Research on Personalised Movie Recommendation Systems Based on Knowledge Graphs and LSTM Model

Abstract

With the rapid advancement of internet and information technologies [1], users face the challenge of information overload, giving rise to personalised recommendation systems. This study proposes an innovative approach, integrating knowledge graphs, Long Short-Term Memory (LSTM) networks, and attention mechanisms to construct an accurate and interpretable personalised movie recommendation system.

The research initially constructs a movie domain knowledge graph based on the IMDb dataset, encompassing entities such as films, actors, directors, and genres. Subsequently, an attention-based LSTM model is designed and implemented to capture the dynamic changes in user interests. This model enhances recommendation accuracy, personalization, and interpretability by fusing knowledge graph embeddings with user historical behaviour sequences. To evaluate model performance, extensive experiments were conducted, comparing the proposed model with multiple benchmarks, including traditional SVD matrix factorization, standard LSTM, Neural Collaborative Filtering (NCF), and deep feedforward neural networks. Experimental results demonstrate that, compared to the standard LSTM model, the proposed model reduced Mean Squared Error (MSE) by 36%, increased the coefficient of determination (R^2) by 5.8%, decreased Mean Absolute Error (MAE) by 21.5%, and improved the Explained Variance Score (EVS) by 7%. Furthermore, the model's interpretability was significantly enhanced through visualization using the Neo4j graph database and analysis of attention weights. This study also specifically addressed the cold-start problem in recommendation systems, achieving significant progress in handling new users and new movies through the introduction of knowledge graphs. Concurrently, the research explored potential challenges in practical applications of the model, including computational complexity and data privacy issues.

In summary, this study provides novel ideas and methodologies for the development of personalised movie recommendation systems, exploring the integration of knowledge graphs and deep learning models, while also offering feasible technical solutions for practical applications. Future research will further investigate the fusion of multimodal data, the application of transfer learning in cross-domain recommendations, and strategies for optimising and deploying the model on large-scale datasets.

[1]This paper was originally written in Chinese and translated into English. Some sections follow academic standards specific to Chinese institutions.

Keywords: Personalised Recommendation System, Knowledge Graph, LSTM Model, Attention Mechanism, Interpretability

Contents

1 Introduction

1.1 Research Background

With the rapid development of the internet and information technology, the amount of information users encounter daily has increased exponentially. While this information explosion has led to an abundance of content, it has also triggered the problem of information overload, making it difficult for users to find truly interesting content amidst the vast amount of information. To address this challenge, personalized recommendation systems have emerged. By analyzing users' historical behaviors and preferences, these systems can recommend content that users are likely to be interested in, thereby enhancing user satisfaction and engagement. In the domain of movie recommendations, personalized recommendation systems are particularly significant, as they not only help users discover new films but also improve their viewing experience.

Existing recommendation system technologies mainly include content-based filtering, collaborative filtering, and hybrid recommendation methods[29]. However, these methods often struggle with issues such as data sparsity and the cold start problem. In recent years, emerging technologies like Knowledge Graphs (KG) and LSTM have been widely applied in the research of recommendation systems[18].

A Knowledge Graph is a semantic network that organizes and represents knowledge by depicting the relationships between entities and concepts as a graphical structure. In movie recommendation systems, Knowledge Graphs can represent entities such as movies, actors, and directors, along with their interconnections, thereby enhancing the understanding of the relationships between films and user interests[10]. Knowledge Graphs can capture the complex relationships among entities like movies, directors, and actors, providing rich contextual information that complements the semantic information missing from traditional collaborative filtering methods, enabling the system to have a more comprehensive understanding of the connections between users and films[13].

LSTM networks, as a type of recurrent neural network (RNN), excel at processing time-series data. Compared to traditional Markov models, LSTM networks are better equipped to handle long-range dependencies, which are crucial for understanding the long-term evolution of user interests[4]. LSTMs can effectively capture user behavior sequences such as clicks, favorites, and ratings, thereby gaining deeper insights into user interests and behavior patterns. However, traditional LSTM models face limitations in capturing the complex interrelations among data.

To address this, this study introduces the Attention Mechanism to further enhance the performance and interpretability of the model. The Attention Mechanism dynamically allocates attention weights, allowing the model to focus on critical parts of user behavior data, thereby improving the accuracy and interpretability of recommendation results[11]. Specifically, the Attention Mechanism enables the LSTM model to better capture long-term dependencies by assigning varying importance weights to different time steps in the sequence, allowing the model to focus more on historical information relevant to the current prediction task. This not only increases the model's sensitivity to changes in user interests but also enhances the interpretability of the recommendation results.

By combining Knowledge Graphs and LSTM models with the Attention Mechanism, it is possible to construct a personalized movie recommendation system that is both accurate and interpretable. Integrating the movie information from the Knowledge Graph with the user's behavior sequence not only improves the accuracy and personalization of the recommendation system but also utilizes the Knowledge Graph to explain the recommendation results, thereby enhancing the system's interpretability and making it easier for users to understand the rationale behind the recommendations.

1.2 Significance of the Research

In the context of rapid advancements in information technology, movie recommendation systems serve as crucial tools for enhancing user experience and commercial value, with extensive application prospects, particularly in streaming platforms, online video services, and e-commerce websites. To address the limitations of traditional recommendation systems concerning data sparsity and cold start problems, this research proposes a novel personalized movie recommendation system model by integrating Knowledge Graphs with LSTM models and incorporating an Attention Mechanism. The significance of this study is primarily reflected in several aspects:

First, this research combines Knowledge Graphs with deep learning models, exploring new methodologies for personalized movie recommendation systems. Traditional recommendation systems often rely solely on users' historical behavior data, which limits their ability to fully comprehend the complex relationships between movies and user interests. By integrating Knowledge Graphs, the model enhances the understanding of the correlations among films, improving the accuracy and personalization of recommendations. Furthermore, utilizing the LSTM model to capture user behavior sequences aids in gaining deeper insights into user interests and behavior patterns, thereby enhancing recommendation effectiveness. This approach provides a new solution for personalized movie recommendation systems with significant application value.

Second, from a practical perspective, this research offers an innovative solution for the movie recommendation domain. Personalized recommendation systems are widely applicable in e-commerce, entertainment, and other fields, with movie recommendation systems being a key application that significantly impacts user experience and the development of the film industry. By constructing a personalized movie recommendation system based on Knowledge Graphs and LSTM models, the study aims to accurately meet users' personalized needs, thereby increasing user satisfaction and loyalty, which in turn promotes the growth of the film industry. Additionally, this research provides references and insights for personalized recommendation systems in other domains, expanding the application range of personalized recommendation technologies and holding substantial practical significance.

Finally, by introducing the Attention Mechanism, this research not only improves the accuracy of the recommendation system but also enhances its interpretability, allowing users to better understand the rationale behind the recommendations. Experimental results indicate that compared to standard LSTM models, the Attention Mechanism-based LSTM model proposed in this study reduces the mean squared error (MSE) by 36%, improves the coefficient of determination (R^2) by 5.8%, decreases the mean absolute error (MAE) by 21.5%, and increases the explained variance score (EVS) by 7%. The findings provide solid technical support for practical applications.

1.3 Research Content and Objectives

This study aims to explore the integration of Knowledge Graphs and LSTM models, along with the introduction of an Attention Mechanism, to construct an accurate and interpretable personalized movie recommendation system. The specific research contents include several aspects:

First, constructing a Knowledge Graph. This study will collect knowledge data in the movie domain, including attributes of films (such as genre, actors, directors, etc.) and the relationships among these attributes. Using this data, a movie Knowledge Graph will be built to represent entities like movies, actors, and directors, along with their interrelations[28]. The Knowledge Graph will provide crucial support for the recommendation system by capturing rich semantic information among films.

Second, embedding the Knowledge Graph. The study will employ graph embedding techniques to map the entities and relationships in the Knowledge Graph into a low-dimensional vector space[9]. This approach will encode the semantic information of the movie Knowledge Graph into vector representations, facilitating a better understanding of the correlations among films.

Third, modeling user behavior based on LSTM. The LSTM model will be utilized to model user behavior sequences, collecting user behavior data (such as ratings) within the movie recommendation system and capturing the temporal correlations and characteristics of user interest evolution through the LSTM model. Specifically, this study will introduce the Attention Mechanism to enhance the performance of the LSTM model, allowing it to dynamically focus on key information in users' historical behaviors, thus improving the accuracy and interpretability of recommendations.

Finally, combining the Knowledge Graph and the Attention Mechanism-based LSTM model to construct a comprehensive personalized movie recommendation system. This will involve using the movie information from the Knowledge Graph and the user behavior sequences to generate recommendation lists that meet users' personalized needs.

The main objectives of this research include constructing a Knowledge Graph that encompasses entities such as movies, users, actors, directors, and genres, providing rich semantic information to support the recommendation system. Additionally, the study aims to design and implement a personalized movie recommendation system that integrates Knowledge Graphs and Attention Mechanism-based LSTM models, enhancing recommendation performance through the incorporation of the Attention Mechanism.

Moreover, the proposed model will be evaluated against several baseline models, including traditional SVD matrix decomposition, standard LSTM, Neural Collaborative Filtering (NCF), and deep feedforward neural networks, to validate its effectiveness. Increasing the interpretability of the recommendation system, allowing users to understand the underlying logic of the recommendations, is also a key goal. To comprehensively assess the model's performance, various evaluation metrics will be employed, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Explained Variance Score (EVS), and Median Absolute Error (MedAE). These metrics will help evaluate the model's predictive accuracy, fitting, and stability from different perspectives.

This research aims to provide an innovative solution for a personalized movie recommendation system through practical applications.

1.4 Innovations of the Research

This study presents several innovations in the field of personalized movie recommendation systems:

First, an innovative model architecture is proposed. This research combines Knowledge Graphs with Attention Mechanism-based LSTM models to construct a novel recommendation system architecture. By leveraging the structured knowledge from Knowledge Graphs and the temporal feature capturing abilities of LSTM, the model's expressiveness and interpretability are significantly enhanced.

Second, the integration of multi-source information is achieved. This study creatively combines the structured knowledge from Knowledge Graphs with the temporal dynamics of user behavior sequences to build a more comprehensive user interest representation model. Through this integration method, not only the semantic relationships within the movie domain are considered, but also the dynamic changes in user behavior are effectively captured.

Third, significant improvements are made to address the cold start problem. The introduction of Knowledge Graphs effectively alleviates the cold start issue. Even in the absence of historical data for new users or new movies, the system can provide reasonable recommendations through knowledge inference.

Fourth, the ability to dynamically capture user interests is enhanced. By introducing the Attention Mechanism, this research enables the LSTM model to more accurately capture the dynamic changes in users' short-term interests, while the Knowledge Graph provides stable domain knowledge support. This combination achieves a balance between users' short-term and long-term interests, further improving the predictive accuracy of the recommendation system.

These innovations not only offer new technical solutions for personalized movie recommendation systems but also possess significant practical application value.

1.5 Research Hypotheses and Questions

Based on the aforementioned research background and significance, this study proposes the following research hypotheses: the combination of Knowledge Graphs and deep learning models can effectively enhance the accuracy and interpretability of movie recommendation systems; the introduction of the Attention Mechanism in LSTM models can better capture the dynamic changes in user interests, thereby improving recommendation performance; and Knowledge Graph-based recommendation systems can provide richer explanations for recommendations, enhancing users' understanding and trust in the results.

To validate these hypotheses, this research focuses on the following research questions: How can the Knowledge Graphs in the movie domain be effectively integrated with LSTM models to enhance the performance of the recommendation system? What role does the Attention Mechanism play in capturing users' long-term and short-term interest changes, and how does it impact the overall performance of the recommendation system? What advantages does the LSTM model based on Knowledge Graphs and the Attention Mechanism have in terms of recommendation accuracy and interpretability compared to traditional recommendation methods and other deep learning models?

By addressing these questions, this study aims to explore the applications of Knowledge Graphs, LSTM, and Attention Mechanisms in personalized movie recommendation systems, providing new methods and insights for enhancing the performance and user experience of recommendation systems.

2 Literature Review

This chapter discusses the development of recommendation system technologies, the research applications and advancements of Knowledge Graph and LSTM-based recommendation systems, and their potential in improving recommendation accuracy, interpretability, and personalization. Finally, it reviews mainstream practices of personalized movie recommendation systems both domestically and internationally, providing additional references and insights into the practical use of personalized recommendation systems.

