Enhancing Performance of Movie Recommendations Using LSTM With Meta Path Analysis

Article *in* IEEE Access · January 2023

DOI: 10.1109/ACCESS.2023.3327271

6 authors, including:


Zulfiqar Ali
COMSATS University Islamabad


Asif Muhammad
National University of Computer and Emerging Sciences


Ahmad Sami Al-Shamayleh
Al-Ahliyya Amman University


Kashif Naseer Qureshi
University of Limerick

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 September 2023, accepted 18 October 2023, date of publication 23 October 2023, date of current version 1 November 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3327271

RESEARCH ARTICLE

Enhancing Performance of Movie Recommendations Using LSTM With Meta Path Analysis

ZULFIQAR ALI [1], ASIF MUHAMMAD [2], AHMAD SAMI AL-SHAMAYLEH [3], KASHIF NASEER QURESHI [4], WAGDI ALRAWAGFEH [5], AND ADNAN AKHUNZADA [5], (Senior Member, IEEE)

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]FAST School of Computing, National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan
[3]Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman 19328, Jordan
[4]Department of Electronic and Computer Engineering, University of Limerick, Limerick, V94 T9PX Ireland
[5]College of Computing and IT, University of Doha for Science and Technology, Doha, Qatar

Corresponding author: Wagdi Alrawagfeh (wagdi.alrawagfeh@udst.edu.qa)

This work was supported by the Qatar National Library (QNL) of Qatar and Al-Ahliyya Amman University.

ABSTRACT Movie recommendation algorithms play an important role in assisting consumers in identifying films that match their likes. Deep Learning, particularly Long Short-Term Memory (LSTM) networks, has shown substantial promise in collecting sequential patterns to improve movie recommendations among the different techniques used for this purpose. Long Short-Term Memory-Inter Intra-metapath Aggregation (LSTM-IIMA) in movie recommendation systems is proposed in this study, with a specific focus on incorporating intra and inter-metapath analysis. The intra-metapath analysis investigates interactions within a single metapath, whereas the inter-metapath analysis investigates links between numerous metapaths. Intra and inter-metapath analyses are used in the LSTM-based movie recommendation system LSTM-IIMA to capitalise on these rich linkages. Each metapath sequence records the dependencies of a user's interactions with films and other things. The LSTM architecture has been modified to handle these metapath sequences, processing them to record temporal dependencies and entity interactions. To optimize the parameters and minimize prediction errors, the model is trained using supervised learning techniques. To measure the quality and usefulness of the recommendations, the LSTM-IIMA evaluation incorporates metrics such as precision, recall, ablation analysis, time efficiency and Area Under the Curve (AUC). The performance of the system is compared to that of alternative recommendation techniques HAN and MAGNN. Overall, incorporating intra and inter-metapath analysis into the LSTM-IIMA improves its ability to capture complex linkages and dependencies between movies, users, and other things.

INDEX TERMS Long short-term memory, inter-metapath, intra-metapath, metapath analysis, metapath instances, ablation analysis, machine learning, deep learning.

I. INTRODUCTION

Graph Neural Networks (GNNs) have attracted a great deal of attention in recent years due to their ability to model and analyse complicated relational data represented as graphs. With the rise of heterogeneous networks, where nodes and edges might be of many sorts, good aggregation methods that can capture the rich structural information become critical. Aggregation methods are critical in GNNs because they let nodes exchange and aggregate information with neighbouring nodes, allowing for effective representation learning [1].

One of the most difficult aspects of gathering information from heterogeneous graphs is dealing with the diversity of meta-paths. Meta-paths are node-type sequences that capture

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

the semantics and structural patterns of the graph. They offer a strong method for examining and utilising the graph's huge variance. Aggregating information over several meta-pathways, on the other hand, is difficult due to the paths' varying lengths, node kinds, and structural characteristics.

In this paper, we offer a novel approach for inter and intra-metapath aggregation in heterogeneous graph neural networks using Long Short-Term Memory (LSTM) [2]. Our method attempts to successfully aggregate information from many meta-paths while accounting for dependencies within each meta-path. The expressive power of LSTMs is used to capture sequential and long-range dependencies between nodes, allowing us to grasp the complex structural patterns seen in heterogeneous graphs.

The purpose of this research is to look into the importance, methodology, and uses of inter- and intra-metapath aggregation techniques in graph analysis [2]. Further, we investigate their benefits, drawbacks, and trade-offs, considering factors like computational efficiency, interpretability, and generalization performance. This work can help researchers and practitioners learn about the most effective ways to integrate metapaths into GNNs. It underlines the importance of scalable algorithms that can handle large-scale graphs effectively while maintaining interpretability and explainability. The study also emphasises the importance of creating adaptive and dynamic metapath-based strategies for capturing temporal and evolving graph structures [3].

The paper focuses on the concepts of inter-metapath and intra-metapath aggregation in GNNs. Inter-metapath aggregation is the integration of many metapaths to capture complex structural linkages in a network. It allows GNNs to leverage many paths and capture higher-order connection patterns, allowing for more expressive graph representations. In contrast, intra-metapath aggregation focuses on information propagation and aggregation along a single metapath, allowing GNNs to capture finer-grained structural relationships. Inter-metapath Aggregation is the collection of many metapaths in a graph to capture more complicated interactions. Intra-Metapath aggregate, on the other hand, concentrates on information propagation and aggregate along a single metapath. GNNs may detect fine-grained structural links within a given path as well [4]. In a variety of applications, the employment of inter-metapath and intra-metapath aggregation techniques in GNNs has given promising results. In social networks, inter-metapath aggregation can uncover significant users and communities, and intra-metapath aggregation can detect sequential trends. In recommendation systems, inter-metapath aggregation can capture various user-item relationships, whereas intra-metapath aggregation can characterise users' sequential preferences. These examples demonstrate how inter-metapath and intra-metapath aggregation tactics can aid in the learning and analysis of graph representations [5].

To grasp the significance of inter-metapath and intra-metapath aggregation, it is necessary to first explore the fundamental components of GNNs. GNNs are primarily based on iterative message forwarding, in which each node absorbs information from neighbouring nodes and modifies its representation. This technique enables nodes to incorporate data from their immediate surroundings, resulting in more informative and context-aware representations. GNN success, on the other hand, is heavily dependent on how well they mirror the underlying graph topologies.

Our major contributions can be summarized as follows:

1. We present a novel method for aggregating information across many meta-paths. LSTMs are used in our strategy to capture the sequential dependencies between nodes in distinct meta-paths. By combining information from numerous meta-paths, we can depict the intricate relationships and interactions between different types of nodes in the heterogeneous graph.
2. We perform an ablation analysis based on components involved in the generated graph. Using Precision-Recall and AUC, the accuracy factor is defined for all possible combinations inside each meta-path. Our method efficiently captures structural patterns and relationships between nodes by considering the sequential nature of the nodes inside a meta-path. This allows us to better model and comprehend the graph's complicated relationships.
3. We run comprehensive experiments on a variety of real-world heterogeneous network datasets to assess the efficacy of our suggested technique. Our experimental results reveal that our technique outperforms state-of-the-art baselines in several assessment parameters, including node classification accuracy and connection prediction ability. This demonstrates the effectiveness of our technique in capturing and using the structural information contained in heterogeneous graphs.

The remainder of the manuscript is structured as follows. Section II goes over the specifics of our proposed approach for inter and intra-metapath aggregation using LSTM. Section III provides details and the importance of the LSTM approach for recommendation systems. Experimental setup, including datasets, assessment metrics, result evaluation and baselines, are described in Section IV. Section V concludes the paper by outlining potential future research directions.

II. PROPOSED FRAMEWORK OF INTER-METAPATH AND INTRA-METAPATH LSTM-BASED RECOMMENDATION SYSTEM

Recommendation systems are critical in assisting users in locating relevant and personalised products from a plethora of possibilities. Traditional recommendation systems frequently confront difficulties in capturing complicated user-item interactions and combining data from several sources [6]. We present a novel architecture for developing a recommendation system that incorporates inter-metapath and intra-metapath LSTM-based aggregation in this research. Our approach attempts to increase recommendation performance by efficiently capturing dependencies within and across distinct meta-paths by harnessing the capabilities of GNNs and LSTM networks. We enable the recommendation system to capture both the diversity of user-item interactions and the sequential dependencies within each interaction by combining the strengths of meta-path-based modelling with LSTM-based aggregation [7]. The suggested framework is made up of numerous critical components, which are discussed in detail below:

A. FRAMEWORK OVERVIEW

Recommendation systems are critical in assisting users in locating relevant and personalised products from a plethora of possibilities. Traditional recommendation systems frequently confront difficulties in capturing complicated user-item interactions and combining data from several sources [8]. We present a novel architecture for developing a recommendation system that incorporates inter-metapath and intra-metapath LSTM-based aggregation in this research. Our approach attempts to increase recommendation performance by efficiently capturing dependencies within and across distinct meta-paths by harnessing the capabilities of GNNs and LSTM networks. The suggested framework is made up of numerous critical components, which are discussed below:

Data Representation: User-Item Graph Construction: We build a heterogeneous graph representation of the recommendation data, with nodes representing users and items and edges representing interactions between them. The graph records several forms of interactions, such as user-item sales, ratings, or views, allowing for a complete depiction of the recommendation data [9], [10].

