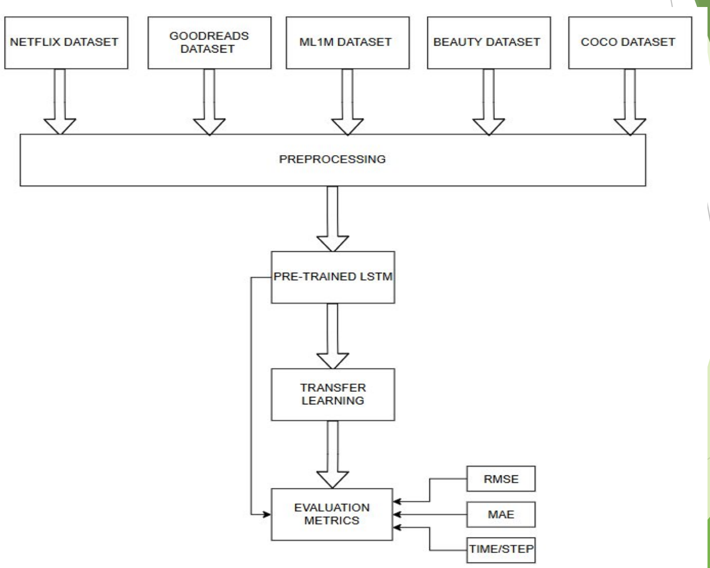
PAPER	DATASET	METHODOLGY	RESULT	ADVANTAGES	DISADVANTAGES
[4] Making recommendations using transfer learning Xing Fang.(2020)	Amazon Review dataset (Electronics, Books, Movies, Music, Home & Kitchen categories)and IMDB Movie Review dataset	Used BERT and XLNet for review representation and fine-tuned it across source and target domain	Transfer learning improved performance across target domains in terms of accuracy.	Bidirectional context understanding and addresses the issue of data sparsity	Requires high computational resources for fine-tuning large models like BERT.
[3] Novel online Recommendation algorithm for Massive Open Online Courses (NoR-MOOCs) (2021)	Coco dataset	Used hypersphere-based embedding to map users and items on a high-dimensional sphere, leveraging angular similarity for improved recommendations.	Improved recommendations with low rmse values.	able to provide personalized course recommendations to users.	-Parameter Sensitivity: as the generation size, radius of hyper-spheres. Cold start problems.

PROPOSED METHOD

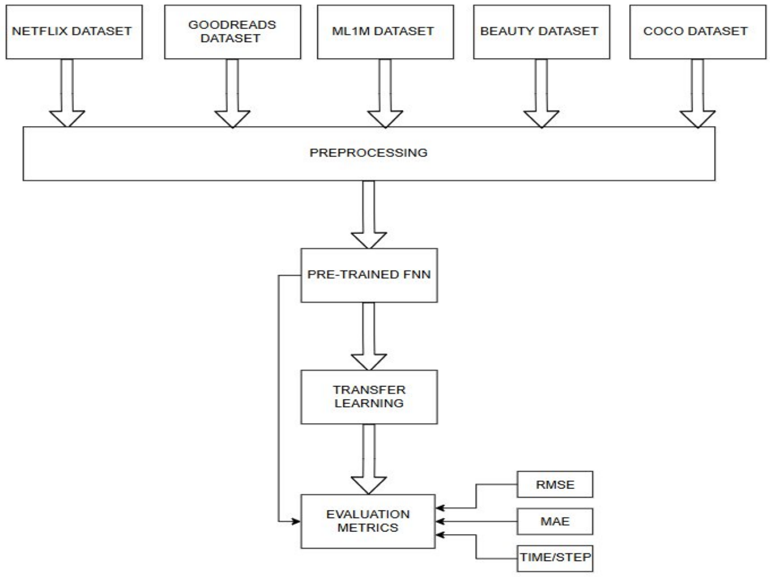
Applied **transfer learning** across four distinct deep learning models (Model 1 to Model 4).
•Trained base models on source datasets, then fine-tuned on **target datasets**.
•Evaluated model performance using **RMSE**, **MAE**, and **Time/Step** before and after transfer learning.
•Conducted **cross-dataset** and **cross-model** comparisons to identify performance patterns.
•Designed a benchmarking framework for analyzing **accuracy-efficiency trade-offs**.