2.1 Literature Review of Recommendation System Technologies

Since the emergence of recommendation system technologies in the 1990s, there has been rapid and significant development. As technology has advanced, recommendation systems have evolved from simple methods to complex intelligent systems. This section provides a comprehensive overview of recommendation system technologies, focusing on historical evolution, key methods, technical challenges, and recent advancements, thereby laying a solid foundation for the theoretical basis and innovations of this study.

Recommendation systems have consistently been a research hotspot in the field of information retrieval and mining[20]. The early stages of recommendation systems primarily focused on three core methods: content-based recommendation, collaborative filtering, and hybrid recommendation methods[5]. Content-based recommendation methods rely on the analysis of project content characteristics, such as directors, actors, and genres. The advantage of this method is its ability to directly utilize explicit features of projects for recommendations, making it particularly suitable for recommending new projects. However, it struggles with data sparsity and capturing the dynamic changes in user interests.

In contrast, collaborative filtering methods recommend based on the analysis of historical behaviors and preferences of similar users. This approach partially addresses data sparsity and helps uncover users' latent interests. However, collaborative filtering still faces the cold start problem, limiting recommendation effectiveness when dealing with new users or projects. To overcome the limitations of singular methods, hybrid recommendation methods emerged. Research indicates that by combining multiple recommendation strategies, hybrid methods effectively integrate the advantages of both content-based and collaborative filtering approaches, significantly improving the accuracy and robustness of recommendation systems[29]. This method not only better addresses data sparsity but also alleviates the cold start problem to some extent, leading to widespread adoption in practical applications.

Roy et al. (2022) propose that with the rapid development of deep learning and big data technologies, deep learning-based recommendation systems have gradually become a research focus[19]. Traditional recommendation methods often struggle with large-scale, high-dimensional data, whereas deep learning methods effectively tackle these challenges. Multi-layer perceptrons (MLP) and convolutional neural networks (CNN) have shown outstanding performance in recommendation tasks, automatically learning feature representations to improve recommendation accuracy and personalization[1]. Particularly, LSTM models demonstrate significant advantages in handling sequential data and temporal correlations. LSTM can effectively capture the long-term dependencies of user behavior, leading to a better understanding of the evolution of user interests and providing more accurate and dynamic recommendations.

In recent years, recommendation system technologies have exhibited diversification and integration trends. The introduction of Knowledge Graphs provides rich semantic information and structured knowledge for recommendation systems, enhancing both accuracy and interpretability. Additionally, the application of attention mechanisms allows models to more precisely capture key features of user interests, further improving recommendation quality. The continuous evolution and innovation of recommendation systems, from early content-based and collaborative filtering methods to hybrid approaches combining multiple strategies and deep learning techniques that mine hidden user behavior features, are driving the development of recommendation systems towards greater efficiency and intelligence.

2.2 Literature Review of Personalized Recommendation Systems Based on Knowledge Graphs

In the study by Guo et al. (2020), it is highlighted that Knowledge Graphs, as powerful semantic networks, are increasingly applied in personalized recommendation systems[7]. This section systematically explores the roles, advantages, challenges, and recent advancements of KGs in recommendation systems, providing a theoretical foundation for the innovative methods in this research.

Knowledge Graphs represent structured knowledge, effectively illustrating complex relationships between entities. As semantic networks, KGs organize and represent relationships between entities and concepts in a graphical structure. Their application in personalized recommendation systems offers significant advantages. KGs provide rich semantic information that allows recommendation systems to better understand the relationships between entities. Additionally, they alleviate data sparsity issues by incorporating external knowledge, particularly excelling in cold-start scenarios. Mezni et al. (2021) indicate that the structured information in KGs enhances the interpretability of recommendations, making it easier for users to understand the basis of recommendations[15]. This enhances personalization, helping systems better comprehend the associations between user interests and item features, thus delivering more accurate personalized recommendations.

In recent years, personalized recommendation systems that integrate KGs have become a research hotspot, demonstrating exceptional performance across various fields. For instance, in movie recommendation, KGs can represent multidimensional relationships among films, such as common directors or similar themes, aiding in the discovery of users' latent interests. In news recommendation, understanding the relationships between news topics allows systems to provide more comprehensive and relevant recommendations. In the e-commerce domain, KGs capture hierarchical relationships and attribute similarities among products, improving recommendation accuracy. Research indicates that personalized recommendation systems combined with KGs not only enhance the accuracy of recommendations but also offer better interpretability, thereby increasing users' trust and satisfaction with the system.

Despite the immense potential of KGs in recommendation systems, their application faces several challenges. Constructing a KG requires collecting and processing vast amounts of data, ensuring completeness and accuracy is a complex task. Solutions include employing semi-automated knowledge extraction techniques combined with manual reviews to ensure KG quality. Graph embedding technologies are also critical for improving recommendation outcomes, focusing on transforming complex relationships within KGs into low-dimensional vector representations. Developing advanced graph embedding algorithms, such as TransE and TransR, is essential for capturing the semantic relationships between entities more effectively.

Moreover, effectively integrating KG information with user behavior data to design end-to-end deep learning models that learn both user behaviors and KG features is crucial for enhancing recommendation performance. Computational efficiency poses another challenge, particularly when performing inference and calculations on large-scale KGs. Efficient graph computing frameworks and parallel processing techniques can optimize algorithm complexity to address these issues. Despite these challenges, combining KGs with deep learning models, such as LSTM networks as used by Pradhan et al. (2021), can leverage the strengths of KGs, especially in handling sequential data and temporal correlations, thereby overcoming these obstacles[16].

2.3 Literature Review of Personalized Recommendation Systems Based on LSTM

LSTM, as an advanced variant of recurrent neural networks, demonstrates exceptional performance in modeling sequential data[14]. This section delves into the applications, advantages, challenges, and recent advancements of LSTM in personalized recommendation systems, providing a theoretical foundation for the innovative methods in this research.

LSTM effectively addresses the vanishing gradient problem that traditional RNNs face when processing long sequences through its introduction of memory cells and gating mechanisms. This unique structure enables LSTM to excel in modeling long-term dependencies[24]. LSTM can selectively remember or forget information, capturing key patterns in user behavior through its gating mechanisms. Furthermore, LSTM's nonlinear activation functions allow it to extract complex nonlinear features, enhancing recommendation accuracy. Its robustness to noise and irrelevant information aids in extracting the essential characteristics of user behavior.

In personalized recommendation systems, existing studies have shown that LSTM is primarily utilized to capture user behavior sequences, such as clicks, favorites, and ratings, to better understand user interests and behavior patterns. LSTM-based personalized recommendation systems have significant advantages in handling time-series features. Specific applications include sequential recommendations, where LSTM models historical user behavior sequences to predict the next actions more accurately[21]. In session-based recommendations, LSTM captures short-term interest fluctuations within a single session, providing real-time, dynamic recommendations. By incorporating contextual information such as time and location, LSTM can generate more personalized and situational recommendations. Additionally, LSTM can learn the correlations between user behavior patterns across different domains, facilitating cross-domain knowledge transfer.

Despite LSTM's strong performance in sequence modeling, several limitations persist in recommendation systems. While LSTM alleviates the vanishing gradient problem, its performance may decline when handling very long sequences. Moreover, LSTM assigns equal weight to all historical states, which may result in insufficient attention to crucial information. Its sequential nature also limits parallel computation capabilities, impacting efficiency when processing large-scale data. Furthermore, LSTM primarily focuses on local contexts, leading to limited modeling of global dependencies. To address these challenges, researchers are exploring the integration of attention mechanisms into LSTM models. The attention mechanism enables models to dynamically focus on important parts of sequences, thereby enhancing sensitivity to changes in user interests and improving recommendation accuracy.

2.4 Attention Mechanisms in Recommendation Systems

Attention mechanisms have emerged as a cutting-edge component in recommendation system research, enabling models to dynamically allocate weights and focus on different parts of the input sequences. The introduction of attention mechanisms aims to address several key challenges faced by LSTM models in recommendation systems, such as modeling long-term dependencies and feature selection. By dynamically assigning importance weights to different time steps or features, attention mechanisms help models more effectively concentrate on relevant information, thereby improving the accuracy and interpretability of recommendations. This section explores the applications, advantages, recent advancements, and integration of attention mechanisms in recommendation systems.

The core idea of attention mechanisms is to simulate the selective nature of human attention by calculating and distributing attention weights, allowing the model to focus on the most relevant and important parts of the input data. In recommendation systems, attention mechanisms typically involve query vectors, key vectors, value vectors, attention scores, attention weights, and weighted sums. These components work together to assign weights based on the similarity between query and key vectors, applying these weights to the value vectors to generate the final attention output.

The advantages of attention mechanisms in recommendation systems can be highlighted in several areas. First, they significantly enhance recommendation accuracy. By assigning more weight to key user behavior data, attention mechanisms can more effectively capture the relationships within user behavior data, optimizing user behavior modeling[3]. Second, attention mechanisms improve the interpretability of the system. Users can visualize the attention weights to understand the rationale behind the recommendations, increasing the system's transparency and user trust. Additionally, attention mechanisms can dynamically adjust the importance of features based on different contexts, making recommendation systems more flexible and adaptive. When dealing with long sequences, attention mechanisms effectively alleviate long-term dependency issues, capturing crucial information. Lastly, attention mechanisms provide a natural framework for multi-modal data fusion, allowing for a more comprehensive construction of user and item representations.

In recent years, recommendation models incorporating attention mechanisms have shown excellent performance across various recommendation tasks. For instance, Gao Guangshang et al. (2024) significantly improved the effectiveness and user satisfaction of movie recommendation systems by introducing attention mechanisms[6]. Zhou Yaowei et al. (2021) proposed a multi-head self-attention mechanism to capture complex patterns in user-item interactions, achieving leading performance[27].

The introduction of attention mechanisms effectively addresses several critical challenges faced by LSTM models in recommendation systems. By dynamically allocating different importance weights to time steps or features, attention mechanisms help models focus more effectively on relevant information, enhancing the accuracy and interpretability of recommendations.

2.5 Mainstream Practices in Personalized Movie Recommendation Systems

With the rapid development of the digital entertainment industry, competition among online movie streaming platforms has intensified. To improve user retention, enhance user experience, and attract new users, major platforms have adopted various strategies, with personalized recommendation systems becoming a core competitive advantage. This section will explore the practices of major online video platforms both domestically and internationally, analyzing their innovations and effectiveness to provide valuable insights for this research.

Netflix is recognized as a leading global streaming service provider, renowned for its powerful personalized recommendation system. Netflix's recommendation system combines deep learning and collaborative filtering algorithms to achieve highly personalized content recommendations. Deep learning models learn complex features and patterns from vast amounts of user behavior data, while collaborative filtering leverages user similarities for recommendations. This hybrid approach effectively captures diverse user interests and improves recommendation accuracy. In addition to the recommendation algorithms, Netflix places significant emphasis on optimizing user experience. For example, it implements seamless playback and automatically plays the next episode to minimize user actions, enhancing viewing continuity. The system also promotes related episodes based on users' viewing history and preferences. Furthermore, Netflix displays different thumbnails for the same content to various users to increase click-through rates. These user experience innovations further enhance the effectiveness of the recommendation system[22]. Netflix also conducts extensive A/B testing to evaluate and optimize its recommendation system, ensuring continuous improvement to meet changing user demands. Through these strategies, Netflix has successfully enhanced user viewing experience and satisfaction, significantly reducing user churn rates.

Hulu, primarily focused on streaming television shows, has a unique recommendation system. Hulu collaborates with traditional television networks to provide a large volume of the latest TV shows and particularly emphasizes recommending newly released content to meet users' demand for timely offerings. Technological innovations at Hulu include the use of deep learning and natural language processing to improve recommendation accuracy. The system analyzes program descriptions, dialogues, and user comments to better understand content features and user preferences. Additionally, Hulu prioritizes exclusive content in its recommendations to attract and retain users. By integrating the latest content, technological innovations, and exclusive content strategies, Hulu has successfully carved out a niche in a competitive market[31].