Extraction of Meta-Paths: We extract meta-paths from the user-item graph to capture the semantics and structural patterns in the data. Meta-paths are node-type sequences, such as "User-Item-User" or "Item-User-Item," that give a powerful approach to exploring and using the graph's deep relational structure [11].

Inter-metapath LSTM Aggregation: For each meta-path, meta-path embedding is performed by aggregating node and edge features along the meta-paths [12]. This captures semantic information as well as the features of various forms of user-item interactions.

Inter-metapath Aggregation Using LSTM Networks: We use LSTM networks to capture the sequential dependencies between nodes in various meta-paths. We aggregate information from various meta-paths and capture the intricate linkages and interactions between different types of interactions by considering the temporal order of meta-paths. This allows us to predict user-item interactions and capture the variety of recommendation patterns [13] more accurately.

We learn low-dimensional embeddings for people and items by merging aggregated information from both inter and intra-metapath LSTM aggregations. These embeddings record user-item interactions and encode the recommendation data's learnt representations. To produce personalised recommendations based on the learned embeddings, we use a recommendation technique such as collaborative filtering or matrix factorization. We can deliver more accurate and diverse suggestions to consumers by exploiting the enhanced representations learned through inter and intra-metapath LSTM aggregations.

Our proposed model is in a step-by-step manner, focusing on the Input, Encoder, and Decoder. We formally introduce three components that comprise the LSTM-IIMA model: (i) On the user side, adjacent sampling is based on numerous meta-paths, such as UMU, UGU, and UAU, and the movie side, MDM, MUM, MAM, and MCM. (ii) The information aggregation of the sampled neighbours, including Vectors and Edges. (iii) Objective optimization and model training using MLP. Figure 1 illustrates the process of finding neighbour sets for each node through meta-paths and creating random walks for triplet sampling. The data is collected at the node level to aggregate properties from diverse sources, and at the neighbour-type level to perform type-based aggregation. The model assigns weights to type-based embeddings using a multi-attention technique. A novel semi-supervised LSTM-IIMA for Heterogeneous Graphs, follows a hierarchical attention structure, starting with node-level attention and then proceeding to semantic-level attention. Initially, node-level attention is introduced to learn the weights of meta-path-based neighbours and aggregate them to obtain the embedding specific to semantic information. Subsequently, LSTM-IIMA employs semantic-level attention to distinguish between meta-paths and obtain an optimally weighted combination of semantic-specific node embeddings for the given task. The initial phase of the framework in this section includes extracting two unique types of data, namely movie and user data. The movie data is then divided into Director, Genres, Country, Reviews and Actor information. However, user data contains gender and age group information. The combination of all these datasets yields a multidimensional graph depicting the relationship between users and the films they've seen, as well as their separate subcategorized datasets. This sophisticated multidimensional data serves as the framework's input and output layer. The likelihood of a user interacting with the item.

We provide formal definitions for several essential terminologies related to enhanced heterogeneous graphs in this section. In Figure 1, for example, the Movie is a metapath-based neighbour of several users since multiple users can watch the same movie at the same time. In another example, a metapath UMRMU defines that multiple users can give reviews regarding multiple movies.

B. METHODOLOGY

In this section, we describe a new metapath aggregated LSTM-based technique for heterogeneous graph embedding. Node content transformation, intra-metapath aggregation, and inter-metapath aggregation are the three key components.

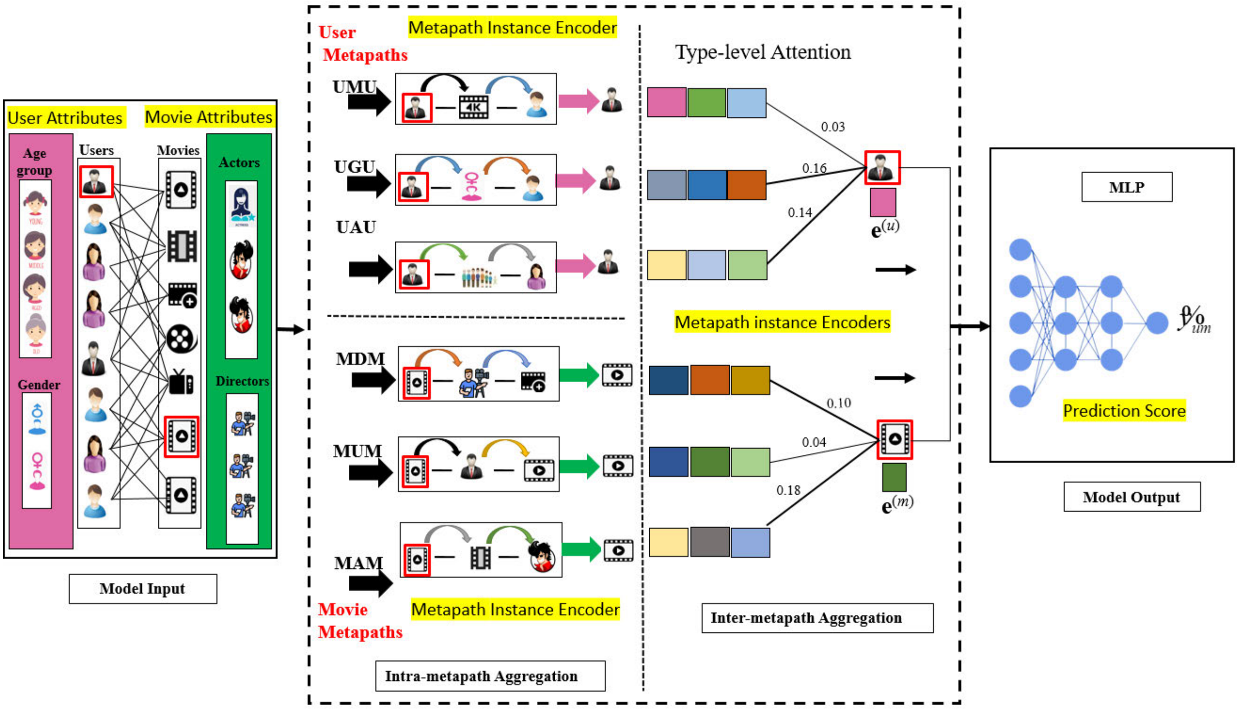


FIGURE 1. **The framework of LSTM-IIMA for movie recommendation system.**

1) NODE-LEVEL METADATA

Before collecting information from each node's meta-path neighbours, keep in mind that each node's meta-path neighbours serve a distinct purpose and have variable importance in learning node embedding for the given job. We provide node-level attention, which can learn the significance of meta-path-based neighbours for each node in a heterogeneous graph and aggregate the representation of these meaningful neighbours to construct a node embedding [14]. Because of node heterogeneity, different types of nodes have different feature spaces. As a result, we generate the type-specific transformation matrix $M_{\emptyset i}$ or each type of node (for example, a node with type i) is used to project the features of different types of nodes into the same feature space. Instead of edge type, the type-specific transformation matrix is based on node type.

The projection procedure is as follows [15]:

$$h_i' = M_{\emptyset i}.h_i$$

where $M_{\emptyset i}$ is the type-specific transformation matrix, h_i and h_i' are the node $i's$ original and projected features, respectively. The node-level attention may handle any type of node via a type-specific projection operation.

The node content transformation addresses the problem of graph heterogeneity produced by node content attributes. Following this procedure, all nodes' projected features will have the same dimension, making the aggregation of the next model component easier.

2) INTRA-METAPATH AGGREGATION

The intra-metapath aggregation layer learns the structural and semantic information encoded in the destination node, the metapath-based neighbours, and the context in between, given a metapath P, by storing the metapath instances of P [16].

$$h_{p(v,u)} = f_{\theta}(P\,(v,\,u))$$

P(v,u) is a metapath instance that connects the target node v with the metapath-based neighbour $u \epsilon N_v^p$; the intermediate nodes of *P(v,u)* are defined as $mP(v,u) = P(v,u) \setminus [u,v]$. Intra-metapath aggregation combines all of the node features along a metapath instance into a single vector using a specific metapath instance encoder.

where $h_{p(v,u)} \epsilon R_d$ has a dimension of d. *P(v,u)* specifies a single instance, though several instances may connect two nodes.

After encoding the metapath instances into vector representations, we employ a graph attention layer to weighted sum the metapath instances of P connected to target node v. The fundamental notion is that distinct metapath instances, to variable degrees, contribute to the representation of the destination node [17].

$$h_v^p = \sigma(\sum_{u \epsilon N_v^p} \alpha_{vu}^p.h_{p(v,u)})$$

We may simulate this by learning a normalised importance weight α_{vu}^p for each metapath instance and then weighting the sum of all instances.

3) INTER-METAPATH AGGREGATION

After aggregating the node and edge data within each metapath, we must use an inter-metapath aggregation layer to integrate the semantic information disclosed by all metapaths. Now for a node type A, we have $|V_A|$ sets of latent vectors: h_v^{p1}, h_v^{p2},..., h_v^{pm}. for $v\in V_A$, where M is the number of metapaths for type A. Taking the element-wise mean of these node vectors is a simple inter-metapath aggregation strategy. This strategy is extended by using the attention mechanism to assign various weights to distinct metapaths. Because metapaths are not equally important in a heterogeneous graph, this operation makes sense [18].