SYSTEM ARCHITECTURE

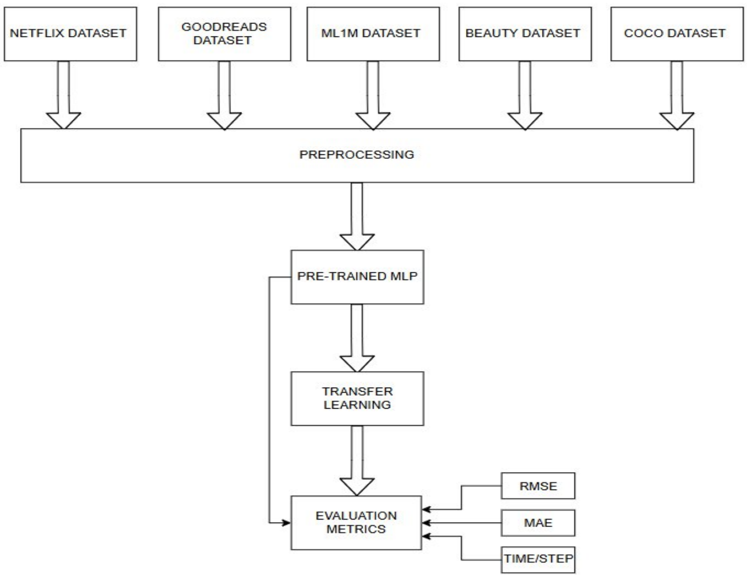
LSTM-1



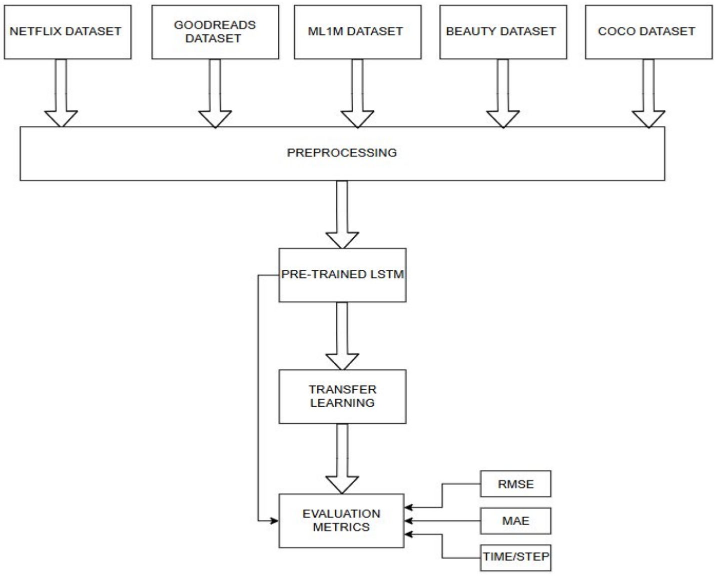
FNN


NETFLIX DATASET
GOODREADS DATASET
ML1M DATASET
BEAUTY DATASET
COCO DATASET
PREPROCESSING
PRE-TRAINED FNN
TRANSFER LEARNING
EVALUATION METRICS
RMSE
MAE
TIME/STEP

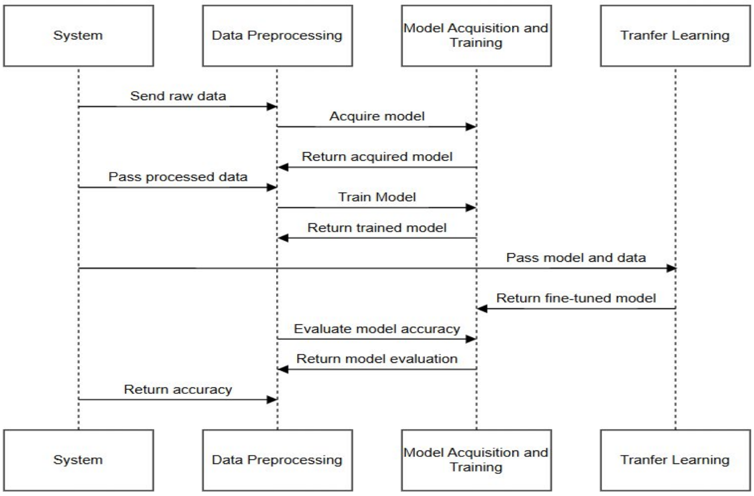
MLP


NETFLIX DATASET
GOODREADS DATASET
ML1M DATASET
BEAUTY DATASET
COCO DATASET
PREPROCESSING
PRE-TRAINED MLP
TRANSFER LEARNING
EVALUATION METRICS
RMSE
MAE
TIME/STEP