Platform	Core Strategy	Technical Features	Unique Advantages
Netflix	Deep learning-driven	Deep learning + Collaborative filtering	A/B testing culture, personalized experience
Hulu	Latest content and technology	Deep learning + NLP	Exclusive content strategy
Chinese Platforms	Commercial resources prioritized, diverse content	Machine Learning + in-depth user behavior analysis	Localization strategy, content personalization

Table 1. Summary of Recommendation Strategies Across Platforms

In China, major video platforms such as Youku and iQIYI are also actively implementing personalized recommendation systems, but their strategies and focuses differ slightly from those of international platforms. Chinese video platforms often prioritize recommending commercially valuable content, such as programs with significant advertising placements or paid content, to enhance monetization. Moreover, these platforms emphasize the production and recommendation of exclusive content to differentiate themselves from competitors and increase user engagement[26]. Chinese video platforms typically offer a more diverse and personalized content portfolio, including movies, TV dramas, variety shows, and animations, requiring their recommendation systems to be more flexible to accommodate different content types. By analyzing users' viewing behaviors, search histories, and interaction data, these platforms continuously optimize their recommendation algorithms to enhance viewing time and content consumption. These strategies have successfully increased user activity and the commercial value of these platforms.

Through the study of mainstream personalized movie recommendation system practices, several insights can be gleaned for this research. First, continuous algorithm innovation is key to maintaining a competitive edge. For example, Netflix's application of deep learning demonstrates the importance of advanced algorithms in enhancing recommendation system performance. Second, maintaining content diversity while providing personalized recommendations is crucial for user satisfaction. The Hulu case indicates that combining new content with technological innovation can effectively boost user engagement. Lastly, adjusting recommendation strategies to meet specific market demands is essential for success. The practices of Chinese platforms emphasize the importance of personalized, commercialized, and localized strategies to meet varying user needs.

These insights provide valuable references for this research. In designing and implementing a recommendation model based on knowledge graphs, LSTM, and attention mechanisms, this study will fully consider these practical experiences to build a more effective and user-friendly personalized movie recommendation system. This includes incorporating advanced algorithms and technologies into the model, maintaining content diversity, and adjusting strategies based on specific market needs.

2.6 Literature Review Summary

Through a comprehensive review of recommendation system technologies, knowledge graphs, LSTM models, attention mechanisms, and mainstream practices both domestically and internationally, this study draws several key conclusions. First, recommendation system technology is evolving toward greater personalization, intelligence, and interpretability. Particularly, deep learning techniques, especially LSTM models, demonstrate significant advantages in handling sequential data and capturing user interest evolution. Secondly, the introduction of knowledge graphs provides rich semantic information and structured knowledge to recommendation systems, enhancing both accuracy and interpretability by offering valuable contextual information about movie features and user preferences.

However, existing research still faces several limitations and areas for improvement. One limitation is the lack of integration among technologies: while there have been studies on knowledge graphs, LSTM, and attention mechanisms separately, a cohesive combination of these three technologies remains underexplored, hindering the realization of their synergistic advantages. Additionally, modeling long-term and short-term interests continues to be challenging. In the realm of movie recommendations, user interests can change rapidly over time, and current methods struggle to capture both dynamic shifts and stable preferences simultaneously. Furthermore, the lack of interpretability is a notable issue; as model complexity increases, providing intuitive and comprehensible recommendations becomes more difficult, particularly with the introduction of deep learning models.

To address these shortcomings, this research proposes a novel personalized movie recommendation system model that innovatively integrates knowledge graphs with attention-based LSTM techniques. By leveraging knowledge graphs to capture rich semantic information in the movie domain and enhance model interpretability, the model employs LSTM to effectively model user interest variations over both the long and short term, incorporating attention mechanisms to further refine expression capabilities and explanations. This multi-technology integration approach aims to overcome existing research limitations, offering more accurate, personalized, and interpretable movie recommendations. The following chapters will provide a detailed account of the model architecture, experimental design, and result analysis to validate the effectiveness and superiority of this approach.

3 Data Processing and Model Design

This chapter provides a detailed overview of the data processing workflow and recommendation system model design in this research. It begins with an introduction to the IMDb movie dataset and the supplementary user rating dataset, followed by an analysis of their features. Next, the data preprocessing steps are discussed, including data cleaning and feature engineering. The chapter also outlines the method for constructing the knowledge graph.

The core section of this chapter focuses on the design of the recommendation system model. It presents a detailed account of the proposed attention-based LSTM model, including its design philosophy, network architecture, and mathematical representation. Special emphasis is placed on the innovative integration of the knowledge graph with the model. Additionally, several benchmark models for comparison are introduced.

These steps lay the foundation for subsequent model training, evaluation, and comparative analysis.

3.1 Dataset Description

This study utilizes a non-commercial dataset provided by the Internet Movie Database (IMDb) alongside a supplementary user rating dataset. IMDb is one of the largest and most popular databases for film and television program information worldwide, encompassing extensive details about movies, TV shows, video games, and more[17]. The IMDb dataset is renowned for its comprehensiveness and authority, providing invaluable resources for research on movie recommendation systems. These datasets offer rich movie information and user behavior data essential for capturing the dynamic changes in user preferences, particularly for the attention-based LSTM model proposed in this study.

The non-commercial IMDb dataset consists of several subsets, each provided in a compressed tab-separated values (TSV) format and encoded using the UTF-8 character set. These datasets cover various aspects of films, including but not limited to:

- Basic Information (title.basics.tsv.gz): Contains the unique identifier (tconst) for each movie, type, primary title, original title, adult content flag, start year, end year (for TV series), runtime, and genres.
- Rating Information (title.ratings.tsv.gz): Includes the average rating and number of ratings for each movie.
- Principal Information (title.principals.tsv.gz): Lists the main cast and crew associated with each movie.
- Director and Writer Information (title.crew.tsv.gz): Provides details about the directors and writers of each film.
- Episode Information (title.episode.tsv.gz): Offers detailed information for each episode in a TV series.
- Personnel Basic Information (name.basics.tsv.gz): Contains basic information about individuals such as actors and directors, including birth year, death year, and primary profession.

These subsets are interconnected through unique identifiers (e.g., tconst and nconst), forming a comprehensive network of movie information. This structure enables the study to analyze film characteristics from multiple perspectives, providing robust data support for the development of sophisticated recommendation algorithms.

3.1.1 Description of the User Rating Dataset.

In addition to the official dataset provided by IMDb, this study employs a supplementary user rating dataset (user_rating.csv). This dataset contains approximately 4.67 million rating records from around 1.5 million users for 350,000 movies[2]. Each record includes the following information:

- userID: A unique identifier for the user.
- tconst: A unique identifier for the movie, corresponding to the tconst in the official IMDb dataset.
- rating: The user's rating for the movie, ranging from 1 to 10.
- review date: The date on which the user submitted the rating.

The uniqueness of this dataset lies in its provision of temporal information, allowing this study to analyze the evolution of user preferences over time and the dynamic changes in movie popularity. This is particularly valuable for developing recommendation algorithms capable of capturing temporal effects.

3.1.2 Feature Analysis of the Dataset.

To gain a deeper understanding of the dataset's characteristics, this study conducted preliminary statistical analyses using Python's pandas and numpy libraries. The main findings are as follows:

Firstly, the distribution of ratings indicates that most ratings are concentrated in the upper middle range. Specifically, the average rating is 6.70, with a standard deviation of 2.59 and a median of 7.0. The 25th and 75th percentiles are 5.0 and 9.0, respectively, indicating a slight positive skew in the distribution.

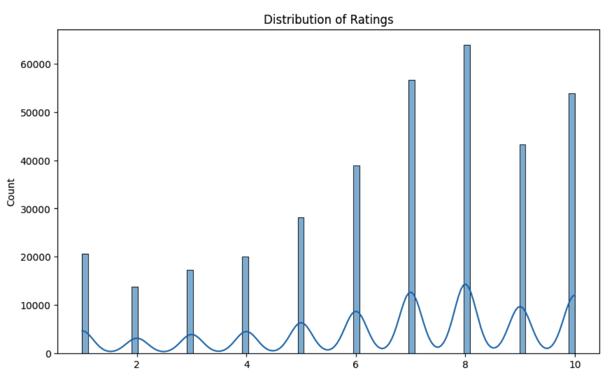

Figure 1. Histogram Distribution of Ratings

Secondly, an analysis of user activity reveals that the average number of ratings per user is 19.85, with a standard deviation of 12.18, a minimum of 2, and a maximum of 50. This suggests significant variability in user activity, which may need to be considered in the recommendation algorithm.

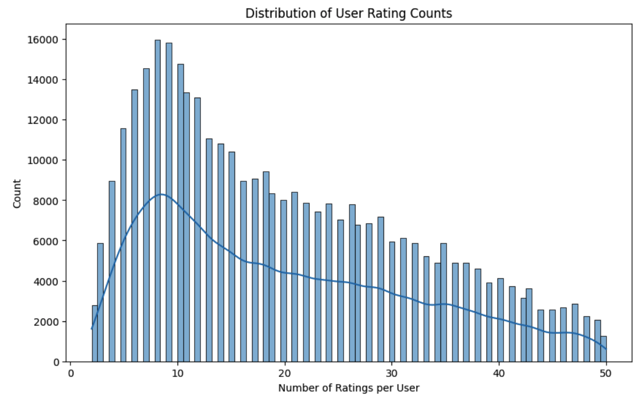

Figure 2. Histogram Distribution of User Activity

Regarding movie popularity, the average number of ratings per movie is 20.03, with a standard deviation of 10.48, a minimum of 1, and a maximum of 30. This variance may lead to a long-tail effect, a common challenge in recommendation systems.

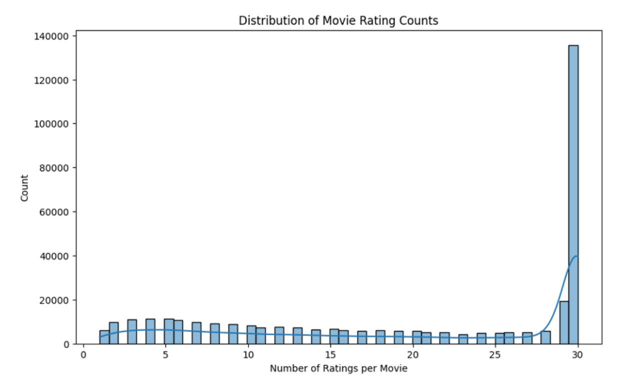

Figure 3. Histogram Distribution of Rating Counts

Analysis of temporal trends indicates a general decline in average ratings from 1998 to 2020. The average rating in 1998 was 7.36, dropping to 6.65 in 2010 and further to 6.53 in 2020. This trend may reflect changes in user rating standards or the evolution of movie quality, which should be considered in time series analysis.

Lastly, the monthly average ratings show slight seasonal fluctuations, with March having the highest average rating of 6.76, while August has the lowest at 6.64. This seasonal pattern may be related to movie release strategies or viewing habits, providing a basis for time-sensitive recommendation algorithms.

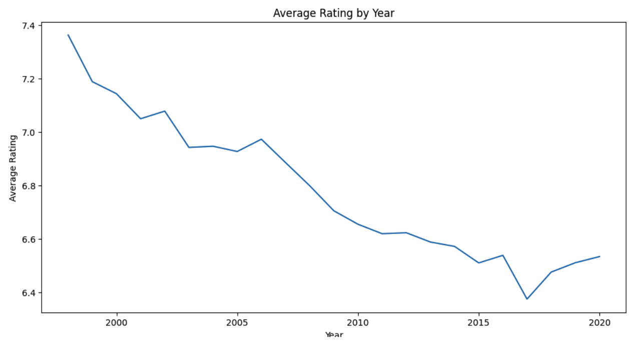

Figure 4. Trend of Annual Average Ratings

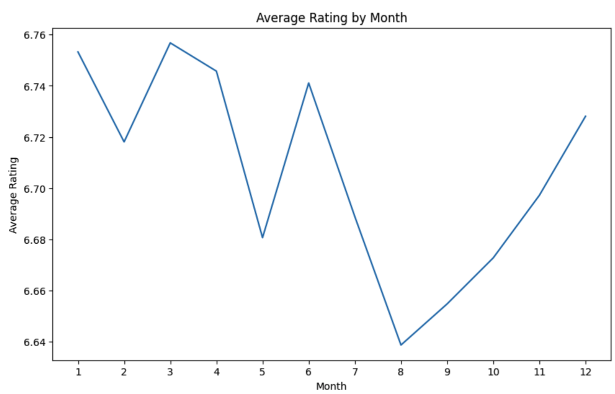

Figure 5. Trend of Monthly Average Ratings

3.1.3 Dataset Quality Assessment. Evaluating the quality of the dataset is crucial for ensuring the reliability of research outcomes. This study assesses the dataset quality based on the framework proposed by Pipino et al. (2002), focusing on several key aspects: completeness, accuracy, consistency, timeliness, representativeness, and privacy, ensuring that the dataset provides a solid foundation for model training and research[16].