Averaging the altered metapath-specific node vectors for all nodes $v\in V_A$, summarize each metapath $P_i \in P_A$. As a result, we employ the attention technique to fuse the metapath-specific node vectors of v in the following manner: [17]:

$$e.P_i = Q_A^T.SP_i$$

$$h_v^{P_A} = \left(\sum_{P\epsilon P_A} \beta_P.h_v^P\right)$$

$Q_A \in R_m^d$ depicts the parameterized attention vector for node type A. βP_i represents the relative relevance of metapath P_i to the nodes of type A. LSTM-IIMA employs an additional linear transformation to convert the node embedding to vector space with the required output dimension.

$$h_v = \sigma(W_o.h_v^{P_A})$$

where $\sigma(\cdot)$ is a function of activation, and $W_o \in R_{\mathrm{o}}^{\mathrm{d*d}}$ is a weight matrix.

III. LSTM APPROACH

LSTM is a type of Recurrent Neural Network (RNN) architecture that is designed to address the vanishing gradient problem, which can occur when training traditional RNNs. LSTMs were introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997 and have become popular for various tasks involving sequential data, such as Natural Language Processing (NLP), speech recognition, and time series analysis [19], [20].

At the core of an LSTM are memory cells, which can maintain information over long sequences, allowing them to capture dependencies and relationships in the input data. Each memory cell consists of three main components:

Cell State (C_t): This represents the long-term memory of the LSTM. It carries information from the earlier time steps in the sequence.

Input Gate (i): This determines how much information from the current time step should be stored in the $\mathbf{C_t}$.

Forget Gate (f): This controls how much information from the previous cell state should be forgotten or discarded.

Output Gate (o): This determines how much of the $\mathbf{C_t}$ should be exposed as the output.

The LSTM network processes input sequences step by step, updating the cell state and producing an output at each time step. The key idea behind LSTM is the use of these gates to control the flow of information. The gates are implemented as sigmoid neural network layers, which generate values between 0 and 1, indicating how much information to let through.

During the forward pass of the LSTM, the gates are computed based on the current input, previous output, and previous $\boldsymbol{C_t}$. The equations governing the computation of the gates are as follows [2], [21], [22]:

Input Gate (i):

$$i[t] = sigmoid(W_{ih}\,[t] + W_{ih}\,[h\,[t-1]] + b_i$$

Forget Gate (f):

$$f\,[t] = sigmoid(W_{fx}\,[t] + W_{fh}\,[h\,[t-1]] + b_f$$

Output Gate (o):

$$o[t] = sigmoid(W_{ox}\,[t] + W_{oh}\,[h\,[t-1]] + b_o$$

The input, forget, and output gates are used to compute the updated $\boldsymbol{C_t}$ and output at time step t:

Cell state update:

$$C\,[t] = f\,[t] * C\,[t-1] + i\,[t] * tanh(W_{cx}\,[t] + W_{ch}\,[h\,[t-1]] + b_c$$

Output:

$$h\,[t] = o\,[t] * \tanh(C\,[t])$$

In the above equations, $x[t]$ represents the input at time step t, $h[t\text{-}1]$ represents the output of the previous time step, and the W and b terms represent the weight matrices and bias vectors of the LSTM. LSTMs can learn and recall long-term dependencies in sequential data by selectively updating and forgetting information through the gates. This makes them especially suitable for activities with context and temporal requirements. Back-Propagation Through Time (BPTT) is a back-propagation enhancement that considers the sequential nature of the data when training LSTMs. Figure 2 below, shows the overall flow of Inter-metapath and Intra-metapath propagation and aggregation.

Overall, LSTMs have been demonstrated to be effective in modelling and processing sequential data, and they are frequently employed in a wide range of applications where understanding and capturing long-term dependencies are crucial. The LSTM design is depicted in mathematical terms in Figure 3.

A. LSTM FOR MOVIE RECOMMENDATION

LSTMs can be used to recommend films by modelling users' sequential patterns and preferences based on their previous encounters with films. Here's a high-level explanation of how LSTM can be used to recommend films [2], [22]:

When we use the LSTM technique to propose films, we go through several processes. These are data preparation, user

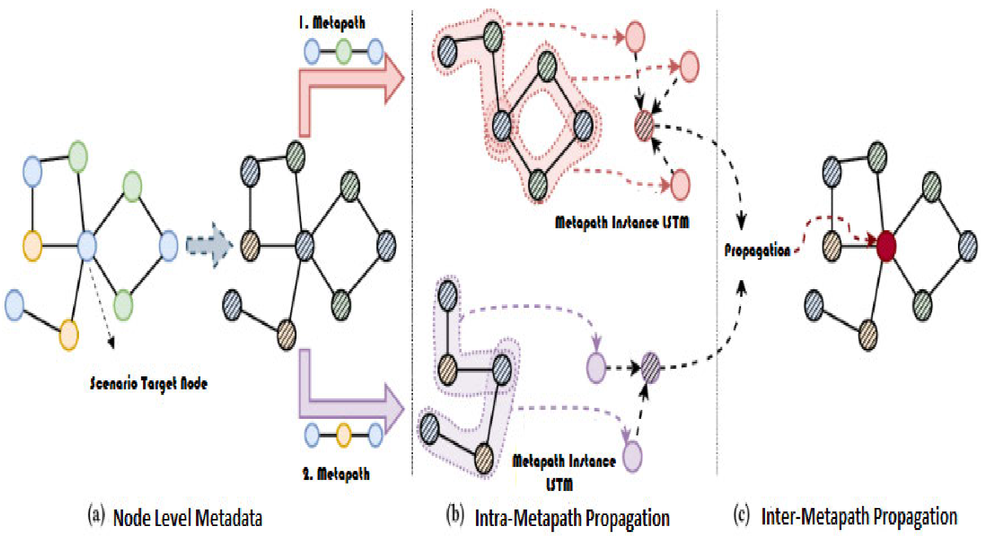


FIGURE 2. Inter-Metapath and Intra-Metapath propagation and aggregation.

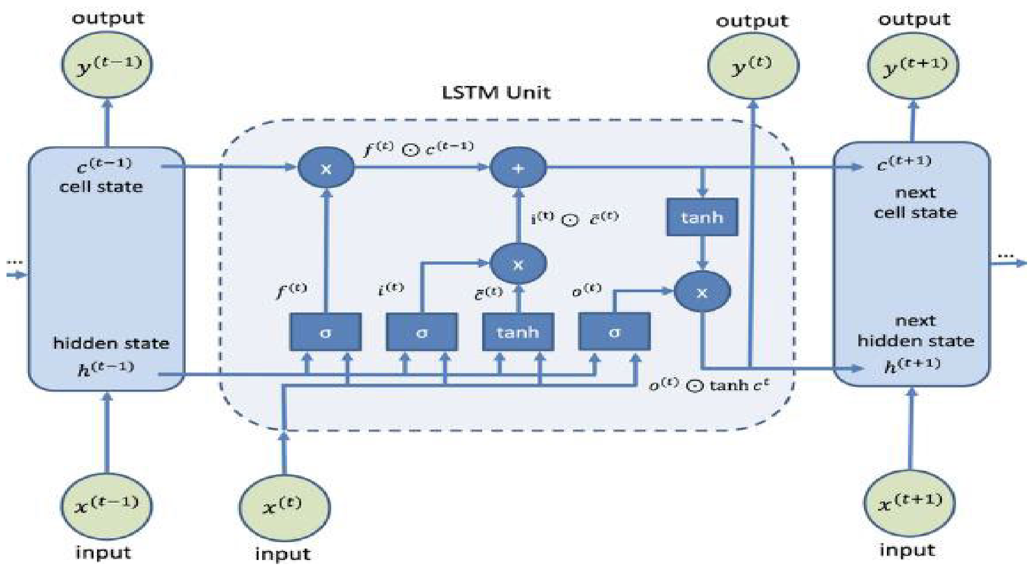


FIGURE 3. LSTM architecture.

sequence encoding, LSTM architecture, data training, evaluation, and prediction. The following are the specifics of each of these components:

1) DATA PREPARATION

The first step is to gather and preprocess the movie recommendation dataset. User ratings, movie information, and user-item interactions are common examples. The dataset is organised sequentially, with each user's interaction history recorded as a list of films rated or watched.

2) ENCODING USER SEQUENCES

The LSTM model is fed user sequences. A sequence of video embeddings or features represents each user's sequence. Techniques such as matrix factorization or deep learning-based approaches such as word embeddings or collaborative filtering methods can be used to learn movie embeddings.

3) LSTM ARCHITECTURE

The LSTM architecture is designed to process sequential data. The input to the LSTM is the sequence of movie embeddings for each user. The LSTM layer(s) process the sequence and capture the sequential patterns and dependencies in the user's movie preferences. The output of the LSTM is a fixed-size representation of the user's sequence. Figure 4 presents a detailed flow of the LSTM model for Movie Recommendation.