LSTM-2



SEQUENCE DIAGRAM



MODULES

MODULE-1 Data Preprocessing

Tasks:

- Data Cleaning: Handle missing values, remove duplicates, and clean noisy text (e.g., typos in reviews or course descriptions). .
- For numerical data, detect and correct errors such as impossible values or inconsistencies. Additionally, ensure that each record is unique and relevant by filtering out any irrelevant or redundant data entries.

Normalization: Scale data where necessary (e.g., rating scores).

MODULE-2 Model Acquisition

Objective: To acquire a predictive model using a multivariate models.

Feature Interactions: Handles multiple correlated variables like learner ratings, leanrer ID, and course ID.

Module-3 Model Training

Objective: Utilize a pre-trained LSTM model to generate predictions based on the project dataset, without modifying the model's existing weights or structure.

Training Strategy:

1.Using a Pre-Trained Model:

- The LSTM model was pre-trained on several datasets to capture general temporal dependencies and patterns related to similar tasks.
- The pre-trained model serves as a ready-to-use tool.

2.Dataset Integration:

- The project dataset, containing fields like learner ID, course ID, learner rating, course ratings, was formatted to align with the input requirements of the pre-trained models.
- Sequential data (e.g., learner-course interactions) was structured in time-series format to match the multivariate input expectations.
  Datasets taken:
  Netflix, Goodreads, ML1M, COCO, Beauty

3.Evaluation:

- The model predictions were evaluated using the following metrics:
  - **Root Mean Square Error (RMSE)**: Assess the magnitude of prediction errors.
  - **Mean Absolute Error (MAE)**: Measure the average prediction error.
  - **Time/step**: Measure the time taken for each epoch

Output: the model was found to produce fairly accurate predictions with the input of data with minimal RMSE.

Module-4 Transfer learning

Transfer Learning in Educational Recommender System:

- Transfer learning leverages a pre-trained model to solve a new, but related task, reducing training time and computational effort.
- In this project, several pre-trained models are used, which have already been trained on a large dataset of educational data.
- The model is fine-tuned using our specific dataset (user-course interactions), adapting it to make relevant course recommendations

How Transfer Learning is Applied:

The pre-trained LSTM model is used as the starting point, which has already learned patterns from general educational data.

Fine-tuning is performed by retraining the model on our specific dataset to tailor it for course recommendations.

The model's last layer alone is trained to specialize in making accurate predictions for the given task with our dataset.

Module-5 Comparison

•**Metric-Based Evaluation**
Compared RMSE, MAE, and Time/Step across all four models before and after applying transfer learning.
•**Visual Analysis**
Bar charts and plots used to highlight performance trends and trade-offs across multiple datasets.
•**Tabular Comparison**
Summarized quantitative differences showing how transfer learning impacted each model's efficiency and accuracy.
•**Performance Insights**
•Some models achieved significant inference speedups (up to 80%)
•Accuracy improved in certain cases; in others, slight degradation revealed domain transfer limitations
•**Cross-Model Trends**
Identified which models benefited most from transfer learning and which showed limited gains — supporting deeper architectural insights.