Firstly, in terms of completeness, the official IMDb dataset covers various aspects of movies, including basic information, ratings, and cast details, providing comprehensive movie metadata. Additionally, the user rating dataset contains a substantial number of rating records, offering rich training data for personalized recommendations.

Regarding accuracy, IMDb is recognized as an authoritative movie database, resulting in high accuracy for its official dataset. While user ratings may contain some subjectivity and bias, the large volume of data helps to mitigate these issues. A random sampling check conducted in this study found no significant data entry errors.

In terms of consistency, the dataset employs unified identifiers (such as tconst) to link different subsets, ensuring data coherence. Furthermore, ratings adhere to a standardized scale of 1 to 10, reinforcing the consistency of the rating data.

Concerning timeliness, the user rating dataset includes rating dates, facilitating time series analysis and allowing for the capture of temporal changes in rating behavior. From a representativeness perspective, the dataset encompasses a wide array of movies and users, demonstrating good representativeness that reflects diverse user behaviors and movie characteristics. Finally, regarding privacy, information about users and cast members has been anonymized to protect personal privacy.

Overall, the combination of the IMDb dataset and user rating data provides a high-quality and comprehensive data foundation for research on movie recommendation systems. The scale, diversity, and temporal span of the dataset enable in-depth analysis of user preferences and movie characteristics, offering rich material for developing and evaluating recommendation algorithms. These features will guide this study in adopting appropriate strategies in subsequent model design and implementation.

3.2 Data Preprocessing

Data preprocessing is a critical step in machine learning and recommendation system research. This section discusses the data cleaning and feature engineering processes applied to the IMDb movie rating dataset, aiming to enhance data quality, improve model predictive capabilities, and provide high-quality input features for subsequent model training.

3.2.1 Data Cleaning and Filtering. The primary objective of data cleaning is to handle missing values, outliers, and ensure data consistency and reliability. This study follows these cleaning steps: Firstly, for missing values in the 'startYear', 'directors', and 'writers' columns (such as empty entries or '
N'), these records were removed to maintain the integrity of time-related features. The 'startYear' was converted to an integer type, eliminating any potential decimal suffixes, while 'review_date' was converted to a datetime type for easier extraction of temporal features.

In the data filtering step, only records with the movie type (titleType == 'movie') were retained, excluding other types such as TV shows and short films. Additionally, thresholds for a minimum rating (4.5) and minimum number of ratings (500) were set to filter out high-quality movie data. During the consistency check, it was ensured that the 'tconst' (movie ID) was consistent across all relevant files (movies, cast, directors, genres, grades, user_ratings), retaining only those movie records present in all files. Finally, duplicate user rating records were removed to ensure each user has only one rating per movie.

3.2.2 Feature Engineering. Feature engineering is the process of transforming raw data into features that better represent the underlying problem, significantly influencing model performance. This study designed and implemented various types of features to comprehensively capture aspects of the movie rating system.

First, basic feature extraction involved obtaining features directly from the raw data, including user ID (user_id_encoded), movie ID (movie_id_encoded), movie year (year), directors (directors), cast (cast), and genres (genres). These features provide raw information about users, movies, and their attributes.

Time series features aim to capture patterns in rating behavior over time, including review year (review_year), review month (review_month), quarter (quarter), day of the week (day_of_week), days since the last review (days_since_last_review), and week of the year (week_of_year). These features help the model understand temporal patterns in rating behaviors, such as seasonal trends or periodic fluctuations.

User-item interaction features reflect the relationship between users and movies, covering user rating count (user_rating_count), movie rating count (movie_rating_count), user-movie rating differences (user_movie_rating_diff), and average ratings for specific genres (user_genre_avg_rating). These features assist in capturing user rating tendencies and movie popularity.

Statistical features provide deeper insights into data distribution, including user rating standard deviation (user_rating_std), average movie rating (averageRating), movie rating count (numVotes), exponentially weighted moving average of user ratings (user_rating_ewma), and exponentially weighted moving average of movie ratings (movie_rating_ewma). These features help the model grasp overall distribution and trends in the data.

Lastly, cyclical features aim to capture recurrent patterns in rating behavior, such as hour sine (hour_sin) and hour cosine (hour_cos). The design of these cyclical features aids the model in recognizing rating patterns at different times of the day.

Through this series of feature engineering steps, the raw IMDb movie rating data has been transformed into a multidimensional, information-rich feature set. These features encompass not only basic user and movie information but also time series, user-item interactions, statistical, and cyclical aspects, providing a comprehensive and detailed input for subsequent recommendation models.

The figure above displays the correlation among the main features after feature engineering. Notable patterns can be observed, such as the positive correlation between user rating count and user rating standard deviation, indicating that users with more ratings may exhibit a greater range of rating variability. Additionally, a positive correlation exists between average movie rating and rating count, which may reflect the phenomenon that popular movies often receive higher ratings.

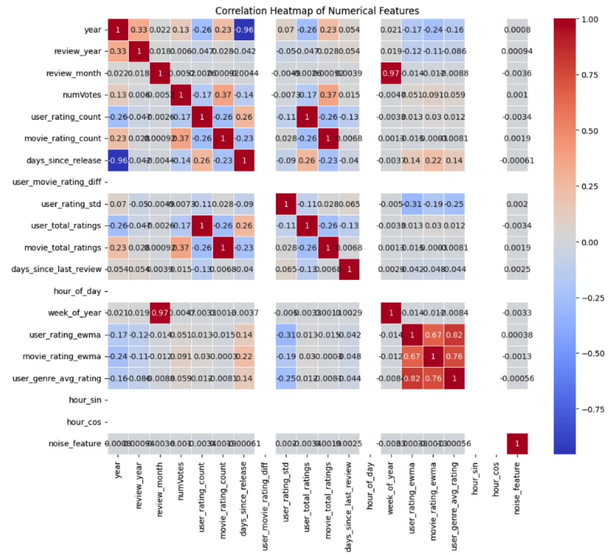

Figure 6. Correlation Heatmap of the Feature-Engineered Dataset

Overall, through these data preprocessing and feature engineering steps, this study significantly enhanced the quality and informational content of the data, laying a solid foundation for subsequent model training and evaluation. These carefully designed and extracted features will aid in capturing the complex patterns within the movie rating system, thereby improving the accuracy and effectiveness of the recommendation system.

3.2.3 Data Integration. Data integration is the process of combining information from multiple data sources into a unified and consistent dataset. In this study, various data files from IMDb were integrated, including basic movie information, ratings, cast, directors, and user ratings. The main steps for data integration are as follows:

First, the integration of basic movie information was performed by reading the 'title.basics.tsv.gz' file, extracting essential details such as 'tconst' (movie ID), 'primaryTitle' (primary title), 'originalTitle' (original title), and 'startYear' (release year).

Next, the ratings data was processed. From the 'title.ratings.tsv.gz' file, the average rating ('averageRating') and the number of votes ('numVotes') were extracted and merged with the basic movie information using the movie ID ('tconst').

For integrating cast and director information, this study extracted data from the 'title.principals.tsv.gz' and 'name.basics.tsv.gz' files, linking this information back to the basic movie details through the movie ID ('tconst').

The integration of user rating data was handled by extracting relevant information from the 'user_rating.csv' file, which included user IDs, movie IDs, ratings, and rating dates. This data was then associated with the movie information.

Finally, the genres field was extracted from the basic movie information, processed, and prepared for subsequent analysis. Through these steps, this study successfully consolidated data from various files into a unified framework.

3.2.4 Data Filtering and Sampling.

Data filtering and sampling are crucial steps in data preprocessing aimed at improving data quality, reducing noise, and ensuring the dataset's representativeness. In this study, a series of filtering and sampling strategies were employed to optimize dataset quality.

To ensure that the analysis and modeling are based on high-quality movie data, strict quality filtering conditions were established. Movies with an average rating of at least 6 and a minimum of 10 ratings were retained. This filtering process focuses on films that have garnered attention and received relatively high ratings, thereby enhancing the reliability of the recommendation system.

Next, to emphasize influential actors and directors, thresholds for occurrence were set. Actors needed to appear in at least two high-quality movies, while directors were required to have directed at least one high-quality film. This filtering method allows the study to focus on active and influential figures in the industry, thereby enhancing the model's predictive capabilities.

In terms of user rating data processing, a strategy of balance and representativeness was adopted. Each user was required to have at least two rating records, and each movie needed to be rated at least once. This approach effectively removed inactive users and movies with insufficient ratings, thus improving the reliability and representativeness of the data.

Through these carefully designed filtering and sampling strategies, not only was the dataset quality improved, but the representativeness and reliability of the data were also ensured. This solid foundation supports subsequent feature engineering and model training, contributing to the performance and reliability of the final recommendation system.

It is noteworthy that the selection of these thresholds was based on careful analysis of data distribution and multiple experimental outcomes. The study aimed to maintain sufficient sample sizes while ensuring data quality and model performance, which is crucial for building an accurate and reliable recommendation system.

3.2.5 Data Encoding and Standardization.

Data encoding and standardization involve converting raw data into a format that machine learning algorithms can directly utilize. In this study, the following encoding and standardization methods were employed:

Firstly, for categorical feature encoding, the 'LabelEncoder' was used to encode categorical features like user IDs and movie IDs, converting string-type IDs into integer format. For multi-category features such as movie genres, directors, and actors, One-Hot encoding was applied. This method transforms categorical variables into binary vectors, effectively avoiding ordinal issues associated with categorical variables.

In terms of numerical feature standardization, features such as year and number of ratings were standardized using 'StandardScaler', which converts feature values into a distribution with a mean of 0 and a variance of 1. For temporal features, year, month, and day were extracted, and cyclic encoding was applied to hours to better capture the periodic nature of these features.

Through these encoding and standardization steps, the raw data was transformed into a format suitable for machine learning algorithms. This not only increased the training efficiency of the model but also prevented features from dominating the model due to differing numerical ranges.

Overall, through data integration, filtering and sampling, and encoding and standardization, the original IMDb movie data was transformed into a structured, high-quality dataset. These processing steps will facilitate the construction of a more accurate and reliable movie recommendation system.

3.3 Knowledge Graph Construction

This section details the construction process of the knowledge graph used in this study, including the definition of entities and relationships, data mapping and transformation, the import process into the Neo4j graph database, and the mathematical representation of the knowledge graph.

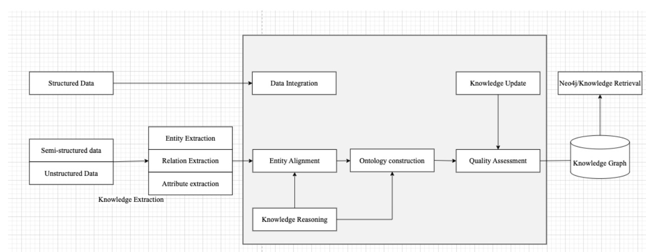

Figure 7. Knowledge Graph Construction Process

3.3.1 Definition of Entities and Relationships.

A knowledge graph focused on the movie domain was constructed based on the IMDb dataset. By analyzing the structure and content of the dataset, the primary entities and relationships were defined. The entities include:

Movie: Represents the films in the dataset. User: Represents users who rate movies. Actor: Represents actors who appear in the films. Director: Represents directors who direct the films. Genre: Represents the various genres of the films.

These entities comprehensively represent the core components of the movie domain[8].

Four main relationship types were defined:

RATED: Represents the rating relationship between users and movies. ACTED_IN: Represents the relationship between actors and the movies they participate in. DIRECTED: Represents the relationship between directors and the movies they direct. HAS_GENRE: Represents the genre classification of movies.

These definitions effectively capture the complex relational network within the movie domain[12].

3.3.2 Data Mapping and Transformation.

To convert the original IMDb dataset into a format suitable for knowledge graph construction, systematic data mapping and transformation operations were performed. First, data cleaning was conducted, which involved handling missing values, removing duplicates, and standardizing data formats to ensure data quality.

Next, relevant entities were extracted from the raw data. For instance, movie entities were extracted from the 'title.basics.tsv.gz' file, while actor and director entities were obtained from the 'name.basics.tsv.gz' file.