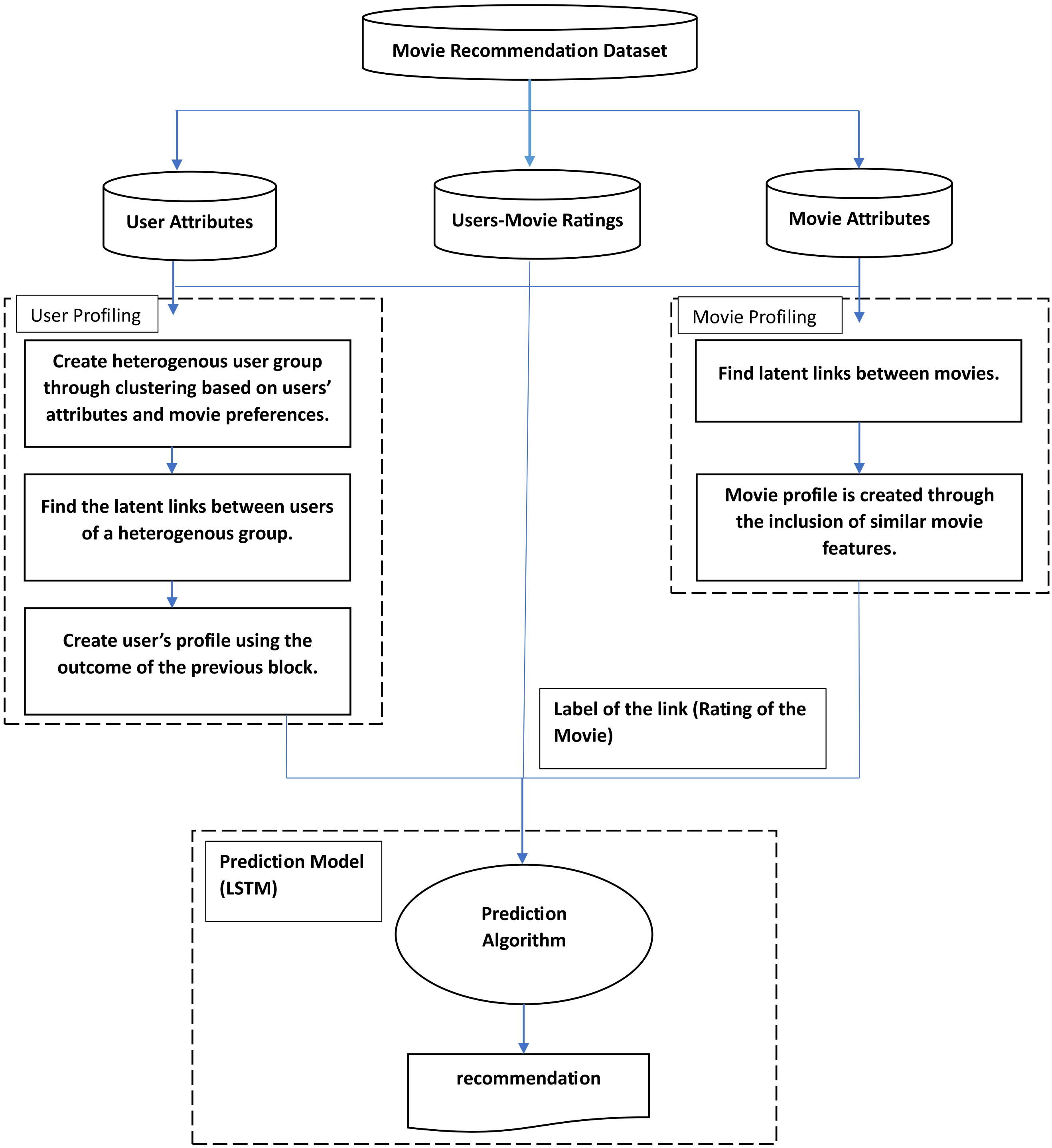


FIGURE 4. LSTM architecture for movie recommendation.

4) MOVIE RECOMMENDATION

For movie recommendation, the fixed-size representation from the LSTM layer is transferred through fully connected layers and a final output layer. A SoftMax layer that predicts the probability of user preference for each movie in the dataset can be used as the output layer.

5) TRAINING

Supervised learning techniques are used to train the LSTM model. The training data consists of user sequences and target labels, which are often binary indicators of whether the user interacted positively (e.g., rated highly) or negatively (e.g., did not rate or rated poorly) with a movie. Loss functions such as binary cross-entropy or ranking-based loss functions are used to optimise the model.

6) EVALUATION AND PREDICTION

Following training, the model can be evaluated using measures like accuracy, recall, or mean average precision to determine the quality of movie suggestions. Given a user's current sequence, the LSTM model may create recommendations by scoring and ranking movies based on expected user preferences.

It's important to note that LSTM-based techniques are only one option to create movie recommendation systems. To improve recommendation performance, other techniques such as matrix factorization, content-based filtering, and hybrid models can be employed in conjunction with LSTM. Incorporating extra features such as movie genres, actors, or user demographic data can also improve the recommendation system's capabilities.

B. AUC ANALYSIS FOR MOVIE RECOMMENDATION

In the context of movie recommendation, AUC prediction can be used to estimate the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) for a recommender system. The AUC-ROC statistic assesses the system's ability to rank positive items (movies that a user likes) higher than negative items (movies that a user dislikes) [23], [24], [25].

Typically, you would conduct the following to create an AUC prediction for movie recommendation:

Preparation of Data: Gather historical data on user-movie interactions, such as users, movies, gender, genre, age, ratings, reviews, and watch history. This data should include which films consumers liked or hated.

Feature Engineering: Extract relevant features from data, such as movie genres, actors, directors, or user demographics. These characteristics will be used to train the recommender system. Validation and training: Divide the data into two categories: training and validation. Create a recommender model using collaborative filtering, content-based filtering, or hybrid techniques based on the training data. Using assessment criteria such as AUC-ROC, evaluate the model's performance on the validation set.

AUC Prediction: After training and validating the model, the AUC-ROC that the model is expected to achieve on unknown data or in a production situation can be predicted. This prediction can be used to assess the efficacy of the model and compare it to other models or benchmarks.

Performance Assessment: Run the model in the real environment and monitor its performance over time. Compare the model's actual AUC-ROC with the expected number to determine the prediction's accuracy. This can also help discover possible modifications or areas for improvement.

It is crucial to remember that AUC prediction does not always accurately forecast how well a recommender system will perform. The similarity between the training/validation data and the real-world data determines the prediction's accuracy. Furthermore, user preferences and movie availability may shift over time, reducing the model's efficacy. To ensure its effectiveness, the recommendation system must be tested and adjusted regularly. Once trained, the model selects the previously indicated items K for each user and movie. The recommendation set is calculated using the Recall(K) function, which is detailed below [26]:

$$AUC(u) = \frac{\Sigma_{i\epsilon\omega(u)}\Sigma_{j\epsilon I\backslash\omega(u)}D(y_{ui} \geq y_{uj})}{|\omega(u)|\,|I\backslash\omega(u)\,||}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall.K(u) = \frac{|R^{1:k}(u) \cap \omega(u)\,|}{|\omega(u)\,|}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Here $D(\cdot)$ is defined as an instruction function, $\omega(u)$ is defined as the list of item sets and $R1$: $K(u)$ indicates the list of top K recommended items.

IV. EXPERIMENTAL RESULTS

The LSTM technique is described in this section for enhancing inter and intra-metapath efficacy in heterogeneous graph embedding. The trials are designed to address the following research objectives:

Research Question 1: How does LSTM-IIMA perform compared to HAN and MAGNN-based recommendation models?

Research Question 2: How do hyper-parameters affect LSTM-IIMA performance?

Research Question 3: What impact do the various components have on LSTM-IIMA?

Research Question 4: What is the LSTM-IIMA model training time efficiency?

The step-by-step flow of LSTM based Inter and Intra-metapath for the Movie Recommendation algorithm is as follows:

Several parameters are used in the LSTM-IIMA algorithm whose details are as follows: In $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{G}$ is the graph having vertices $\mathcal{V}$ and edges $\mathcal{E}$. The h_i denotes the attributes of nodes or meta-path, $\{\Phi_0, \Phi_1, \ldots, \Phi_P\}$ is the meta-path set, the number of attention head is denoted by K, S is the symbol used for final embedding. Further, we assign different weights like weight for node level attention is a and overall attention weight is β.

A. DATASET DESCRIPTIONS

We began by expanding the movie data gathering for the suggested framework experiment from one of the available movie coding websites, IMDB. The dataset has a sample size of 5500 movies, 5902 actors, 2273 directors, and 6000 users, as shown in Table 1. As previously indicated, we build six groups for grouping and clustering movies based on genre, one framework, two parallel sides of the movie, and a user. For the user movie-watching analysis, we employ "Netflix

Algorithm 1 Enhanced LSTM-IIMA

Algorithm: Enhanced LSTM-IIMA.

Input: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
Attributes of Node: $\{h_i, \forall_i \in V\}$,
Elements of the set (Meta-path),
The number of attention head K

Output: The final embedding S,
Node level-attention weight: a.
Attention weight: β.
Ablation analysis for LSTM-IIMA.
Time Cost analysis of LSTM-IIMA.