WORK BREAKDOWN AND RESPONSIBILITIES

Sona Sebastian
LSTM-2 and FNN:
Model training and transfer learning

Sreya S
LSTM-1 and MLP:
Model training and transfer learning

Susan Sara Joby
Data collection and preprocessing

Vaishnavi M
Model acquisition and comparison

HARDWARE AND SOFTWARE REQUIREMENTS

- Software

Python 3.x, Jupyter Notebook / VS Code / Google Colab

Libraries: TensorFlow / PyTorch, NumPy, Pandas, Scikit-learn, Matplotlib

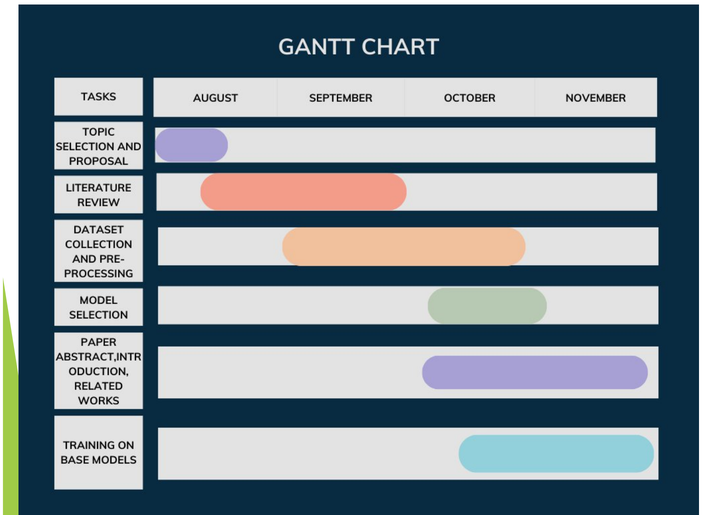
- Hardware

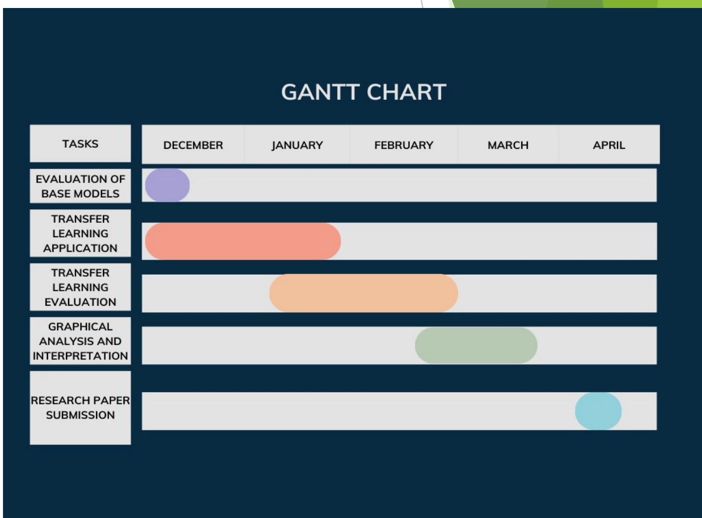
System with at least 8 GB RAM (16 GB preferred for parallel processing)

GPU (NVIDIA 4GB+ VRAM) if not using cloud; else Google Colab Free/Pro is enough

Stable internet connection (for model downloads and cloud-based execution)

GANTT CHART





RISKS AND CHALLENGES

•Performance Degradation Due to Domain Shift
If the source domain (pretraining data) is significantly different from the target domain (e.g., educational data), the model may fail to generalize well, leading to poor performance.
•Impact of Data Sparsity
Limited labeled data in the target domain can make it difficult for the model to learn task-specific patterns, reducing the effectiveness of transfer learning.
•Overfitting During Fine-Tuning
Transfer learning with small datasets can cause the model to overfit to the target data instead of generalizing.

RESULT

•Improved Inference Time
Reduced model execution time from up to **10ms** to just **2ms** per step using transfer learning.
•Improved Accuracy
On applying transfer learning, the model performed efficiently on minimal datastet
•Accuracy vs. Efficiency Trade-off Analysis
Evaluated how transfer learning affects RMSE and MAE across different models and datasets.
•Cross-Model Performance Comparison
Identified which models (1–4) benefit most from transfer learning in terms of both speed and accuracy.
•Reusable Evaluation Framework
Established a benchmarking approach using RMSE, MAE, and Time/Step for future research.
•Deployment-Ready Optimized Models
Fine-tuned models are efficient and suitable for **real-time and low-resource environments**