For relationship construction, associations between data files were analyzed. The 'title.principals.tsv.gz' file was utilized to establish the ACTED_IN relationship between actors and movies. Additionally, attributes were defined for each entity and relationship. For example, the movie entity includes attributes such as a unique identifier ('tconst'), primary title ('primaryTitle'), and release year ('startYear').

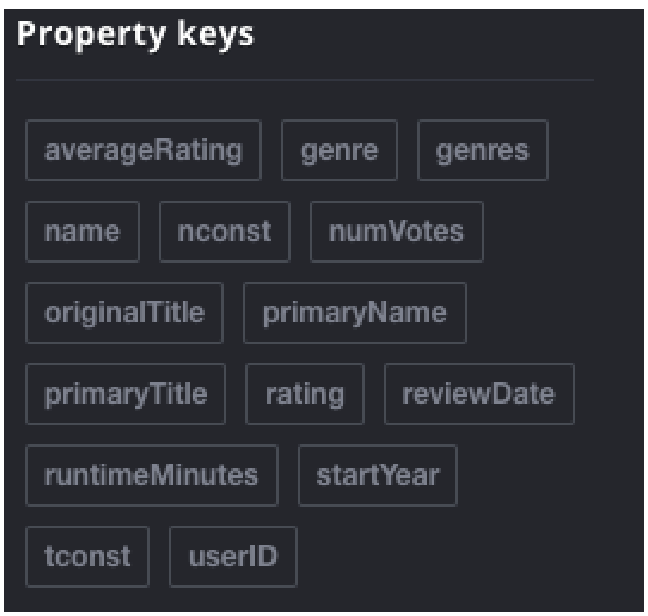

Figure 8. Completed Attribute List

Finally, the processed data was converted into a CSV format that could be directly imported into Neo4j. This series of operations ensured that structured data suitable for constructing the knowledge graph was extracted from the original IMDb dataset and successfully imported into the Neo4j graph database. The resulting knowledge graph contains entities such as movies, users, actors, directors, and genres, along with their complex interrelationships.

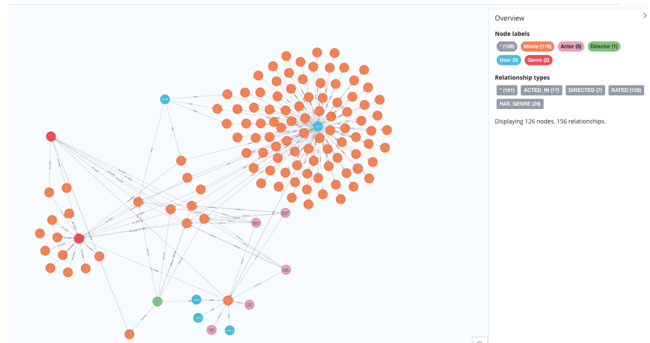

Figure 9. Completed Knowledge Graph of Complex Relationships

3.3.3 Mathematical Representation of the Knowledge Graph.

To better understand and analyze the constructed knowledge graph, this study employs mathematical representation methods to describe the graph's structure and features. First, the knowledge graph can be formalized as a directed graph:

$$G = (V, E) \tag{3.1}$$

where V is the set of nodes representing all entities (movies, users, actors, directors, genres), and E is the set of edges representing the relationships between these entities (RATED, ACTED_IN, DIRECTED, HAS_GENRE). The structure of the graph can be represented by an adjacency matrix. Given a graph with n nodes, it can be represented using an $n \times n$ adjacency matrix A:

$$A[i, j] = \begin{cases} 1 & \text{if there is an edge from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Additionally, considering that the knowledge graph in this study contains multiple types of relationships, it can be represented using a three-dimensional tensor T:

$$T[i, j, k] = \begin{cases} 1 & \text{if there exists the } k\text{th type of relationship from entity } i \text{ t} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

To facilitate subsequent machine learning tasks, each entity and relationship in the graph can be represented as low-dimensional vectors. For an entity e, this study can represent it using the vector:

$$v_e \in \mathbb{R}^d \tag{3.4}$$

where d is the dimension of the embedding space. For knowledge graph completion, given a head entity h, a relation r, and a tail entity t, a scoring function $f(h, r, t)$ can be defined to evaluate the plausibility of the triplet (h, r, t). Common scoring functions include:

TransE:

$$f(h, r, t) = -\|h + r - t\| \tag{3.5}$$

DistMult:

$$f(h, r, t) = \langle h, r, t \rangle \tag{3.6}$$

where $\langle h, r, t \rangle$ denotes the trilinear product. Additionally, random walks are an effective method for exploring graph structures and generating sequences of nodes. Given the current node v, the probability of transitioning to a neighbor node u can be expressed as:

$$P(u|v) = \begin{cases} \frac{1}{\deg(v)} & \text{if } (v, u) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

where $\deg(v)$ is the degree of node v.

These mathematical representations provide the theoretical foundation for analyzing and leveraging the knowledge graph, enabling the application of various graph algorithms and machine learning techniques to uncover patterns and relationships within the graph. This supports subsequent recommendations and other applications.

Through these steps, this study successfully constructed a comprehensive knowledge graph in the movie domain. This knowledge graph not only contains rich information about movies but also captures the complex relational network among movies, users, actors, directors, and genres.

3.4 Recommendation System Model Design

This section provides a detailed overview of the design process for the recommendation system model used in this study, including the overall model architecture, the design concepts, structure, and mathematical representation of the LSTM model enhanced by attention mechanisms, as well as a description of several baseline models for comparison.

3.4.1 LSTM Model with Attention Mechanism. This research designs a movie recommendation system model based on deep learning, primarily utilizing an LSTM neural network integrated with attention mechanisms. The model aims to leverage users' historical behavior sequences and movie feature information to provide more accurate and personalized movie recommendations. The key features of the model include capturing the temporal dependencies of users' viewing histories using LSTM, employing attention mechanisms to differentiate the importance of various historical behaviors, integrating multi-dimensional features from users, movies, and context, and implementing end-to-end learning from raw features to final predictions[25].

The LSTM network serves as the foundational component of the model, effectively capturing long-term dependencies crucial for modeling users' prolonged viewing histories. The choice of LSTM as the base model is primarily due to its capacity for long-term dependency, selective memory, sequential processing capabilities, and gradient stability. The gated mechanism of LSTM allows the model to selectively remember or forget information, aligning with the variability in user preferences over time. Compared to traditional RNNs, LSTM mitigates the vanishing gradient problem, facilitating more effective model training.

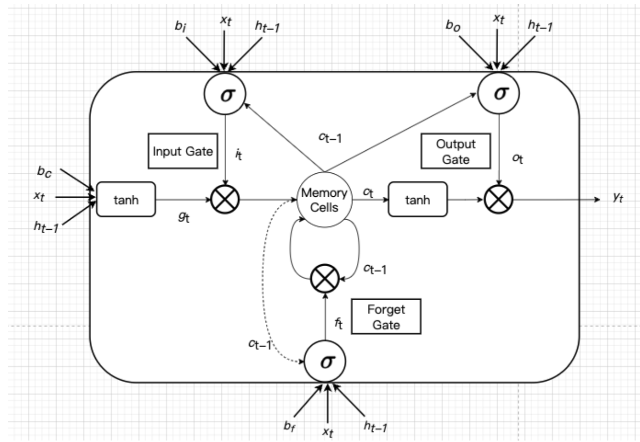

Figure 10. LSTM Network Model

Although the LSTM model effectively handles sequential data, it still encounters challenges when processing long sequences. To further enhance model performance, this study introduces the attention mechanism. The main advantages of the attention mechanism include importance weighting, capturing long-distance dependencies, interpretability, and computational efficiency. It enables the model to assign different importance weights to various historical behaviors, effectively capturing long-distance dependencies and overcoming LSTM's limitations in handling very long sequences. By analyzing attention weights, this research gains insights into which historical behaviors the model prioritizes when making recommendations, thus enhancing the model's interpretability. Compared to merely increasing the number of LSTM layers, the attention mechanism offers a more efficient means of boosting model performance.

The LSTM model with attention mechanism primarily comprises several components: an input layer (including embeddings for user features, movie features, and context features), an LSTM layer (processing users' historical behavior sequences), an attention layer (calculating the importance weights of different historical behaviors), a fusion layer (integrating LSTM outputs and attention-weighted results), and a fully connected layer (performing the final score prediction)[23].

This structural design allows the model to effectively utilize users' historical behavior information while employing

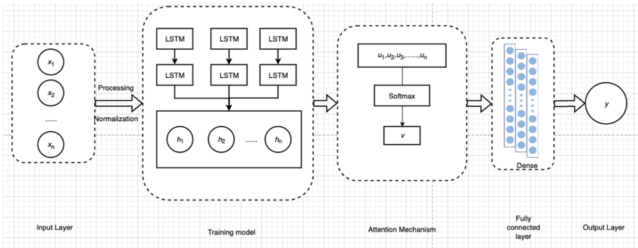

Figure 11. LSTM with Attention Mechanism Structure

the attention mechanism to weight the importance of different historical actions, thereby improving the accuracy and personalization of recommendations. The LSTM layer captures long-term dependencies in users' viewing histories, while the attention layer highlights significant historical behaviors, enhancing the model's understanding of current user preferences. The fusion layer and fully connected layer further consolidate this information, generating the final score predictions.

Through this design, the model can adeptly respond to dynamic changes in user interests, placing appropriate emphasis on recent behaviors while considering long-term patterns. This results in more timely and accurate movie recommendations.

The model's components can be described mathematically to precisely define its structure and operational processes:

1. Input Embeddings: User embedding:

$$e_u = E_u(u) \tag{3.8}$$

Movie embedding:

$$e_m = E_m(m) \tag{3.9}$$

Feature vector:

$$f = [e_u; e_m; \text{"context features"}] \tag{3.10}$$

where E_u is the mapping function for user embeddings, E_m is the mapping function for movie embeddings, and "context features" represents the concatenation of user embeddings, movie embeddings, and contextual features.

2. LSTM Layer:

$$h_t, c_t = \text{LSTM}(x_t, h_{t-1}, c_{t-1}) \tag{3.11}$$

where x_t is the input at time t, and h_t and c_t are the hidden state and cell state, respectively.

3. Attention Layer:

$$\alpha_t = \text{softmax}(v^T \tanh(W_h h_t + W_s s + b)) \tag{3.12}$$

$$c = \sum \alpha_t h_t \tag{3.13}$$

where α_t is the attention weight, c is the context vector, and W_h, W_s, v are learnable parameters.

4. Fusion Layer:

$$z = [h_T; c] \tag{3.14}$$

where h_T is the output from the last time step of the LSTM, and c is the attention-weighted context vector.

5. Fully Connected Layer:

$$y = \sigma(W_2 \text{ReLU}(W_1 z + b_1) + b_2) \tag{3.15}$$

where W_1, W_2, b_1, b_2 are parameters of the fully connected layer, and σ is the activation function.

6. Loss Function:

$$L = \text{MSE}(y, y_{\text{true}}) + \lambda \|\theta\|_2 \tag{3.16}$$

where y_{true} is the true rating, θ represents all model parameters, and λ is the L2 regularization coefficient.

This mathematical representation clearly delineates each component of the model and its operational process. With this design, the model can simultaneously account for long-term user preferences (captured by LSTM) and short-term interests (highlighted by attention), integrating movie feature information to yield more accurate and personalized recommendations.

The LSTM layer is crucial for capturing long-term dependencies in users' viewing histories, which is vital for understanding overall preference trends. The attention mechanism allows the model to assign different importance weights to various historical behaviors, aiding in emphasizing the most relevant recent actions, thus better capturing dynamic shifts in user interests.

The fusion layer creates a comprehensive representation by combining the LSTM output, the attention-weighted context vector, and the original features, incorporating sequential information, importance-weighted data, and raw features. This multi-faceted information integration aids the model in making more nuanced and accurate predictions.

The fully connected layer processes this integrated information through nonlinear transformations to generate the final score predictions. The loss function design not only considers prediction errors but also incorporates a regularization term to prevent overfitting, enhancing the model's generalization capabilities.

By employing this improved attention mechanism-enhanced LSTM model, the system is better equipped to capture the dynamic changes in user interests, paying appropriate attention to recent behaviors while considering long-term preferences, thus providing more accurate and personalized movie recommendations. This design not only boosts recommendation accuracy but also enhances model interpretability, as the analysis of attention weights can reveal the critical historical behaviors the model focuses on when making recommendations.