1 **for** $\Phi_i \epsilon \left\{\Phi_0, \Phi_1, \ldots\ldots\ldots \Phi_P\right\}$ *do*
2 **for** $K = 1 \ldots K$ *do*
3 Type-specific transformatione_{u-i}, e_{v-i};
4 **For** $i \epsilon V$ *do*
5 Find the metapath-based neighbors N_i^{Φ};
6 **for** $j \epsilon N_i^{\Phi}$ *do*
7 Find the weight coefficient h_{u-v};
8 **End**
Calculate the hidden semantics of the node
9 $X_{u-v} \leftarrow \sigma\left(\sum_{j\in} N_i^{\Phi} h_{u-v}, e_{v-i}\right)$;
10 **End**
11 Concatenate the learned embedding from all attention head.
12 **End**
Calculate the weight of meta-path $\beta\Phi_i$;
Calculate Intra and Inter-metapath attributes accordingly.
h_v^P; $h_v^{P_A}$
13 Fuse the semantic-specific embedding

$$S \leftarrow \sum_{i=1}^{P} G_{u-v}, X_{u-v};$$

14 **End**
15 Calculate the Cross-Entropy L;
16 Update parameters in LSTM-IIMA.
17 **return** S, a, β.

TABLE 1. Datasets used for LSTM-IIMA (E_S: embedding size, Z_1: Co-contrastive learning weight).

Datasets	Hyper-parameter settings	Nodes	Edges	Metapaths
IMDB	E_s = 128 Z_1= 1e-5	# Movie (M): 2,500 # Director (D): 2,273 # Actor (A): 5,902	# movie (M): 8,278 # director (D): 6,081 # actor (A): 5,257	MDM MAM DMD
Netflix	E_s = 256 Z_1= 1e-3	# Users (U): 1,892 # Gender (G): 17,632	# U-A-U: 12,717 # U-G-U: 92,834	UAU UGU
TV Show	E_s= 128 Z_1= 1e-6	# Users (U): 4,108 # Movie(M): 30,00	#U-M-U: 12,717 # M-U-M: 92,834	UMU MUM

Movie'' and "TV Show'' tags [27]. Data sets are publicly available on Kaggle; these enormous data sets contain information from over 8,000 films and television shows, as well as trends from over 200 million people. The embedding size (E_s), the L2 coefficient, and the learning rate are searched in {8, 16, 32, 64, 128, 256}, {0, 1e-6, 1e-5, …, 1e-3}.

Figure 5 examines the performance of AUC in three recommendation situations (datasets) based on embedding size (E_s).

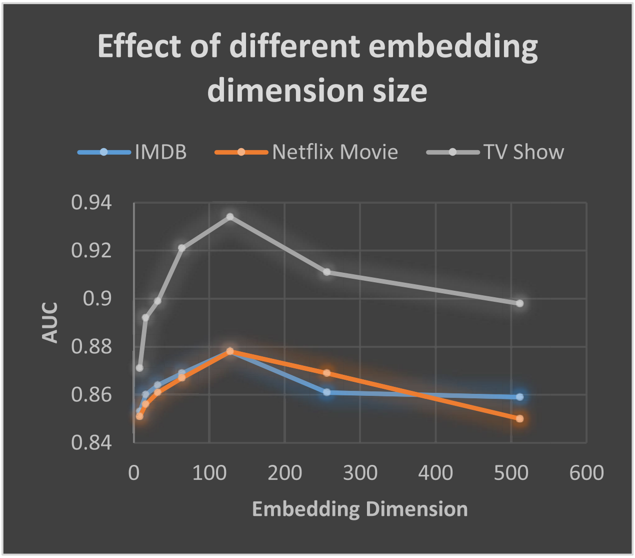


FIGURE 5. Effect of different embedding dimension sizes (E_s).

Experiments reveal that the embedding dimension should be set to 128 and 256 to maximize recommendation performance. With the increase in embedding dimension, the model performance demonstrates a continuous trend of growing and then dropping. This suggests that increasing embedding space (size) increases the embedding encoding ability to some level and allows for more extensive information. However, a large embedding size can degrade model performance.

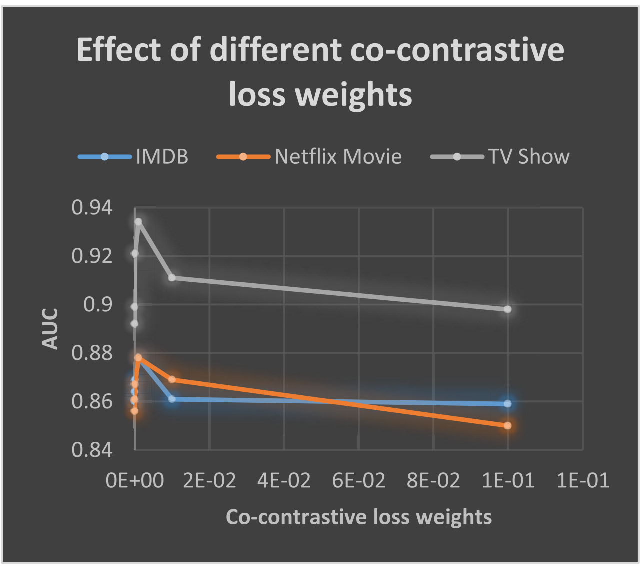


FIGURE 6. Effect of different co-contrastive loss weights (Z_1).

Figure 6 presents the effect of different co-contrastive loss weights (Z_1) embedding dimension sizes in terms of AUC. The importance of co-contrastive loss in the multi-task training process is determined by hyperparameter Z_1. Figure 6 depicts the performance trend to investigate the effect of co-contrastive loss weight 1 on model AUC performance. The following are our conclusions: The model performs best when 1 is taken to be uniformly 1e-3. To investigate the cause, when the co-contrastive loss weight is set too high, it misleads the direction of the recommendation model optimization and results in more pronounced side effects. When the value of 1 is set too low, co-contrastive learning fails to produce meaningful optimization results and even impedes correct model representation. As a result, selecting the appropriate 1 value can increase the node representation learning ability and hence the recommendation performance.

B. RESULT EVALUATION

Before providing the experimental results, this section describes the model's parameter values. The results of the experiment, which included a variety of activities, are then presented and discussed.

The F1-measure, often known as the F1 score, is a metric used to assess a model's accuracy on a given dataset. It is used to assess binary classification systems, which categorize examples as 'positive' or 'negative'. The F1-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. The F1-score is widely used to evaluate information retrieval systems such as search engines, as well as many types of machine learning models, particularly in natural language processing. The traditional F1-score formula is the harmonic mean of precision and recall. The F-score of a perfect model is 1. Precision is the proportion of positive forecasts that are correct. Recall is the proportion of true positives that are projected to be true positives.

$$\boldsymbol{F1-Score} = 2 * (Precision * recall)/(Precision + recall)$$

A high F1-score means that the model is both accurate and precise. A low F1-score means that the model is either inaccurate or imprecise. Because it considers both precision and recall, the F1-score is a valuable indicator for evaluating classification models. This is significant because a model can be extremely accurate but not precise, or highly precise but not highly accurate. The F1-score assists in balancing these two aspects and provides a more accurate measurement of the model's overall performance.

Ablation analysis is a research method in machine learning that includes eliminating or deactivating specific components of a model to identify their contribution to the model's overall performance. In the context of machine learning, an ablation study can be used to understand the importance of different features, parameters, or layers in a model. Ablation experiments can be performed in a variety of ways, but the most typical are training a model with all of its components enabled and then retraining the model with one or more components disabled. The retrained models' performance is then compared to the original model's performance to identify the impact of the deactivated components.

In Table 3, LSTM-IIMA$_{\text{Com}}$ is the terminology, that the proposed algorithm uses for all the components like genre, gender, and age groups to recommend movies.

TABLE 2. Evaluation metrics for recommendation (%).

Dataset	Metrics	Training	HAN	MAGNN	LSTM-IIMA
IMDB	Macro-F1	50% 80%	52.71 54.38	56.12 57.97	58.21 61.15
	Micro-F1	50% 80%	57.97 58.51	56.91 58.34	58.33 61.91
	AUC	50% 80%	60.01 61.23	57.28 58.91	59.11 62.01
Netflix Movie	Macro-F1	50% 80%	52.92 53.98	57.05 58.18	59.11 60.04
	Micro-F1	50% 80%	56.19 58.83	57.01 58.97	59.03 60.01
	AUC	50% 80%	57.05 59.51	57.61 59.63	60.23 61.05
TV Show	Macro-F1	50% 80%	52.01 53.71	56.32 58.13	59.50 60.50
	Micro-F1	50% 80%	58.17 58.90	57.01 57.84	59.30 61.01
	AUC	50% 80%	58.81 59.13	58.05 59.03	60.25 62.08

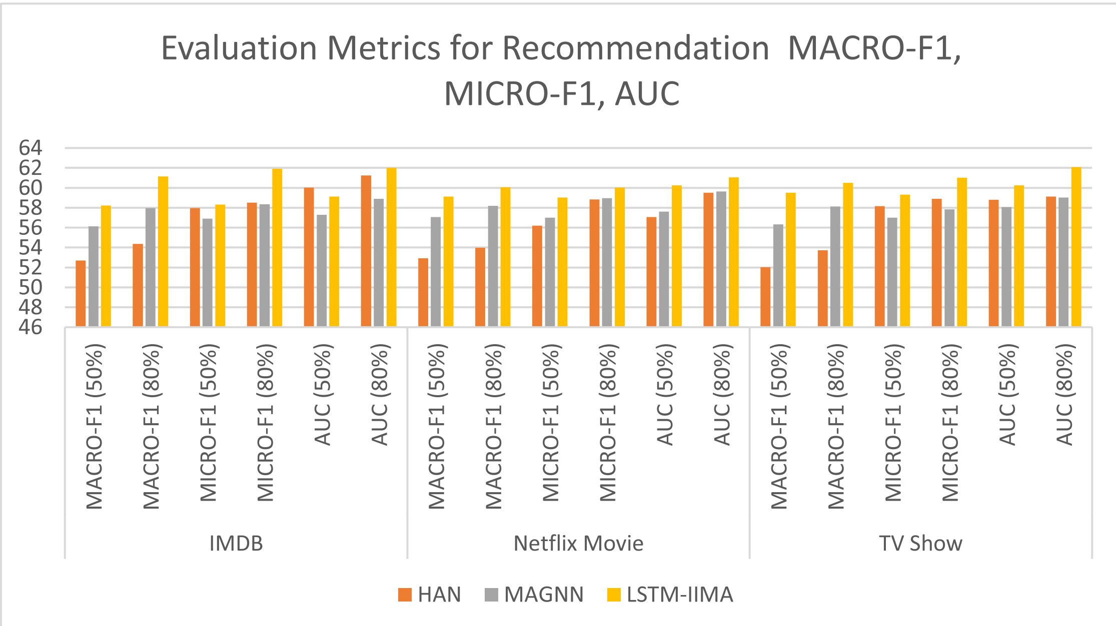


FIGURE 7. Evaluation metrics for recommendation in terms of MACRO-F1, MICRO-F1, AUC.