COMPARISON OF RESULTS BEFORE AND AFTER APPLYING TRANSFER LEARNING

DATASETS	MODEL 1(LSTM)	MODEL 2 (FNN)	MODEL 3(MLP)	MODEL 4(LSTM)
DATASET 1	Before: RMSE:0.8851 MAE:0.7301 TIME/STEP:11ms After: RMSE:0.9876 MAE:0.7754 TIME/STEP:3ms	Before: RMSE:0.97 MAE:0.77 TIME/STEP:656us After: RMSE:0.97 MAE:0.77 TIME/STEP:509us	Before: RMSE:0.96 MAE:0.79 TIME/STEP:847us After: RMSE:0.96 MAE:0.78 TIME/STEP:491us	Before: RMSE:0.84 MAE:0.65 TIME/STEP:10ms After: RMSE:0.93 MAE:0.74 TIME/STEP:2ms
DATASET 2	Before: RMSE:0.3491 MAE:0.2456 TIME/STEP:10ms After: RMSE:0.5373 MAE:0.4293 TIME/STEP:3ms	Before: RMSE:0.68 MAE:0.34 TIME/STEP:658us After: RMSE:0.96 MAE:0.70 TIME/STEP:437us	Before: RMSE:0.37 MAE:0.17 TIME/STEP:526us After: RMSE:0.41 MAE:0.25 TIME/STEP:536us	Before: RMSE:0.63 MAE:0.35 TIME/STEP:5ms After: RMSE:0.73 MAE:0.57 TIME/STEP:2ms

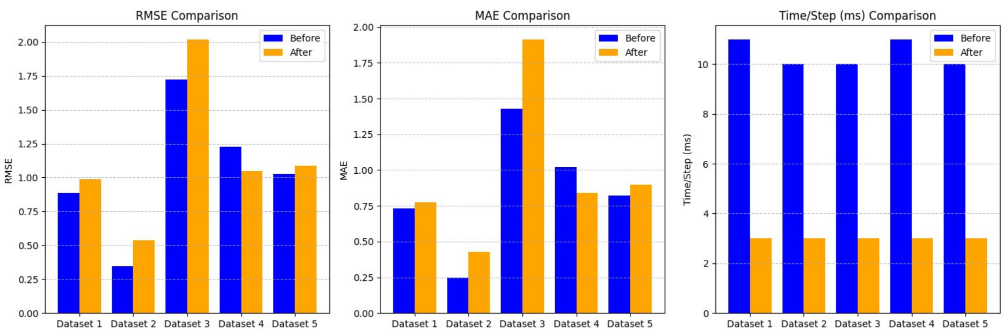
COMPARISON OF RESULTS BEFORE AND AFTER APPLYING TRANSFER LEARNING

DATASET 3	Before: RMSE:1.721 MAE:1.42966 TIME/STEP:10ms After: RMSE:2.0208 MAE:1.9150 TIME/STEP:3ms	Before: RMSE:0.89 MAE:0.76 TIME/STEP:742us After: RMSE:0.93 MAE:0.83 TIME/STEP:669us	Before: RMSE:1.68 MAE:1.48 TIME/STEP:860us After: RMSE:0.54 MAE:0.39 TIME/STEP:417us	Before: RMSE:0.97 MAE:0.86 TIME/STEP:7ms After: RMSE:0.98 MAE:0.92 TIME/STEP:2ms
DATASET 4	Before: RMSE:1.228 MAE:1.021 TIME/STEP:11ms After: RMSE:1.0461 MAE:0.8397 TIME/STEP:3ms	Before: RMSE:0.97 MAE:0.79 TIME/STEP:617us After: RMSE:1.02 MAE:0.85 TIME/STEP:556us	Before: RMSE:1.13 MAE:0.95 TIME/STEP:692us After: RMSE:1.14 MAE:0.95 TIME/STEP:506us	Before: RMSE:1.15 MAE:0.93 TIME/STEP:6ms After: RMSE:1.00 MAE:0.83 TIME/STEP:2ms