3.4.2 Integration of Knowledge Graph and Attention-Based LSTM Model. To enhance model performance, this study innovatively combines a knowledge graph with an attention-based LSTM model. This integration aims to leverage structured knowledge and user behavior data for more accurate and rich movie recommendations. The integration process includes three key steps: knowledge graph embedding, LSTM sequence modeling, and the fusion method.

First, during the knowledge graph embedding phase, graph embedding algorithms like TransE map entities such as movies, actors, and directors into a low-dimensional vector space. For each movie, the embedding vectors of relevant entities (e.g., director, leading actors, genre) are extracted and combined using weighted averaging or attention mechanisms to create a comprehensive knowledge representation. This approach captures semantic relationships and structured information among movies, providing rich contextual knowledge for the recommendation system[30].

Next, in the LSTM sequence modeling phase, the user's viewing history is treated as a time series, where each time step contains movie ID, viewing time, and rating information. The LSTM network processes these sequences to output dynamic user interest representations, effectively capturing the temporal evolution of user preferences.

Finally, in the fusion method phase, the user interest representation outputted by the LSTM is integrated with the knowledge graph embedding of the movie to be predicted. This fusion employs an attention mechanism that dynamically adjusts the importance of the two representations, allowing the model to flexibly balance the significance of historical user behavior and movie characteristics across different scenarios.

The fusion formula is as follows:

$$f_{\text{fused}} = \text{Attention}(f_{\text{kg}}, f_{\text{lstm}}) \quad (3.17)$$

Where f_{kg} represents the knowledge graph embedding features and f_{lstm} represents the user interest features output by the LSTM. The advantages of this fusion method lie in its ability to simultaneously utilize static knowledge graph information and dynamic user behavior data. Knowledge graph embeddings provide semantic correlations and structured information among movies, addressing cold-start issues and enhancing recommendation diversity. LSTM sequence modeling captures the temporal dynamics of user interests, adapting to changes in user preferences. Through attention mechanism fusion, the model can dynamically adjust the importance of these information sources according to specific situations, yielding more accurate recommendations.

Overall, the integration of knowledge graphs with attention-based LSTM models represents a significant direction in recommendation system research, demonstrating how to effectively combine diverse information sources to improve recommendation performance. This approach is applicable not only to movie recommendations but can also extend to personalized recommendation tasks in other domains.

3.4.3 Benchmark Models.

To comprehensively evaluate the performance of the proposed attention-based LSTM model, various typical models are chosen as benchmarks. These benchmark models include the standard LSTM model, SVD-based matrix factorization methods, Neural Collaborative Filtering (NCF) models, and Deep Feedforward Neural Networks (FFNN). Each model represents significant methodologies in the field of recommendation systems, and comparing these models provides insights into their strengths and limitations in movie recommendation tasks.

The standard LSTM model is a powerful tool for sequence modeling, effectively capturing long-term dependencies essential for modeling the long-term variations in user interests. Its memory cells and gating mechanisms enable it to learn and remember critical information across long sequences while selectively forgetting irrelevant data. However, the standard LSTM may face information loss issues when handling very long sequences and struggle to distinguish the importance of different historical behaviors. By comparing the non-attention LSTM model with the proposed attention-based LSTM, this study can directly assess the contribution of the attention mechanism in movie recommendation tasks and understand its impact on model sensitivity to changes in user interests and overall performance.

Matrix factorization is a classic method in recommendation systems, especially the SVD-based matrix factorization techniques that excel in collaborative filtering tasks. In this study, SVD-based matrix factorization is employed as one of the benchmark methods. The core idea is to decompose the user-item rating matrix R into the product of two low-dimensional matrices:

$$R \approx U \Sigma V^T \quad (3.18)$$

Where U is the user feature matrix, V is the item feature matrix, and Σ is the singular value matrix. In recommendation systems, variants of SVD, such as FunkSVD or BiasedSVD, are often used, which account for user and item bias terms. The implementation in this study is based on the SVD algorithm in the Surprise library, mathematically expressed as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (3.19)$$

Where $\hat{r}_{ui}$ is the predicted rating of item i by user u, μ is the global average rating, b_u is the bias term for user u, b_i is the bias term for item i, q_i is the latent feature vector for item i, and p_u is the latent feature vector for user u. The objective function for the model is:

$$\min \sum (r_{ui} - \hat{r}_{ui})^2 + \lambda(||b_u||^2 + ||b_i||^2 + ||q_i||^2 + ||p_u||^2) \quad (3.20)$$

Where λ is a regularization parameter to prevent overfitting. In this implementation, 100 latent factors (n_factors=100) are set, with 20 training epochs (n_epochs=20), a learning rate of 0.005 (lr_all=0.005), and a regularization parameter of 0.02 (reg_all=0.02). These hyperparameters are selected based on extensive experiments.

Neural Collaborative Filtering (NCF) applies deep learning techniques to collaborative filtering. The core idea of the NCF model is to use neural networks to learn non-linear interactions between users and items. The NCF model structure includes an input layer (user ID and movie ID), an embedding layer (converting user and movie IDs into low-dimensional dense vectors), a multilayer perceptron (MLP) to learn complex interactions between user and movie embeddings, and an output layer (predicting user ratings for movies). The mathematical representation of the NCF model is: 1. Embedding layer:

$$V_u = E_u(u), \quad V_m = E_m(m) \quad (3.21)$$

Where E_u and E_m are the embedding matrices for users and movies, respectively. 2. Feature vector concatenation:

$$z = [V_u; V_m] \quad (3.22)$$

3. MLP layer:

$$a_1 = \mathrm{ReLU}(W_1 z + b_1), \quad a_2 = \mathrm{ReLU}(W_2 a_1 + b_2), \quad \ldots, \quad a_L = \mathrm{ReLU}(W_L a_{L-1} + b_L) \quad (3.23)$$

4. Output layer:

$$\hat{y} = \sigma(h^T a_L) \quad (3.24)$$

Where σ is the sigmoid activation function, and h is the weight vector of the output layer.

In this implementation, 50-dimensional embedding vectors are used, with two hidden layers (128 and 64 neurons), ReLU activation functions, and the Adam optimizer for training. The NCF model's strength lies in its ability to learn complex non-linear user-item interactions and its scalability to include additional features.

Deep Feedforward Neural Networks (FFNN) are powerful machine learning models capable of learning complex non-linear relationships. In movie recommendation systems, FFNN can effectively handle various input features, including user characteristics, movie features, and contextual information, to provide accurate rating predictions. The structure of FFNN includes an input layer (receiving all relevant features), hidden layers (multiple fully connected layers with non-linear activation functions), and an output layer (one neuron for predicting user ratings). The mathematical representation is as follows: - Input layer: Receives all relevant features. - First hidden layer:

$$h_1 = \mathrm{ReLU}(W_1 x + b_1) \quad (3.25)$$

- Second hidden layer:

$$h_2 = \mathrm{ReLU}(W_2 h_1 + b_2) \quad (3.26)$$

- L-th hidden layer:

$$h_L = \mathrm{ReLU}(W_L h_{L-1} + b_L) \quad (3.27)$$

Where W_i and b_i are the weight matrix and bias vector of the i-th layer, and ReLU is the rectified linear unit activation function. The output layer predicts user ratings as:

$$y = W_{\text{out}} h_L + b_{\text{out}} \quad (3.28)$$

In this study, a FFNN model with three hidden layers (128, 64, and 32 neurons) is implemented. The model uses the Adam optimizer for training, with mean squared error (MSE) as the loss function. Additionally, dropout layers are included to prevent overfitting and improve the model's generalization capability. The strengths of FFNN include flexibility in feature handling, non-linear modeling capability, scalability, and end-to-end learning. However, FFNN has limitations, such as difficulty in directly handling sequential data and capturing long-term dependencies, which are strengths of LSTM and attention mechanisms. By comparing with the FFNN model, the advantages of the attention-based LSTM model in handling temporal information and long-term dependencies can be assessed, as well as the impact of complex non-linear modeling on recommendation system performance. By implementing these different types of models, a comprehensive comparison of their performance in movie recommendation tasks can be conducted. Each method has unique strengths and limitations, and comparative analysis provides insights into the capabilities of different models in handling user-item interactions, feature learning, and non-linear modeling, particularly the impact of attention mechanisms on sequence model performance. This systematic comparison aids in evaluating the advantages of the proposed attention-based LSTM model while offering valuable insights for future model improvements and feature engineering. Through this comprehensive evaluation, the performance of each model can be assessed in various aspects, such as predictive accuracy, computational efficiency, interpretability, and the ability to handle long-term dependencies.

3.5 Chapter Summary

This chapter provides a detailed introduction to the proposed attention-based LSTM model, which effectively leverages user historical behavior sequences and movie feature information. It also explores the innovative method of integrating knowledge graphs with LSTM models, alongside several benchmark models for comparison. Through knowledge graph embeddings, the model can effectively utilize rich structured knowledge in the movie domain, while the LSTM network captures the temporal patterns of user behavior. The introduction of a multi-layer attention mechanism further enhances the model's expressiveness and interpretability.

The primary advantages of this integration method include: first, the effective use of domain knowledge, improving recommendation accuracy and diversity; second, capturing the dynamic changes in user interests for personalized recommendations; third, enhancing model interpretability through attention mechanisms; and finally, flexibly addressing cold-start problems, providing reasonable recommendations for new users and movies. The next chapter will focus on experimental evaluation and comparative analysis of these models.

4 Empirical Research and Results Analysis

This chapter provides a detailed overview of the empirical research process and results analysis for the attention-based LSTM model in the movie recommendation system. First, the chapter describes the experimental design, including the experimental environment, data processing methods, and experimental steps. Next, it presents the evaluation methods and metrics, as well as the implementation and optimization strategies for the main model. Subsequently, a comprehensive assessment of the proposed model's performance is conducted through comparisons with multiple benchmark models. Finally, the chapter explores the model's interpretability, utilizing the Neo4j graph database to visualize recommendation results and analyze the recommendation logic in depth. This series of experiments and analyses aims to validate the effectiveness and superiority of the attention-based LSTM model in personalized movie recommendation tasks.

4.1 Experimental Design

Category	Component	Details
Hardware Environment	CPU	Intel Core i7-10700K @ 3.80GHz
	RAM	32GB DDR4
	GPU	NVIDIA GeForce RTX 3080 (10GB VRAM)
Software Environment	Operating System	Linux 6.1.85+
	Python Version	3.10.12
	CUDA Version	12.1
Major Dependencies	Data Processing	pandas (2.0.3), numpy (1.25.2)
	Machine Learning	scikit-learn (1.2.2)
	Deep Learning	TensorFlow (2.15.0), Keras (2.15.0)
	Collaborative Filtering	Surprise (1.1.4)
	Visualization	matplotlib (3.7.1), seaborn (0.13.1)
Development Tools	IDE	IntelliJ IDEA 2022.1.1
	Version Control	Git 2.39.2
	Database	Neo4j 5.13.0
Computing Resources	Cloud Platform	Google Colab

Table 2. Experimental Environment and Tools

4.1.1 Experimental Environment and Tools. The selection of these tools and environments ensures not only the efficient execution of experiments and the reliability of results but also provides necessary support for handling large-scale data and complex models. Utilizing GPU acceleration for training deep learning models significantly improves experimental efficiency, while the Neo4j graph database helps better leverage the rich relational information within movie data, thereby providing a more comprehensive feature set for the recommendation system.

4.2 Data Processing and Feature Engineering

The data preprocessing stage consists of three main steps: first, using the 'data_preprocessing.py' script for initial data cleaning and integration, removing obvious outliers and duplicates. Next, a more in-depth cleaning is conducted with the 'Data_cleaning.py' script to ensure data consistency and completeness. Finally, the 'Data_magnitude_filter.py' script is applied to filter for high-quality movies and significant actors/directors, thereby improving the signal-to-noise ratio of the data.

Feature engineering is a critical component of this research, where a multidimensional feature set is constructed. User features include the number of ratings, average rating, and rating standard deviation, reflecting user activity and rating tendencies. Movie features encompass attributes such as year, genre, main actors, directors, and total rating counts, capturing the basic properties and popularity of the movies. Temporal features consider aspects like rating year, month, quarter, and day of the week to capture potential temporal patterns and seasonal trends. Interaction features are designed to represent user-movie rating differences and average ratings for specific genres, illustrating unique interactions between users and movies.