In LSTM-IIMA$_{\mathrm{GeGr}}$, Ge stands for genre and Gr stands for gender. It tells us that the proposed algorithm only takes gender and genre into account. The same is the case for all other variants. In Table 3, the Macro-F1 and Micro-F1 scores are on the higher side, because LSTM-IIMA$_{\mathrm{Com}}$ includes all the components at the time of the training dataset.

In Ablation Analysis symbols used for ***Genre*** is **Ge, *Gender*** is **Gr and *Age*** is **A.** Other symbols are as follows:

LSTM-IIMA$_{\mathrm{GeGr}}$*:* Ablation Analysis based on ***Genre*** and ***Gender*** components using LSTM-IIMA.

LSTM-IIMA$_{\mathrm{Gr}}$*:* Ablation Analysis based on ***Gender*** components using LSTM-IIMA.

LSTM-IIMA$_{\mathrm{GeA}}$*:* Ablation Analysis based on ***Genre*** and ***Age*** components using LSTM-IIMA.

LSTM-IIMA$_{\mathrm{Com}}$*:* Ablation Analysis based on all the combined components (***Genre, Age and Gender***) using LSTM-IIMA.

Figure 8 provides an ablation analysis for LSTM-IIMA in terms of accuracy in percentage. The components involved in ablation analysis are Genre (Ge), Gender (Gr) and Age (A). Firstly, we perform ablation analysis by using Ge and Gr with LSTM-IIMA in terms of Macro-F1, Micro-F1 and AUC. Secondly, we use Gr with LSTM-IIMA, to observe the impact of precision recall. We also use a combination of Ge

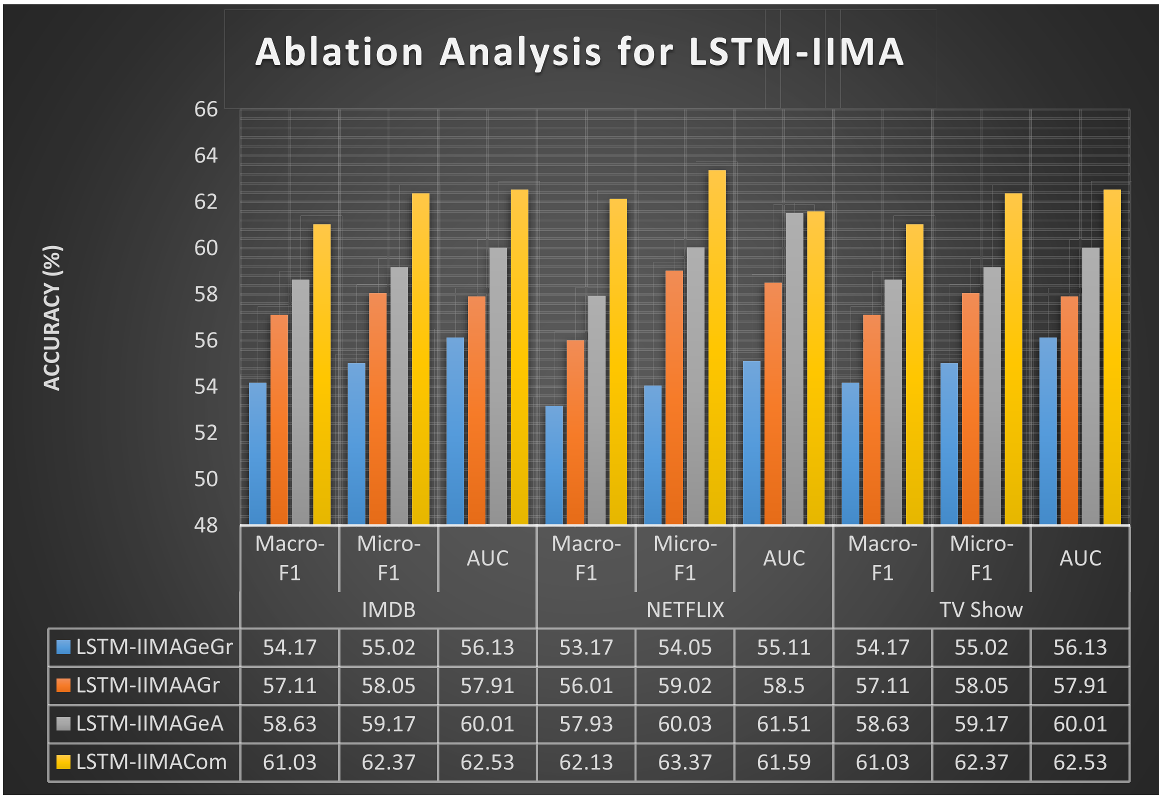


FIGURE 8. Ablation analysis for LSTM-IIMA.

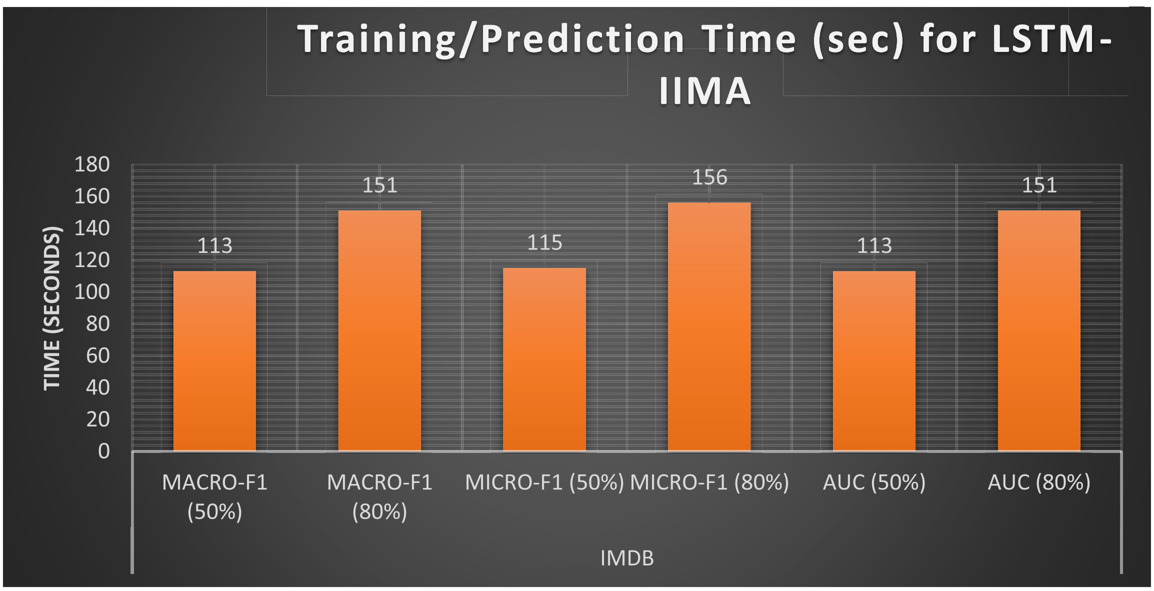


FIGURE 9. Training/Prediction Time (sec) for LSTM-IIMA.

and Age to see the impact of Precision and recall. Lastly, we combine all the components and perform an ablation analysis. By combining all the components, the accuracy in terms of Macro-F1, Micro-F1 and AUC are on the higher side. The results are generated for all 3 datasets IMDB, Netflix and TV Show. The evaluation metrics selected for this simulation are Macro-F1, Micro-F1 and AUC.