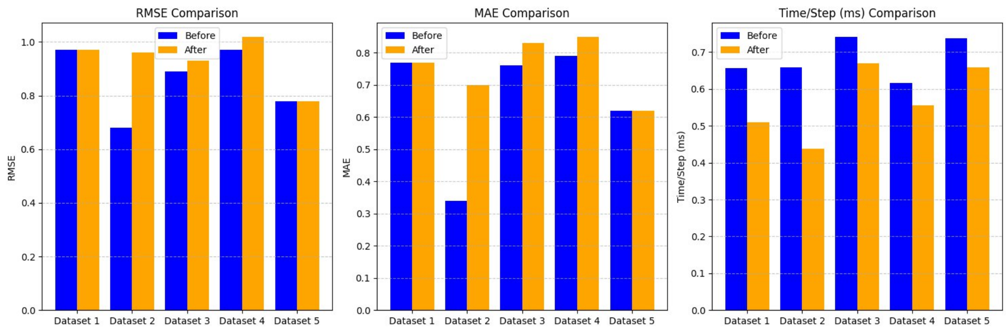
COMPARISON OF RESULTS BEFORE AND AFTER APPLYING TRANSFER LEARNING

DATASET 5	Before: RMSE:1.0269 MAE:0.822 TIME/STEP: 10ms After: RMSE:1.0865 MAE:0.8981 TIME/STEP: 3ms	Before: RMSE:0.78 MAE:0.62 TIME/STEP: 737us After: RMSE:0.78 MAE:0.62 TIME/STEP: 659us	Before: RMSE:0.94 MAE:0.71 TIME/STEP: 802us After: RMSE:1.28 MAE:1.02 TIME/STEP: 564us	Before: RMSE:0.82 MAE:0.66 TIME/STEP: 7ms After: RMSE:1.2 MAE:1.06 TIME/STEP: 2ms

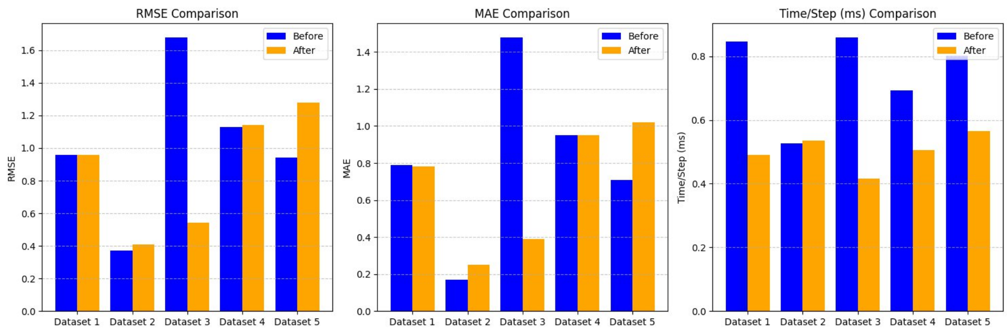
MODEL-1(LSTM)



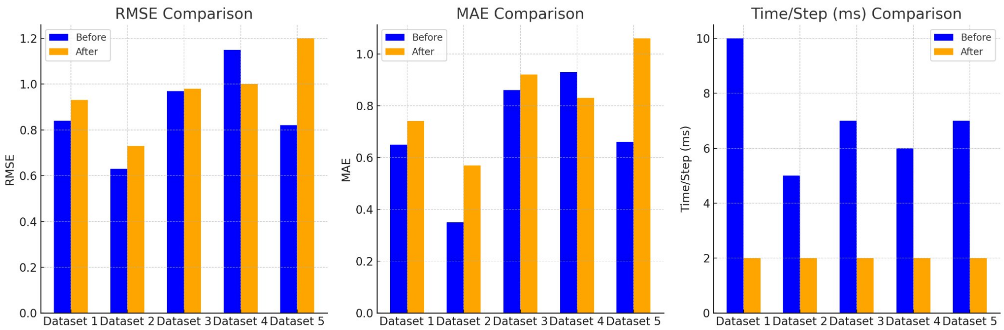
MODEL-2(FNN)



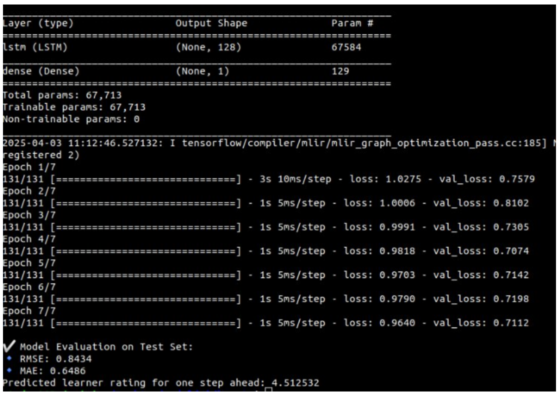
MODEL-3(MLP)