Data division employs a time series splitting method, aligning with the practical application scenarios of recommendation systems and aiding in the assessment of the model's ability to capture long-term changes in user preferences. Specifically, the dataset is divided into a training set (80%) and a test set (20%) based on the chronological order of ratings. Additionally, 20% of the training set is reserved as a validation set for model tuning and early stopping. Care is taken to ensure that each user and movie has sufficient samples in both the training and test sets, thereby alleviating cold-start issues.

This data processing and division approach ensures the quality and representativeness of the experimental data while simulating the time series prediction scenarios found in actual recommendation systems. This is particularly important for evaluating the proposed attention-based LSTM model's capacity to capture changes in user long-term preferences.

4.2.1 Experimental Methods and Steps. This experiment employs a systematic approach to design and implement a personalized movie recommendation system. The experimental workflow encompasses the entire process from data processing to results analysis, as illustrated in Figure 4.1.

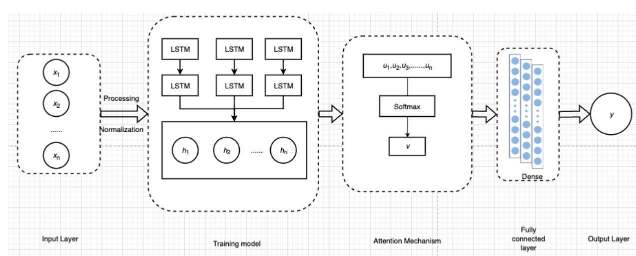

Figure 12. Experimental Workflow Diagram

Through this series of systematic steps, the aim is to comprehensively evaluate the performance of the attention-based LSTM model in movie recommendation tasks. By comparing it with various benchmark models, the advantages of the attention mechanism in capturing long-term dependencies in user-movie interactions are further understood.

4.3 Evaluation Methods and Metrics

This section provides a detailed overview of the evaluation metrics, model training strategies, and related parameter settings used in the experiments, laying the foundation for subsequent results analysis.

4.3.1 Evaluation Metrics. To thoroughly assess the performance of the recommendation system, this study selects the following evaluation metrics:

1. Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum(y_{\text{true}} - y_{\text{pred}})^2 \quad (4.1)$$

MSE measures the average squared difference between predicted and true values, being more sensitive to larger errors.

2. Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum|y_{\text{true}} - y_{\text{pred}}| \quad (4.2)$$

MAE measures the average absolute difference between predicted and true values, exhibiting lower sensitivity to outliers than MSE.

3. R-squared (R^2):

$$R^2 = 1 - \frac{\sum(y_{\text{true}} - y_{\text{pred}})^2}{\sum(y_{\text{true}} - y_{\text{mean}})^2} \quad (4.3)$$

R^2 indicates the proportion of variance in the dependent variable that is explained by the model, with a range of [0, 1]. A value closer to 1 indicates better model fit.

4. Explained Variance Score (EVS):

$$\mathrm{EVS} = 1 - \frac{\mathrm{Var}(y_{\text{true}} - y_{\text{pred}})}{\mathrm{Var}(y_{\text{true}})} \quad (4.4)$$

EVS measures the extent to which the model captures the variance of the target variable, with a best value of 1.

5. Median Absolute Error (MedAE):

$$\mathrm{MedAE} = \mathrm{median}(|y_{1,\text{true}} - y_{1,\text{pred}}|, \ldots, |y_{n,\text{true}} - y_{n,\text{pred}}|) \quad (4.5)$$

MedAE is less sensitive to outliers and is suitable for assessing the stability of prediction results.

The selection of these metrics is based on the following considerations: MSE and MAE provide intuitive reflections of prediction error magnitude, R^2 and EVS assess overall model fit, while MedAE offers an outlier-insensitive error evaluation. Utilizing these metrics comprehensively assesses the model's performance across various aspects, providing a basis for model selection and optimization.

4.3.2 Hyperparameter Settings. The choice of hyperparameters significantly impacts model performance. This study adopts a combined method of grid search and random search for hyperparameter optimization.

The hyperparameter optimization process employs 5-fold cross-validation, using MSE as the primary optimization objective. The final selected hyperparameter combination achieves the best performance on the validation set.

Model	Key Hyperparameters
Attention-based LSTM	LSTM Units: 64 Embedding Dimension: 32 Dropout Rate: 0.5 L2 Regularization Coefficient: 1×10^{-6}
LSTM (without Attention)	LSTM Units: 32 Embedding Dimension: 32
SVD Matrix Factorization	Number of Latent Factors: 100 Learning Rate: 0.005 Regularization Coefficient: 0.02
Neural Collaborative Filtering (NCF)	Embedding Dimension: 50 Hidden Layer Units: [128, 64]
Deep Feedforward Neural Network	Hidden Layer Structure: [128, 64, 32] Dropout Rate: 0.3

Table 3. Main Hyperparameters and Their Search Ranges

4.3.3 Loss Function Selection. Considering the characteristics of the movie rating prediction task, this study primarily adopts Mean Squared Error (MSE) as the loss function. The choice of MSE is based on its intuitiveness, differentiability, and ability to penalize large errors. For certain models, such as autoencoders, the Mean Absolute Error (MAE) loss function is also tested to compare the impact of different loss functions on model performance.

4.3.4 Optimizer Configuration. This study mainly utilizes the Adam (Adaptive Moment Estimation) optimizer, an adaptive learning rate algorithm that combines the advantages of momentum and RMSprop. The selection of the Adam optimizer is based on its adaptive learning rate, computational efficiency, and robustness to hyperparameters. For the SVD matrix factorization model, the Stochastic Gradient Descent (SGD) optimizer is employed to maintain consistency with traditional collaborative filtering methods.

4.3.5 Learning Rate Adjustment Strategy. To improve training effectiveness and convergence speed, a dynamic learning rate adjustment strategy is implemented using the ReduceLROnPlateau callback function. This strategy monitors validation loss and reduces the learning rate to 20% of its original value if there's no improvement over five consecutive epochs, with a minimum learning rate set to 0.00001. This approach allows for a larger learning rate during the initial training phase for rapid convergence, followed by automatic reduction in later stages to fine-tune model parameters, effectively enhancing performance and training efficiency.

4.3.6 Early Stopping Strategy. To prevent overfitting and improve training efficiency, an early stopping strategy is employed through Keras's EarlyStopping callback function. This strategy monitors validation loss with a patience value set to 10 epochs, enabling the restoration of the best weights. If validation loss does not improve over 10 epochs, training stops automatically, restoring the model to its best-performing state. This method effectively avoids overfitting while ensuring optimal model performance.

In summary, this section details the rationale behind the choice of evaluation metrics, hyperparameter settings, loss function selection, optimizer configuration, learning rate adjustment strategies, and early stopping strategies. These carefully designed settings ensure efficient and effective model training, laying the groundwork for a high-quality recommendation system. In the subsequent model comparisons, this study will analyze the differences in these metrics across various models, providing a comprehensive assessment of their strengths and weaknesses. For instance, the performance of the attention-based LSTM model will be compared with other models in terms of MSE, MAE, R^2, EVS, and MedAE, aiming to identify the model with the best accuracy, interpretability, and robustness.

4.4 Main Model Implementation and Optimization

This section introduces the implementation and optimization of the attention-based LSTM model. This model combines the sequence modeling capabilities of LSTM with the dynamic feature weighting ability of the attention mechanism, aiming to enhance the accuracy and personalization of movie recommendations.

The core structure of the model includes an input layer, embedding layer, LSTM layer, attention layer, fully connected layer, and output layer. To improve performance, optimization strategies such as regularization, learning rate scheduling, early stopping, and gradient clipping are employed.

Next, we will explore the implementation of the attention mechanism and how these optimization strategies enhance model performance, demonstrating the model's effectiveness in capturing the dynamic changes in user interests.

4.4.1 Specific Implementation of the Attention Mechanism. In this study, a linear transformation is applied to the output of each time step of the LSTM, followed by normalization using the softmax function to obtain attention weights. These weights are then used to perform a weighted sum of the LSTM outputs, generating a context vector. Finally, the context vector is concatenated with other features to serve as input to the subsequent fully connected layer. This implementation allows the model to dynamically focus on important parts of the user's historical behavior, effectively capturing both long-term and short-term interests.

Through the aforementioned model implementation and optimization strategies, a recommendation system capable of capturing dynamic changes in user interests has been constructed.

4.5 Performance Evaluation

In this section, we will evaluate the performance of the main model against four benchmark models in the movie recommendation task: traditional collaborative filtering methods (e.g., SVD matrix factorization), deep learning models (e.g., standard LSTM, Neural Collaborative Filtering), and deep feedforward neural networks. These models represent different paradigms of recommendation systems, from traditional methods to deep learning approaches. By comparing their performances, we can gain a comprehensive understanding of the advantages and disadvantages of various methods in the movie recommendation task.

4.5.1 Performance Analysis of the Attention-Based LSTM Model. The proposed attention-based LSTM model demonstrates excellent performance in the movie rating prediction task. The following metrics reflect the model's predictive accuracy and stability from various perspectives:

Mean Squared Error (MSE): 0.59, indicating a low average squared difference between predicted and actual values, suggesting close predictions with minimal error. Mean Absolute Error (MAE): 0.51, representing an average absolute difference of approximately half a point on a 10-point rating scale, which is a commendable outcome. R-squared (R^2): 0.91, indicating that the model explains 91% of the variance in ratings, reflecting strong predictive capability. Explained Variance Score (EVS): 0.92, showing the model's effectiveness in capturing data variance. Median Absolute Error (MedAE): 0.33, providing a median error that is insensitive to outliers, confirming the model's stability and accuracy.

In conclusion, the attention-based LSTM model exhibits outstanding performance and stability in the movie rating prediction task. Low error metrics and high explained variance suggest the model can effectively capture the complex patterns in user rating behaviors.

4.5.2 Comparison Analysis with Benchmark Models. To comprehensively evaluate the performance of the proposed attention-based LSTM model, we compare its results against various benchmark models using the same dataset.

Model	MSE	MAE	R^2	EVS	MedAE
Attention-based LSTM	0.59	0.51	0.91	0.92	0.33
Standard LSTM (without Attention)	0.92	0.65	0.86	0.86	0.43
SVD Matrix Factorization	4.95	1.74	0.26	0.26	1.42
Neural Collaborative Filtering (NCF)	5.96	1.84	0.11	0.11	1.39
Deep Feedforward Neural Network	1.07	0.87	0.84	0.88	0.87

Table 4. Performance Metrics Comparison of Models

The analysis indicates that the attention-based LSTM model outperforms most benchmarks across various metrics. Specifically, it reduces MSE by 35.9% compared to the standard LSTM and improves R^2 by 5.8%. This enhancement highlights the attention mechanism's effectiveness in capturing long-term dependencies in user-movie interactions.

When compared to the traditional SVD matrix factorization, the attention-based LSTM model shows remarkable superiority, with an MSE reduction of 88.1% and a 250% increase in R^2. This disparity underscores the deep learning model's ability to handle complex nonlinear relationships effectively.

Similarly, the attention-based LSTM model significantly outperforms the NCF model, with MSE dropping by 90.1% and R^2 improving by 727.3%. This indicates the advantages of sequence modeling and attention mechanisms in understanding the temporal dynamics of user ratings.

Even against the deep feedforward neural network, the attention-based LSTM model shows superiority, with MSE reducing by 44.9% and R^2 improving by 8.3%. This suggests that sequence models are more adept at handling temporal data than static feedforward networks.

In summary, the attention-based LSTM model consistently outperforms other benchmarks, especially traditional methods and simpler deep learning models. The integration of attention mechanisms clearly enhances LSTM performance, validating its capacity for capturing long-term dependencies. The analysis illustrates the potential of deep learning models in recommendation systems.