In Table 4, the time cost analysis of LSTM-IIMA is presented. In the context of machine learning, time cost analysis refers to the process of assessing and evaluating the

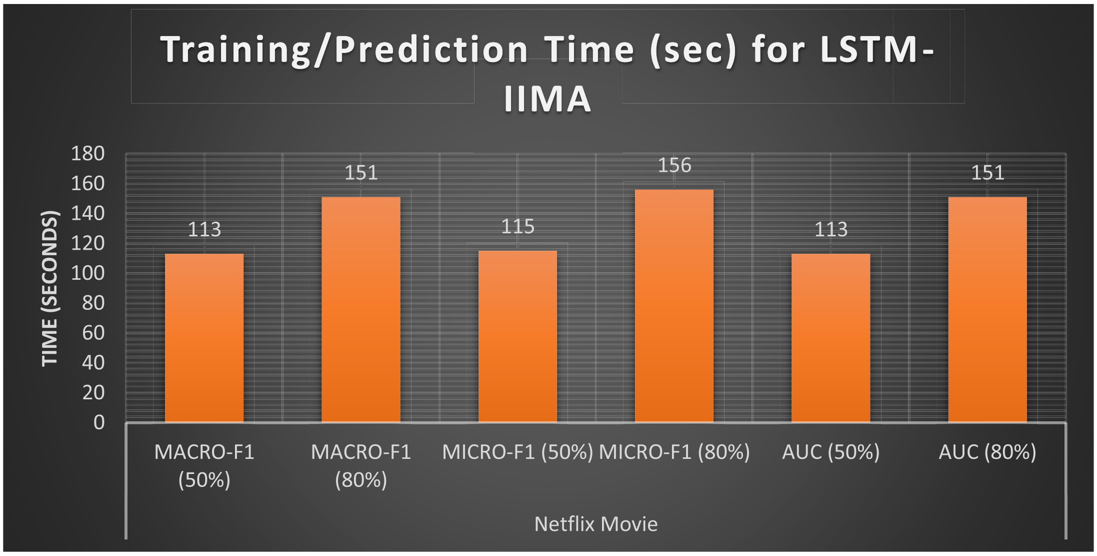


FIGURE 10. Training/Prediction Time (sec) for LSTM-IIMA.

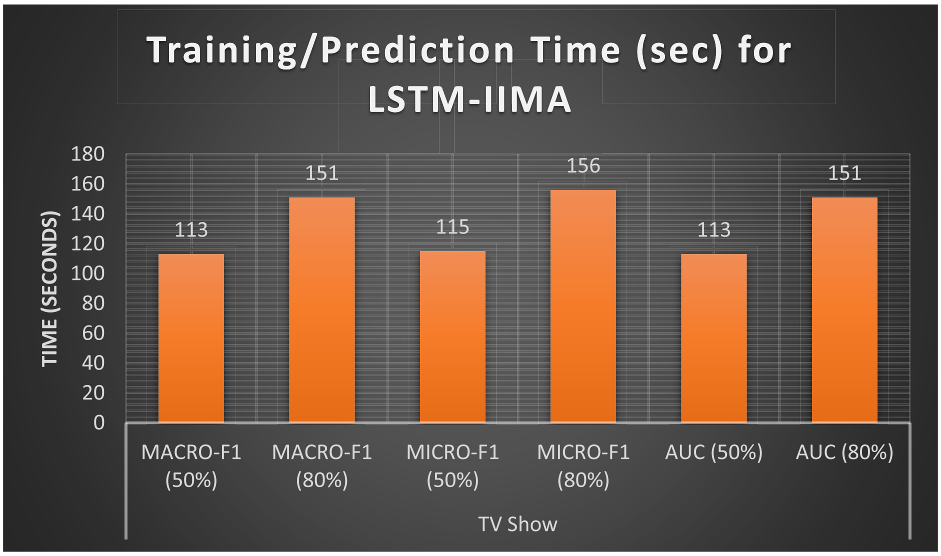


FIGURE 11. Training/Prediction Time (sec) for LSTM-IIMA.

TABLE 3. Quantitative results for ablation analysis (%).

Variant	IMDB			NETFLIX			TV Show		
	Macro-F1	Micro-F1	AUC	Macro-F1	Micro-F1	AUC	Macro-F1	Micro-F1	AUC
LSTM-$IIMA_{GeGr}$	54.17	55.02	56.13	53.17	54.05	55.11	54.17	55.02	56.13
LSTM-$IIMA_{AGr}$	57.11	58.05	57.91	56.01	59.02	58.50	57.11	58.05	57.91
LSTM-$IIMA_{GeA}$	58.63	59.17	60.01	57.93	60.03	61.51	58.63	59.17	60.01
LSTM-$IIMA_{Com}$	61.03	62.37	62.53	62.13	63.37	61.59	61.03	62.37	62.53

computational time required to train and deploy a machine learning model. It involves measuring and analysing the time taken at various stages, such as data preprocessing, feature engineering, model training, tuning, and inference. Training/Prediction time for different sample sizes (50%, 80%) for Macro-F1, Micro-F1 and AUC is given in Table 4.

TABLE 4. Time cost analysis for LSTM-IIMA.

Dataset	Metrics	Training	Training/Prediction Time (sec) for EHGNNs
IMDB	**Macro-F1**	50% 80%	**113** **151**
	Micro-F1	50% 80%	**115** **156**
	AUC	50% 80%	**113** **151**
Netflix Movie	**Macro-F1**	50% 80%	**113** **151**
	Micro-F1	50% 80%	**115** **156**
	AUC	50% 80%	**113** **151**
TV Show	**Macro-F1**	50% 80%	**113** **151**
	Micro-F1	50% 80%	**115** **156**
	AUC	50% 80%	**113** **151**

Figure 9 presents the results of Macro-F1, Micro-F1 and AUC in terms of time. The experimentations are performed by taking 50% and 80% of IMDB dataset records accordingly. The results are measured in seconds on the y-axis showing the time it takes to successfully recommend movies. Figure 10 performs the same time analysis specifically for the Netflix Movie dataset. It also considers 50% and 80 % records for Macro-F1, Micro-F1 and AUC approaches. Figure 11 performs experimentations for Macro-F1, Micro-F1 and AUC on the TV Show dataset. The extracted results from datasets IMDB, Netflix Movie and TV show respectively are in terms of Time (sec) for LSTM-IIMA.

V. CONCLUSION

This study focuses on using LSTM networks for intra and inter-metapath movie recommendation, with a two-level attention mechanism. Using LSTM networks to model inter and intra-metapath interactions in movie recommendation systems provides several benefits and opportunities. LSTM models can capture sequential dependencies, so they can effectively incorporate the temporal dynamics of user-item interactions, resulting in more accurate and personalised movie suggestions. The emphasis on inter and intra-metapath linkages allows for a thorough comprehension of the intricate links between users, films, and numerous contextual elements. The use of LSTM-IIMA to inter and intra-metapath interactions allows for the modelling of long-term user preferences as well as the recording of changing patterns of movie consumption over time. This aids in the comprehension of user interests, preferences, and behavioural alterations, resulting in more exact recommendations. The capacity of LSTM to handle extended sequences is very useful here since it can effectively collect extensive user histories. Using LSTM for intra-metapath connections also investigates short-term dynamics and instantaneous influences on movie tastes. The algorithm can account for the impact of recently consumed films and create recommendations that match users' current preferences by considering the sequential order of user interactions.

However, applying LSTM for inter and intra-metapath interactions in movie recommendation systems presents certain obstacles. Addressing sparse data and the cold start problem, modelling and adapting to temporal dynamics effectively, managing interpretability and explainability of LSTM-based models, incorporating contextual information, ensuring scalability for large-scale datasets, and enabling real-time recommendations are among the challenges. Overcoming these obstacles requires additional study and innovation in the field of recommendation systems. Nonetheless, by addressing these issues, LSTM-based models for inter and intra-metapath relationships have the potential to significantly improve the accuracy, personalization, and effectiveness of movie recommendation systems, providing users with more relevant and engaging movie recommendations.

AVAILABILITY OF DATA AND MATERIALS

Not applicable

COMPETING INTERESTS

All the authors declare that they have no conflict of interest.

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Apr. 2001, pp. 285–295.

[2] W. Wang, C. Ye, P. Yang, and Z. Miao, "Research on movie recommendation model based on LSTM and CNN," in *Proc. 5th Int. Conf. Comput. Intell. Appl. (ICCIA)*, Jun. 2020, pp. 28–32.

[3] C.-Y. Wang, "Gate-tunable van der Waals heterostructure for reconfigurable neural network vision sensor," *Sci. Adv.*, vol. 6, no. 26, Jun. 2020, Art. no. eaba6173.

[4] C. Wang, Y. Wang, X. Jiang, J. Xu, W. Huang, F. Zhang, J. Liu, F. Yang, Y. Song, Y. Ge, Q. Wu, M. Zhang, H. Chen, J. Liu, and H. Zhang, ''MXene Ti3C2Tx: A promising photothermal conversion material and application in all-optical modulation and all-optical information loading,'' *Adv. Opt. Mater.*, vol. 7, no. 12, Jun. 2019, Art. no. 1900060.

[5] A. Wakrime, A. ben Halima, and M. R. Sellami, ''Track report of future internet services and applications (FISA'2020),'' Tech. Rep., 2020.

[6] W. Zhou, D. D. Huang, Y. P. Wu, J. Zhao, T. Wu, J. Zhang, D. S. Li, C. Sun, P. Feng, and x. Bu, ''Stable hierarchical bimetal-organic nanostructures as highperformance electrocatalysts for the oxygen evolution reaction,'' *Angew. Chemie Int. Ed.*, vol. 58, no. 13, pp. 4227–4231, 2019.

[7] B. Li, X. Li, Z. Zhang, and F. Wu, ''Spatio-temporal graph routing for skeleton-based action recognition,'' Tech. Rep., 2019.

[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, ''On the properties of neural machine translation: Encoder–decoder approaches,'' 2014, *arXiv:1409.1259*.