MODEL-4(LSTM)



BEFORE AND AFTER TRANSFER LEARNING(MODEL-4)

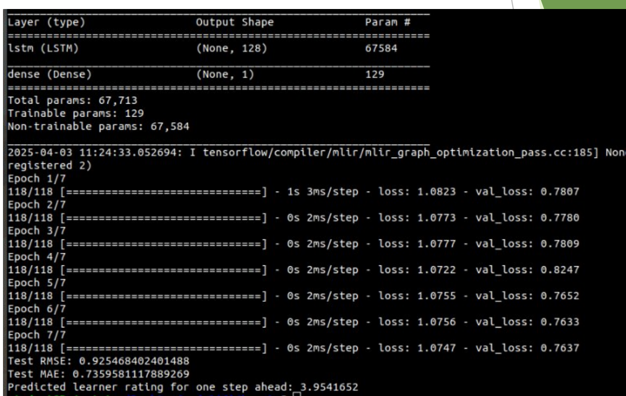


```
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 128)               67584

dense (Dense)                (None, 1)                 129
=================================================================
Total params: 67,713
Trainable params: 67,713
Non-trainable params: 0

2025-04-03 11:12:46.527132: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185]
registered 2)
Epoch 1/7
131/131 [==============================] - 3s 10ms/step - loss: 1.0275 - val_loss: 0.7579
Epoch 2/7
131/131 [==============================] - 1s 5ms/step - loss: 1.0006 - val_loss: 0.8102
Epoch 3/7
131/131 [==============================] - 1s 5ms/step - loss: 0.9991 - val_loss: 0.7305
Epoch 4/7
131/131 [==============================] - 1s 5ms/step - loss: 0.9818 - val_loss: 0.7074
Epoch 5/7
131/131 [==============================] - 1s 5ms/step - loss: 0.9703 - val_loss: 0.7142
Epoch 6/7
131/131 [==============================] - 1s 5ms/step - loss: 0.9790 - val_loss: 0.7198
Epoch 7/7
131/131 [==============================] - 1s 5ms/step - loss: 0.9640 - val_loss: 0.7112

✓ Model Evaluation on Test Set:
 • RMSE: 0.8434
 • MAE: 0.6486
Predicted learner rating for one step ahead: 4.512532
```



```
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 128)               67584

dense (Dense)                (None, 1)                 129
=================================================================
Total params: 67,713
Trainable params: 129
Non-trainable params: 67,584

2025-04-03 11:24:33.052694: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] Non
registered 2)
Epoch 1/7
118/118 [==============================] - 1s 3ms/step - loss: 1.0823 - val_loss: 0.7807
Epoch 2/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0773 - val_loss: 0.7780
Epoch 3/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0777 - val_loss: 0.7809
Epoch 4/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0722 - val_loss: 0.8247
Epoch 5/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0755 - val_loss: 0.7652
Epoch 6/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0756 - val_loss: 0.7633
Epoch 7/7
118/118 [==============================] - 0s 2ms/step - loss: 1.0747 - val_loss: 0.7637
Test RMSE: 0.925468402401488
Test MAE: 0.7359581117889269
Predicted learner rating for one step ahead: 3.9541652
```

FUTURE WORK

•**Feedback-Driven Adaptation**
Integrate user or system feedback to further fine-tune models in real-time.
•**Expand to More Domains**
Test transfer learning across additional real-world educational datasets and learning tasks.
•**Model Compression & Optimization**
Explore pruning, quantization, or knowledge distillation to further reduce inference time.
•**Hybrid Recommender Architectures**
Combine collaborative filtering with deep learning and transfer learning for robust recommendations.

CONCLUSION

Our efficient educational recommender system utilizes transfer learning to significantly enhance accuracy and reduce training/testing time in course recommendations.

By leveraging knowledge from related domains, transfer learning addresses key challenges data sparsity, enabling the system to deliver more relevant recommendations even with limited data.

By effectively integrating transfer learning, the project demonstrates the potential to boost the performance of educational recommender systems, making learning experiences more personalized and impactful.