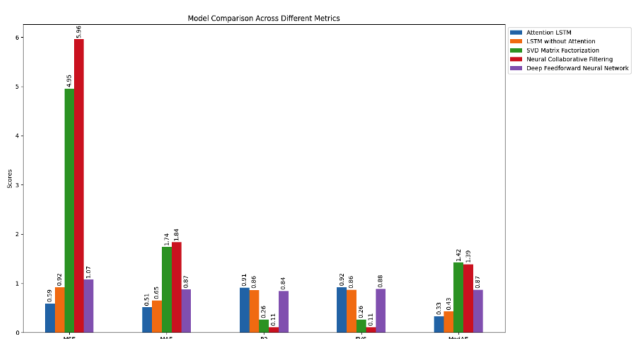

Figure 13. Bar chart comparing model performance, highlighting MSE and R^2 metrics

The bar chart reinforces that the attention-based LSTM model excels in most metrics, particularly in MSE and R^2. This confirms the attention mechanism's strength in capturing long-term dependencies in user-movie interactions.

This comparative analysis not only validates the effectiveness of the proposed model but also suggests future directions for improvement, such as exploring the integration of attention mechanisms with other deep learning techniques to further enhance performance. It emphasizes the importance of model selection in practical applications, requiring careful consideration of specific issues and data characteristics. Future research may delve into more complex attention mechanisms or combine the attention-based LSTM model with advanced technologies like graph neural networks or reinforcement learning to boost recommendation system performance.

4.6 Model Interpretability Analysis

This section explores the interpretability of the attention-based LSTM model and how the Neo4j graph database can be used to visualize and understand recommendation results. To gain insights into the model's recommendation logic, a user-movie-feature relationship network is constructed using Neo4j.

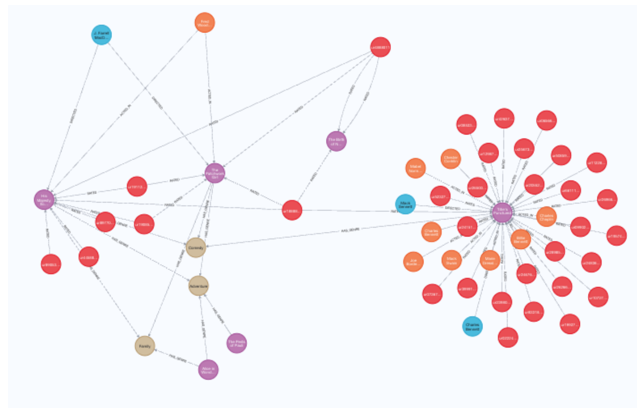

Figure 14. Neo4j relationship graph depicting the user-movie-feature network

By analyzing the recommendation paths in the graph, this study identifies that the recommendation logic is primarily based on several factors: genre similarity, director associations, and actor bridges. The model tends to recommend new films similar to genres the user has previously liked while also considering other works by directors the user has watched. Additionally, the recommendation paths can establish connections between different genres through co-starring actors. However, the quality of these paths relies on thorough data cleaning and diverse dimensional attributes. The visualization of these paths provides interpretable evidence for the recommendation results.

4.7 Summary of the Chapter

This chapter provides a detailed account of the experimental design, evaluation methods, model implementation, and performance assessment. Through systematic experimentation and multi-dimensional evaluation metrics, the performance of the attention-based LSTM model is comprehensively compared with various benchmark models. The results indicate that the proposed model exhibits excellent performance in movie rating prediction tasks, surpassing traditional methods and other deep learning models in accuracy while also offering good interpretability. The interpretability analysis further reveals how the attention mechanism effectively captures dynamic changes in user interests, providing a reliable foundation for personalized recommendations.

5 Conclusion and Future Directions

5.1 Conclusion

This study presents a unified solution that explains how knowledge graphs, LSTM, and attention mechanisms complement and collaborate mathematically. Specifically, knowledge graph embeddings are used as the initial state and additional input for LSTM, while the attention mechanism is employed to dynamically adjust the importance of different time steps and types of information. This framework overcomes the limitations of traditional recommendation systems in modeling the complex relationships between users, movies, and context. Compared to using knowledge graphs alone, the combined model better captures the temporal dynamics of user interests. Furthermore, the introduction of knowledge graphs and attention mechanisms enhances the model's expressive power and interpretability relative to a pure LSTM model. This combination addresses the limitations of individual methods, achieving more comprehensive and precise recommendations.

In practical applications, this combined approach not only improves recommendation accuracy but also enhances personalization and interpretability. The inclusion of knowledge graphs enables the system to provide explanations based on movie features and relationships, while the attention mechanism helps elucidate the model's focal points, significantly increasing user understanding and trust in the recommendations. This combined approach has broad application potential, extending beyond movie recommendations to domains such as music and literature. Future research directions include incorporating more complex graph neural networks to better leverage knowledge graph information and exploring the integration of reinforcement learning techniques into the existing framework to achieve more dynamic recommendation strategies.

5.2 Research Limitations

Despite achieving significant results in personalized movie recommendation systems, this study acknowledges certain limitations that provide important guidance for future research directions. First, considering the scale and complexity of actual commercial recommendation systems, there is still room for expansion regarding the size of the dataset used in this research. Therefore, designing a series of experiments targeting datasets of varying scales to comprehensively assess the model's generalization capabilities would help deepen the understanding of the model's strengths and potential improvement areas.

Secondly, while this research considers various features for recommendations, it has not fully utilized rich content features such as movie text descriptions and visual information. These multimodal data could provide additional valuable insights for enhancing the accuracy, diversity, and personalization of the recommendation system. Hence, exploring how to effectively integrate these rich content features into the recommendation model will be a crucial future research direction.

By recognizing and addressing these limitations, future research aims to further enhance the model's performance, efficiency, and applicability, better meeting the demands of large-scale commercial recommendation systems and providing users with more precise and diverse personalized movie recommendations.

5.3 Future Research Directions

Based on the findings and limitations of this research, several promising future research directions are proposed. First, plans are underway to explore the incorporation of content features such as movie text descriptions and actor information into the model to enhance recommendation diversity and interpretability. This may involve using natural language processing techniques to handle text data or employing graph neural networks to better capture relationships between actors and directors.

Additionally, considering the construction of a multi-task learning framework that combines rating prediction with other related tasks (such as viewing time prediction) could help the model learn richer feature representations and improve overall performance.

Moreover, further optimization of the model is planned to better capture the rapid changes in user interests, such as introducing time decay factors or utilizing more complex time series models (like Transformers). This is particularly important for accurately capturing short-term shifts in user interests. This research also aims to explore the potential application of the current model in other domains (such as music and book recommendations) and investigate methods for knowledge transfer across domains, which may involve the application of transfer learning techniques or domain adaptation methods.

Finally, given the demands of large-scale applications, this study will investigate how to improve computational efficiency while maintaining model performance, possibly including techniques such as model compression and distributed computing. Through these improvements and expansions, it is anticipated that the performance and applicability of the recommendation system will be further enhanced, providing users with a more precise and personalized movie recommendation experience.

References

[1] L Antony Rosewelt and J Arokia Renjit. 2020. A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN. *Journal of Intelligent & Fuzzy Systems* 39, 1 (2020), 795–808.

[2] Vahid Baghi. 2020. IMDb Users' Ratings Dataset. https://doi.org/10.21227/br41-bd49

[3] He Changlong and Wen Bin. 2024. Recommendation Model Combining Gated Attention Mechanism and Bilinear Feature Interaction. *Computer Applications and Software* (2024).

[4] Wu Di, Yang Lijun, and Ma Wenli. 2024. Time-Aware User Long- and Short-Term Interest Feature Separation Recommendation Algorithm. *Computer Engineering and Design* (2024).

[5] MP Geetha and D Karthika Renuka. 2019. Research on recommendation systems using deep learning models. *International Journal of Recent Technology and Engineering (IJRTE)* 8, 4 (2019).

[6] Gao Guangshang. 2024. A Review of Neural Networks Combined with Attention Mechanisms in Recommender Systems. *Computer Engineering and Applications* (2024).

[7] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3549–3568.

[8] Wang Haofen, Ding Jun, Hu Fanghuai, and Wang Xin. 2020. Review of Large-Scale Enterprise-Level Knowledge Graph Practices. *Computer Engineering* (2020).

[9] Zhou Jiaqi, Song Ran, and Yu Zhengtai. 2024. Low-Resource Knowledge Graph Completion Based on Joint Learning of Rules and Embeddings. *Computer Engineering and Applications* (2024), 1–13.

[10] Hu Jimi, Wan Weibing, and Cheng Feng. 2024. Research on Rule Extraction and Reasoning Integrating Relation and Structure Encoding. *Journal of East China Normal University (Natural Science Edition)* (2024), 1–13.

[11] Tian Jishuai and Ai Fangju. 2024. Aspect-Level Sentiment Analysis Based on Enhanced Syntactic Information and Multi-Feature Graph Convolution Fusion. *Journal of Computer Science and Exploration* (2024), 1–15.

[12] Zhang Jixiang, Zhang Xiangsen, Wu Changxu, and Zhao Zengshun. 2022. A Review of Knowledge Graph Construction Techniques. *Computer Engineering* (2022).

[13] Wang Lianhong, Lin Feipeng, and Li Xiaoyao. 2024. KMAKT Prediction Integrated with Course Knowledge Graph. *Computer Engineering* (2024).

[14] Dong Manru. 2022. Research on CPI Prediction Based on Machine Learning Theory. *University of International Business and Economics* (2022).

[15] Haithem Mezni, Djamal Benslimane, and Ladjel Bellatreche. 2021. Context-aware service recommendation based on knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5225–5238.

[16] Tribikram Pradhan, Prashant Kumar, and Sukomal Pal. 2021. CLAVER: An integrated framework of convolutional layer, bidirectional LSTM with attention mechanism based scholarly venue recommendation. *Information Sciences* 559 (2021), 212–235.

[17] Saeed Mian Qaisar. 2020. Sentiment analysis of IMDb movie reviews using long short-term memory. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*. IEEE, 1–4.

[18] Chen Qiang, Zhang Dong, Li Shoushan, and Zhou Guodong. 2024. Multimodal Knowledge Graph Completion Integrating Task Knowledge. *Journal of Software* (2024), 1–15.

[19] Deepjyoti Roy and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data* 9, 1 (2022), 59.

[20] Benkessirat Selma, Boustia Narhimène, and Rezoug Nachida. 2021. Deep learning for recommender systems: Literature review and perspectives. In *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*. IEEE, 1–7.

[21] Li Shuxian, Zhang Xiaojun, and Hu Chengyu. 2023. LSTM-Based Policy Effect Prediction Model and Its Application. *Statistics and Decision* (2023).

[22] Harald Steck, Linas Baltrunas, Ehtsham Elahi, Dawen Liang, Yves Raimond, and Justin Basilico. 2021. Deep learning for recommender systems: A Netflix case study. *AI Magazine* 42, 3 (2021), 7–18.

[23] Yuezhong Wu, Qiang Liu, Rongrong Chen, Changyun Li, and Ziran Peng. 2020. A group recommendation system of network document resource based on knowledge graph and LSTM in edge computing. *Security and Communication Networks* 2020, 1 (2020), 8843803.

[24] Tang Xiaobin, Dong Manru, and Zhang Rui. 2020. Research on Consumer Confidence Index Prediction Based on Machine Learning LSTM&US Model. *Statistical Research* (2020).

[25] Chen Xusong. 2021. Research on Recommendation Algorithms Based on User Behavior Sequence Modeling. *University of Science and Technology of China* (2021).

[26] Liu Yannan, Liu Shuang, and Zhang Xuejing. 2015. Comparison of Paid Video Websites in China and the US: Users, Content, and Models. *Journal of China University of Geosciences (Social Science Edition)* (2015).

[27] Zhou Yaowei, Kong Lingjun, and Dai Qi. 2024. Lightweight Parcel Damage Detection Algorithm Based on Multi-Head Self-Attention Mechanism. *Radio Communications Technology* (2024), 1–11.

[28] Yao Yi, Chen Chaoyang, and Du Xiaoming. 2024. A Review of Multimodal Knowledge Graph Construction Technology and Its Applications in the Military Field. *Computer Engineering and Applications* (2024), 1–22.

[29] Qiu Yunfei and Tian Fengwei. 2024. Recommendation Method Based on Hybrid Interest Topic Model. *Journal of Liaoning Technical University (Natural Science Edition)* (2024).

[30] Fengsheng Zeng and Qin Wang. 2022. Intelligent recommendation algorithm combining RNN and knowledge graph. *Journal of Applied Mathematics* 2022, 1 (2022), 7323560.

[31] Pengyu Zhao, Xin Gao, Chunxu Xu, and Liang Chen. 2023. M5: Multi-Modal Multi-Interest Multi-Scenario Matching for Over-the-Top Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5650–5659.