[9] D. Jannach, A. Manzoor, W. Cai, and L. Chen, ''A survey on conversational recommender systems,'' *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, Jun. 2022.

[10] G. Adomavicius and Y. Kwon, ''New recommendation techniques for multicriteria rating systems,'' *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 48–55, May 2007.

[11] Y. Jiang, H. Ma, Y. Liu, Z. Li, and L. Chang, ''Enhancing social recommendation via two-level graph attentional networks,'' *Neurocomputing*, vol. 449, pp. 71–84, Aug. 2021.

[12] D. Ye, Y. Lin, J. Du, Z. Liu, P. Li, M. Sun, and Z. Liu, ''Coreferential reasoning learning for language representation,'' 2020, *arXiv:2004.06870*.

[13] N. Wang, S. Wang, Y. Wang, Q. Z. Sheng, and M. A. Orgun, ''Exploiting intra- and inter-session dependencies for session-based recommendations,'' *World Wide Web*, vol. 25, no. 1, pp. 425–443, Jan. 2022.

[14] J. Wang, X. Hu, J. Wu, J. Ye, and S. Wang, ''Application of parallel acquisition technique combined with echo-planar diffusion-weighted imaging technology in the diagnosis of liver micrometastasis for the patients with prooperation of colon cancer,'' *Chin. J. Primary Med. Pharmacy*, vol. 12, pp. 3070–3071, Jan. 2018.

[15] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, ''Heterogeneous graph attention network,'' in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.

[16] M. A. Russell, *Mining, the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. Sebastopol, CA, USA: O'Reilly Media, 2013.

[17] X. Fu, J. Zhang, Z. Meng, and I. King, ''MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding,'' in *Proc. Web Conf.*, Apr. 2020, pp. 2331–2341.

[18] A.-C. Müller and S. Guido, *Introduction to Machine Learning With Python: A Guide for Data Scientists*. Sebastopol, CA, USA: O'Reilly Media, 2016.

[19] H. Daneshvar and R. Ravanmehr, ''A social hybrid recommendation system using LSTM and CNN,'' *Concurrency Computation, Pract. Exper.*, vol. 34, no. 18, Aug. 2022, Art. no. e7015.

[20] Z. Ali, K. N. Qureshi, A. S. Al-Shamayleh, A. Akhunzada, A. Raza, and M. F. U. Butt, ''Delay optimization in LoRaWAN by employing adaptive scheduling algorithm with unsupervised learning,'' *IEEE Access*, vol. 11, pp. 2545–2556, 2023.

[21] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, ''A survey on long short-term memory networks for time series prediction,'' *Proc. CIRP*, vol. 99, pp. 650–655, Jan. 2021.

[22] Z. Ali, K. N. Qureshi, K. Mustafa, R. Bukhsh, S. Aslam, H. Mujlid, and K. Z. Ghafoor, ''Edge based priority-aware dynamic resource allocation for Internet of Things networks,'' *Entropy*, vol. 24, no. 11, p. 1607, Nov. 2022.

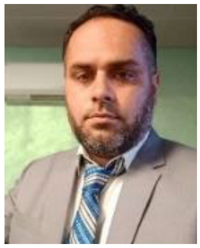
[23] R. Zhang, H. Ma, Q. Li, Y. Wang, and Z. Li, ''Dual-view self-supervised co-training for knowledge graph recommendation,'' in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, Apr. 2023, pp. 113–128.

[24] B. Ramzan, I. S. Bajwa, N. Jamil, R. U. Amin, S. Ramzan, F. Mirza, and N. Sarwar, ''An intelligent data analysis for recommendation systems using machine learning,'' *Sci. Program.*, vol. 2019, pp. 1–20, Oct. 2019.

[25] M. Ibrahim, I. S. Bajwa, N. Sarwar, H. A. Waheed, M. Z. Hasan, and M. Z. Hussain, ''Improved hybrid deep collaborative filtering approach for true recommendations,'' *Comput., Mater. Continua*, vol. 74, no. 3, pp. 5301–5317, 2023.

[26] M. Ibrahim, I. S. Bajwa, N. Sarwar, F. Hajjej, and H. A. Sakr, ''An intelligent hybrid neural collaborative filtering approach for true recommendations,'' *IEEE Access*, vol. 11, pp. 64831–64849, 2023.

[27] N. Aslam, W. Y. Ramay, K. Xia, and N. Sarwar, ''Convolutional neural network based classification of app reviews,'' *IEEE Access*, vol. 8, pp. 185619–185628, 2020.



ZULFIQAR ALI received the M.S. degree in telecommunication and networks (CS) from CUST (MAJU) University, Islamabad, Pakistan, in 2010, and the Ph.D. degree in computer science with an emphasis in low power long range wireless technologies from Bahria University Islamabad (BUI). He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad. His research interests include wireless sensor networks (WSNs), low power wide area networks (LPWAN), the Internet of Things (IoT), artificial intelligence, fiber optics communication, and machine learning/deep learning.

ASIF MUHAMMAD received the M.S. degree in software engineering from the Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan, and the Ph.D. degree in computer science from the Capital University of Science and Technology, Islamabad, in 2023. His profound passion for research is evident through his diverse interests, with a primary focus on cutting-edge areas, such as big data processing. Additionally, he has a keen interest in scheduling mechanisms for heterogeneous computing and optimizing computational resource allocation to enhance performance and efficiency. Previously, he held a position with COMSATS University Islamabad and he is currently with the National University of Computer and Emerging Sciences, Islamabad.

AHMAD SAMI AL-SHAMAYLEH received the master's degree in information systems from The University of Jordan, Jordan, in 2014, and the Ph.D. degree in artificial intelligence from the University of Malaya, Malaysia, in 2020.

He is currently an Assistant Professor with the Faculty of Information Technology, Al-Ahliyya Amman University, Jordan. His research interests include artificial intelligence, human–computer interaction, the IoT, Arabic NLP, Arabic sign language recognition, language resources production, the design and evaluation of interactive applications for handicapped people, multimodality, and software engineering.

KASHIF NASEER QURESHI received the Ph.D. degree from the University of Technology Malaysia (UTM), in 2016. He is currently an Associate Professor in cyber security with the Department of Electronic and Computer Engineering, University of Limerick, Ireland. He is also the Principal Investigator in Cyber Reconnaissance and Combat Project in the Higher Education Commission. His research interests include the Internet of Everything (IoE), the Internet of connected Vehicles (IoV), electronic vehicles charging management planning and recommendation (EV), the Internet of Things (IoT) use cases implementation in wireless and wired networks and cybersecurity, the general areas of cyber security, wireless networks, security perspective perspectives for IoE and IoT networks, and privacy-preserving data aggregation, computer forensics, and cloud security. His name is included top 2 percent for consecutive two years from stand ford university USA. He has published various high-impact factor papers in international journals and conference proceedings and served on several conferences IPCs and journals editorial boards. He has five book chapter books related to security and trust architectures. He has also various projects related to routing and cyber security domains in the U.K., China, Ireland, Malaysia, and Pakistan.

WAGDI ALRAWAGFEH received the Ph.D. degree in computer science (multi-agent systems) from the Memorial University of Newfoundland, Canada. He is currently an Assistant Professor with the College of Computing and IT, University of Doha for Science and Technology, Qatar. He has more than ten years of experience working with environmental aspects directly related to education, programming, and information technology. Through his experiences working as a Teacher Assistant, a Research Assistant, and a Software Developer with Memorial University, Canada, and a Teacher with Al Isra Private University, Jordan, he gained a variety of skills in imparting knowledge regarding both theoretical and practical topics of information technology. During his experience as a computer science teacher, he is successfully communicated the material to master's students and used several teaching techniques to overcome any challenges during the class. His continuous experience in the IT field has enhanced his aspiration to utilize and apply his IT knowledge to business solutions and computerizing business processes. As a result, this aspiration motives him to encourage the students to apply what they learn in the class to solve or automate the business processes in daily life.

ADNAN AKHUNZADA (Senior Member, IEEE) is currently with the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan. He is an enthusiastic and dedicated professional with extensive 12 years of research and development experience both in ICT industry and academia, with demonstrated history and a proven track record of high impact published research (i.e., patents, journals, transactions, books, reputable magazines, chapters, conference, and conference proceedings). His experience as an educator and a researcher is diverse. He is also a Lecturer, a Senior Lecturer, a Year Tutor, and an Assistant Professor with the COMSATS Institute of Information Technology (CIIT), a Senior Researcher with RISE SICs Vasteras AB, Sweden, and an Associate Professor and the Deputy Director with the Namibia University of Science and Technology, Windhoek, Namibia, having mentor-ship of graduate students, and supervision of academic and research and development projects both at UG and PG level. He is rigorously been involved in international accreditation, such as Accreditation Board for Engineering and Technology (ABET), and curriculum development according to the latest guidelines of ACM/IEEE. He is also working as an in-charge of the B.S. Software Engineering (SE) and Telecommunication Networks (TN) Program, CUI. He is also involved in various EU and Swedish funded projects of cyber security on industrial controls systems (ICS), process automation, cyber physical systems (CPS), the Internet of Things (IoT), large scale distributed systems, and modeling and design secure software defined networks. He is a member of technical programme committee of varied reputable conferences and editorial boards. He is also serving as an Associate Editor for IEEE ACCESS.