REFERENCES

1. X. Fang, “Making recommendations using transfer learning,” Neural Comput. Appl.,vol. 33, no. 15, p. 9663–9676, Aug 2021.
2. M. Weber, M. Auch, C. Doblander, P. Mandl, and H.-A. Jacobsen, “Transfer Learning With Time Series Data: A Systematic Mapping Study,” *IEEE Access*, vol. 9, pp. 165409–165432, 2021.
3. Khalid, A., Lundqvist, K., Yates, A., & Ghzanfar, M. A. (2021). Novel online Recommendation algorithm for Massive Open Online Courses (NoR-MOOCs). *PLOS ONE*, *16*(1), e0245485.
4. M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, “Dynamic Graph Neural Networks for Sequential Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, Jan. 2022.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision
To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission
To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision
To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission
To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)
Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

After the completion of the course the student will be able to:

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course, the student will be able to:

SL.NO	DESCRIPTION	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level:Apply).	Level3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications. (Cognitive knowledge level:Apply).	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams and comprehend and execute designated tasks. (Cognitive knowledge level:Apply).	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level:Apply).	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level:Analyze).	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level:Apply).	Level 3: Apply

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO2	2	2	2		1	3	3	1	1		1	1		2	
CO3									3	2	2	1			3
CO4					2			3	2	2	3	2			3
CO5	2	3	3	1	2							1	3		
CO6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
101003/CS822U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
101003/CS822U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO5	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.1-PO6	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
101003/CS822U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
101003/CS822U.1-PO9	L	Project development using a systematic approach based on well-defined principles will result in teamwork.
101003/CS822U.1-PO10	M	Project brings technological changes in society.
101003/CS822U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development, and implementation of algorithms.

101003/CS822U.1-PO12	H	Knowledge for project development contributes engineering skills in computing and information gatherings.
101003/CS822U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing, and implementation in computer science solutions in various domains.
101003/CS822U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
101003/CS822U.2-PO3	H	Identifying, formulating, and analyzing the project results in a systematic approach.
101003/CS822U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
101003/CS822U.2-PO6	H	Systematic approach in the technical and design aspects provides valid conclusions.
101003/CS822U.2-PO7	H	Systematic approach in the technical and design aspects demonstrates the knowledge of sustainable development.
101003/CS822U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
101003/CS822U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
101003/CS822U.2-PO11	H	Systematic approach also includes effective reporting and documentation, which gives clear instructions.
101003/CS822U.2-PO12	M	Project development using a systematic approach based on well-defined principles will result in better teamwork.
101003/CS822U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.

101003/CS822U.3-PO10	H	Identification, formulation, and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
101003/CS822U.3-PO11	H	Identification, formulation, and justification in technical aspects provides the betterment of life in various domains.
101003/CS822U.3-PO12	H	Students are able to interpret, improve, and redefine technical aspects with mathematics, science, and engineering fundamentals for the solutions of complex problems.
101003/CS822U.4-PO5	H	Students are able to interpret, improve, and redefine technical aspects with identification, formulation, and analysis of complex problems.
101003/CS822U.4-PO8	H	Students are able to interpret, improve, and redefine technical aspects to meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
101003/CS822U.4-PO9	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
101003/CS822U.4-PO11	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.4-PO12	H	Students are able to interpret, improve, and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.

101003/CS822U.5-PO1	H	Students are able to interpret, improve, and redefine technical aspects, apply ethical principles, and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.5-PO2	M	Students are able to interpret, improve, and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
101003/CS822U.5-PO3	H	Students are able to interpret, improve, and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
101003/CS822U.5-PO4	H	Students are able to interpret, improve, and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
101003/CS822U.5-PO5	M	Students are able to interpret, improve, and redefine technical aspects in acquiring skills to design, analyze, and develop algorithms and implement those using high-level programming languages.
101003/CS822U.5-PO12	M	Students are able to interpret, improve, and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design, and knowledge engineering.
101003/CS822U.6-PO5	M	Students are able to interpret, improve, and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing, and providing IT solutions for different domains, which helps in the betterment of life.

101003/CS822U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS822U.6-PO9	H	Students will be able to associate with a team as an effective team player to identify, formulate, review research literature, and analyze complex engineering problems.
101003/CS822U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
101003/CS822U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data.
101003/CS822U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and committing to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
101003/CS822U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
101003/CS822U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
101003/CS822U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
101003/CS822U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.

101003/CS822U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills..