A novel recommender system for adapting single machine problems to distributed systems within MapReduce

Kamila Orynbekova[1], Shirali Kadyrov[2], Andrey Bogdanchikov[1], Saidakmal Oktamov[1]
[1]Department of Computer Sciences, Faculty of Engineering and Natural Sciences, Suleyman Demirel University (SDU), Almaty, Kazakhstan
[2]Department of General Education, New Uzbekistan University, Tashkent, Uzbekistan

Article Info

Article history:

Received Feb 21, 2024
Revised Aug 22, 2024
Accepted Sep 28, 2024

Keywords:

Distributed system
Knowledge-based approach
Machine learning model
MapReduce
Recommender system
Text-based approach

ABSTRACT

This research introduces a novel recommender system for adapting single-machine problems to distributed systems within the MapReduce (MR) framework, integrating knowledge and text-based approaches. Categorizing common problems by five MR categories, the study develops and tests a tutorial with promising results. Expanding the dataset, machine learning models recommend solutions for distributed systems. Results demonstrate the logistic regression model's effectiveness, with a hybrid approach showing adaptability. The study contributes to advancing the adaptation of single-machine problems to distributed systems MR, presenting a novel framework for tailored recommendations, thereby enhancing scalability and efficiency in data processing workflows. Additionally, it fosters innovation in distributed computing paradigms.

Corresponding Author:

Kamila Orynbekova
Department of Computer Sciences, Faculty of Engineering and Natural Sciences
Suleyman Demirel University (SDU)
Abylai khan street 1/1, Kaskelen, Almaty, Kazakhstan
Email: kamila.orynbekova@sdu.edu.kz

1. INTRODUCTION

The emergence of big data has led to the development of efficient processing techniques, with MapReduce (MR) and Hadoop being at the forefront [1]. Unlike traditional single-machine solutions, MR enables parallel processing across clusters of machines, offering scalability and fault tolerance. This shift towards distributed computing has revolutionized the field, allowing organizations to tackle complex data processing tasks with unprecedented speed and efficiency.

True to their word, the shift from single-machine paradigms to distributed systems such as MR can be a steep learning curve for many. Conventional modes of learning MR typically involve extensive tutorials and reading extensive documentation, which can take time and is difficult to understand fully. Thus, prospective learners are tasked with understanding difficult concepts and difficult implementations, which poses a challenge to their level of competence in distributed computing.

Recognizing the need for a more intuitive and efficient approach to learning MR, a recent study [2] introduces a novel project-based learning approach to learn MR solutions. Statistical analysis demonstrates that the proposed methodology has significant improvement over traditional approaches. By allowing learners to engage directly with practical projects akin to those they anticipate encountering in real-world scenarios, this approach offers a tangible and contextualized learning experience. However, a critical question arises: how can learners identify the most suitable tutorial to embark on their MR journey?

Text-based recommender systems for problem-solving, powered by natural language processing (NLP), have emerged as crucial tools in guiding users to relevant solutions [3]–[7]. Leveraging techniques such as semantic analysis and keyword extraction, these systems analyze textual queries and descriptions to understand user needs and provide tailored recommendations. They sift through vast repositories of textual data, offering targeted advice or solutions, thereby streamlining problem-solving processes [8]. In the context of problem-solution repositories, these systems can generate intelligent query suggestions, improving upon traditional mechanisms [9]–[13]. Knowledge-based recommender systems, which leverage explicit domain knowledge, have shown promise in addressing challenges such as data sparsity and the "early rater" problem [14], [15]. These systems leverage explicit domain knowledge to provide more precise recommendations [16], particularly when user preferences and item attributes are available. They have also been found to be effective in addressing data sparsity and cold-start problems [17]. However, it is important to note that both content-based and collaborative systems also require a significant amount of data and may suffer from poor recommendations or lack of coverage in cases of limited data availability [18].

To address these issues, the author proposes a text-based recommender system explicitly designed to assist learners in the process of choosing the appropriate tutorial to obtain knowledge about MR. More specifically, the system analyzes the summaries of the projects offered by learners to recommend MR learning tasks based on the nature and goals of the learner’s needs. With the aid of machine learning algorithms, the recommender system seeks to optimize the MR learning experience, enabling learners to earn the necessary knowledge and skills as efficiently and personalized as possible. In this paper, the author explains how the text-based recommender system was researched and implemented with success, outlining the methods used during its development and evaluation process. Moreover, the paper presents a rigorous and comparative empirical evaluation of different existing machine learning algorithms.

This research contributes by presenting a novel text-based recommender system tailored specifically for aiding learners in selecting the most suitable MR tutorial tasks based on project summaries. By leveraging NLP techniques, the system offers personalized recommendations, streamlining the learning process and enhancing comprehension of distributed computing paradigms. This innovative approach bridges the gap between traditional learning methods and the practical application of MR solutions, providing learners with a more intuitive and efficient means of acquiring essential skills in distributed computing. The use of the proposed tool for teaching and learning MR can be employed throughout multiple domains and technologies requiring a project-based learning approach. Additionally, the potential applications of the results of this research are not limited to an individual approach, and they can be generalized to educational institutions, professional development institutions, and industrial facilities.

In the forthcoming sections, a comprehensive literature review of related work will lead to an understanding of literature and methods already existing that are in line with our research objectives. This will flow into the method section, where the approach utilized to develop the text-based recommender system and perform the experiments will be discussed. The findings from the experiments performed will be shared and discussed thoroughly, which will give an insight into how the proposed system fared in terms of performance and efficiency. Finally, the article will conclude with a detailed overview of the compendium of findings, the implications from the findings, and future research, which will help summarize our contributions and the general ground our work covers in the field of project-based learning of distributed computing.

2. LITERATURE REVIEW

Recommender systems play a crucial role in providing personalized recommendations to users across various domains. These systems typically consist of candidate generation, scoring, and re-ranking components. Candidate generation involves selecting a subset of items from a large corpus, scoring ranks of these candidates, and re-ranking considering additional constraints like user preferences or content freshness. Raghuwanshi and Pateriya [19] provide a comprehensive overview of recommendation techniques, challenges, and evaluation methodologies, highlighting the importance of application-focused approaches. There are three main types of recommender systems: content-based, collaborative, and hybrid systems. Content-based systems focus on item features to recommend similar items based on user preferences. Collaborative filtering methods use past interactions between users and items to suggest items based on similarity between users or items. Hybrid systems combine aspects of both content-based and collaborative filtering methods for more accurate recommendations Roy and Dutta [20], Kanwal *et al.* [21] delve into text-based recommender systems, offering insights into methods, datasets, and open challenges, thereby contributing to a better understanding of current paper recommendation practices. Meanwhile, Omar *et al.* [22] present a cloud-based recommender system leveraging NLP techniques and machine learning, tackling the complexities posed by big data environments.

Further advancements in recommendation techniques are evident in Kong *et al.* [23] proposal of VOPRec, a method for enhancing recommendation accuracy by learning vector representations of papers using text information and structural identity. Chughtai *et al.* [24] introduce an ontology-based recommendation model tailored for topic-specific article recommendations, aiming to identify best-fit reviewers efficiently. Moreover, Pérez-Núñez *et al.* [25] explore a text-based recommender system with explanatory capabilities, enhancing user understanding and transparency through explanatory content. Collectively, these studies have made significant contributions to enhancing recommendation accuracy, developing novel techniques, and advancing personalized recommendation engines across various domains, contributing to the evolving landscape of recommender systems.

Recommender systems have gained prominence in educational contexts due to their potential to personalize learning experiences and improve problem-solving abilities. In recent years, several studies have explored the impact of recommendation algorithms on learners' outcomes. Recommender systems in e-learning environments have the potential to personalize learning experiences and improve problem-solving abilities [26]. However, they require adaptations to facilitate learning, including system-centered and social adaptations [27]. These systems can assist in the discovery and retrieval of relevant and personalized learning content in online learning environments [28]. They strongly depend on the context or domain they operate in, and personalized recommendations can help learners overcome information overload [29].

Liu *et al.* [30] conducted a study investigating the impact of recommender systems on internet-based learning, revealing that personalized recommendations positively influenced learning outcomes by enhancing learner engagement, content relevance, and overall performance. Another study [8] further emphasized the role of recommender systems in problem-solving contexts, showcasing how the retrieval of relevant case-based information facilitated efficient access to resources and improved problem-solving efficiency. Building upon this, Tawfik *et al.* [31] extended the exploration by examining the influence of recommender systems on knowledge structure development. Their findings underscored the significance of personalized recommendations in scaffolding learners' understanding and fostering deeper knowledge acquisition through the recommendation of relevant case studies and learning materials. Collectively, these studies highlight the growing interest in recommender systems within educational settings, emphasizing their potential to personalize learning experiences and enhance problem-solving abilities. It synthesizes findings from several studies, indicating that personalized recommendations positively impact various aspects of learning, including engagement, content relevance, overall performance, problem-solving efficiency, and knowledge structure development.

MR a programming model introduced by Google in 2004, is a powerful tool for processing large data sets in distributed environments. It simplifies development and increases efficiency by breaking down data processing tasks into map and reduce phases [32]. The model is particularly useful for parallelizing computing in large clusters of machines and has been implemented in various environments, including graphics processors and shared-memory systems [33]. Its key advantage is its ability to isolate applications from the complexities of running distributed programs, such as data distribution, scheduling, and fault tolerance [34]. This programming model is used for processing and generating big datasets with a parallel, distributed algorithm on a cluster. MR offers scalability and fault tolerance, making it beneficial for various applications. The framework has been widely adopted in industry and academia, serving as the foundation for distributed computing frameworks like Apache Hadoop and Apache Spark.

In conclusion, the literature review demonstrates the significant role of recommender systems in various domains, including education, where they have shown promise in personalizing learning experiences and enhancing problem-solving abilities. While existing research has explored different recommendation techniques and their impacts on learning outcomes, there remains a gap in leveraging these systems specifically for accelerating the implementation of MR methodologies. The reviewed studies underscore the importance of personalized recommendations in facilitating access to relevant learning materials and improving learner engagement and performance. However, the literature primarily focuses on recommender systems within educational contexts and lacks emphasis on their integration with programming paradigms like MR. Therefore, there is a critical need for the development of a recommender system tailored to MR implementation, which can leverage user-provided project details to recommend suitable tutorials and resources, thereby streamlining the learning process and enhancing effectiveness. By bridging this gap, the proposed recommender system aims to address the challenges associated with learning MR and empower users to efficiently implement projects in distributed computing environments.

3. METHOD

In the data collection process, a dataset containing 107 distinct problem instances was initially sourced from two prominent books in the field of distributed systems and big data processing: "Hadoop in Action" [35] and "MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and

Other Systems" [36]. These problems encompassed fundamental concepts and standard algorithms used in MR applications, including word count, pirate talk, and finding the minimum and maximum values. Additionally, problems were collected from scientific articles to expand the dataset with more complex scenarios. These included advanced topics such as sentiment analysis [37], k-means classification [38], decision trees [39], and the Apriori algorithm [40]. By incorporating problems from books and scientific articles, the aim was to represent problem instances encountered in real-world MR applications comprehensively.

The collected dataset comprised attributes such as title, category, keywords, input, output, and goal. The selection criteria for problems were twofold: first, problems that demonstrated diversity in their approaches to solving distributed systems problems were sought. This ensured that the dataset encompassed a broad spectrum of techniques and methodologies. Second, problems that exhibited similarities in their solution strategies but varied in their specific problem conditions were prioritized. This approach allowed for exploring the effectiveness of different algorithms and techniques across a range of problem scenarios.

Overall, the data collection process aimed to create a diverse and representative dataset that could serve as a foundation for MR problem classification. The problems for dataset inclusion were meticulously chosen by experts well-versed in the domain of distributed systems employing the MR paradigm. Following an in-depth analysis of the dataset, three independent experts identified five primary characteristics representing solutions to challenges within distributed MR systems:

a. MR category-lines of data do not interact with each other.
b. MR category-related to counting.
c. MR category-contains few known keys and many unknown values.
d. MR category-needs to join data and use the output of one MR job in another MR job.
e. MR category-the problem is related to the condition.

This process culminated in the formation of a problems dataset, presented in Table 1. Each problem was assigned a binary label "0" or "1," signifying true/false, as determined by the established characteristics, based on prior research conducted by Orynbekova *et al.* [2]. The preprocessing steps involve converting textual data into numerical representations through vectorization of term frequency-inverse document frequency (TF-IDF). Initially, text features are extracted from the dataset's 'Goal' column. These features are then subjected to TF-IDF vectorization, transforming into numerical representations. The vectorizer is configured to disregard common English stop words and standardize all text to lowercase to ensure consistency.

Table 1. Labeled data: examples of classified problems by five categories

#	Problem attributes (title, category, keywords, input, output, and goal)	#1 MR category-lines of data do not interact with each other	#2 MR category-related to counting	#3 MR category-contains few known keys and many unknown values	#4 MR category-needs to join data and use the output of one MR job in another MR job	#5 MR category-the problem is related to the condition
1	problem 1	1	0	0	0	0
2	problem 2	1	1	1	0	0
3	problem 3	0	1	1	1	1
…	…	…	…	…	…	…

Subsequently, the target variables for classification, namely 'DataNotInteract', 'CountingRelated', 'FewKeys', 'JoinData', and 'ConditionProblem', are extracted from relevant columns in the dataset. Additional combinations of text features are created by concatenating different columns from the dataset to enrich the feature set. These combinations include 'Goal' and 'Title'; 'Goal', 'Title' and 'Category'; 'Goal', 'Title', 'Category' and 'Input'; 'Goal', 'Title', 'Category', 'Input' and 'Output'; and 'Goal', 'Title', 'Category', 'Input', 'Output' and 'Keywords'.

For each of these combinations, TF-IDF vectorization is applied using the same configuration as before. The target variables remain consistent across all these additional feature combinations. These preprocessing steps lay the groundwork for transforming textual data into a format suitable for classification tasks, while simultaneously facilitating the extraction of target variables.

Integrating an artificial expansion strategy addresses the challenge of working with a limited dataset. Specifically, the script employs a paraphrasing technique facilitated by the nlpaug library, utilizing contextual word embedding models such as bidirectional encoder representations from transformers (BERT). The paraphrase function takes a given text and generates a paraphrased version. This augmentation process is applied to each textual entry in the dataset, effectively expanding the original dataset. Consequently, the dataset grows to encompass 523 tuples, each achieved through the application of paraphrasing on the existing textual data. This approach enhances the dataset's diversity and aims to mitigate the limitations associated with the initially small dataset, ultimately contributing to improved model training and performance.

Feature selection was a crucial step in the methodology for developing classification models using both the Naive Bayes algorithm and logistic regression. Various combinations of attributes, including 'Goal', 'Title', 'Category', 'Input', 'Output', and 'Keywords' were explored as potential features. Specifically, combinations such as: "goal + title", "goal + title + category", "goal + title + input + output", "goal + title + keywords", "goal + title + category + input + output", "goal + title + category + keywords", and "goal + title + input + output + keywords" were considered. Necessarily, it was ensured that all combinations included the main features "Goal" and "Title". This approach aimed to comprehensively evaluate the impact of different feature combinations on the performance of the classification models.

A comprehensive approach incorporating cross-validation was employed to evaluate the performance of classification models within the methodology of our scientific study. The dataset underwent stratified k-fold cross-validation with k=5, ensuring thoroughness and mitigating the risk of overfitting. This technique involves partitioning the data into k equal-sized folds while preserving the class distribution. During each iteration of the cross-validation process, one fold was used for testing, while the remaining folds were utilized for training. This process was repeated k times, with each fold serving as the testing set precisely once. Subsequently, hyperparameter tuning was conducted using logistic regression, considering a range of regularization parameter values (C) defined in a parameter grid. A logistic regression model was trained on the training data for each C value and evaluated on the testing set. The performance of each model was assessed using the F1 score metric, calculated with the f1_score from the sklearn.metrics module, which offers a balanced measure considering both precision and recall. The model yielding the highest F1 score on the testing data was deemed the optimal choice. Finally, the selected model for each classification task was saved for further analysis and comparison. This rigorous evaluation process, augmented by cross-validation, ensured the robustness and reliability of the classification models employed in our study.

4. RESULTS AND DISCUSSION

The logistic regression model underwent rigorous testing across incremental sets of textual features, demonstrating its effectiveness in the binary classification problem and its capability to leverage varied textual features for enhanced predictive accuracy. A comparative analysis between the Naive Bayes and logistic regression models was conducted to gain deeper insights into their predictive capabilities. A hybrid approach was employed, leveraging the strengths of Naive Bayes and logistic regression models. The optimization process involved selecting the best-performing model and feature combination for each column, demonstrating the adaptability of utilizing different models and feature combinations for distinct target variables. This nuanced approach capitalized on the unique strengths of each model, resulting in a tailored and optimized predictive framework for each column.

Table 2 shows the perceptible findings from our Naive Bayes models applied to different MR categories, showcasing the impact of varying feature combinations. Remarkably, the first MR category achieved a remarkable best cross-validated F1 Score of 0.99 and a flawless best F1 score of 1 on the test set, indicating robust predictive performance. The second and third categories also demonstrated strong capabilities, emphasizing the effectiveness of our chosen feature sets.

Table 2. Naive Bayes results

MR category	Features	Best cross-validated F1 score	Best F1 score on test set	Alpha
#1 MR category	Goal+Title+Category+Input+Output	0.99	1	0.01
#2 MR category	Goal+Title+Category	0.94	1	0.01
#3 MR category	Goal+Title+Category+Input+Output+Keywords	0.99	1	0.001
#4 MR category	Goal+Title+Category+Input+Output	0.96	0.97	0.001
#5 MR category	Goal+Title+Category	0.98	0.97	0.001

Table 3 presents the perceptible findings from our logistic regression models applied to different MR categories, emphasizing key features and performance metrics. Noticeably, the first MR category exhibited an exceptional best cross-validated F1 score of 0.98 and a perfect best F1 score of 1 on the test set, underscoring its robust predictive capabilities. Similarly, the second and third categories, with varying feature sets, demonstrated high predictive accuracy with F1 scores of 0.98 and 0.97, respectively. Notably, the optimal regularization parameter (C) for all categories was consistently set to 1, indicating the stability and reliability of our logistic regression models across different feature combinations. These results underscore the effectiveness of our logistic regression approach in providing accurate and consistent recommendations for single-machine problems adapting to distributed systems in MR.

Table 3. Logistic regression results

MR category	Features	Best cross-validated F1 score	Best F1 score on test set	C
#1 MR category	Goal+Title+Category+Input+Output	0.98	1	1
#2 MR category	Goal+Title	0.98	1	1
#3 MR category	Goal+Title+Category+Input+Output	0.97	1	1
#4 MR category	Goal+Title+Category+Input+Output+Keywords	0.97	1	1
#5 MR category	Goal+Title+Category+Input+Output+Keywords	0.99	1	1

The results obtained from the rigorous testing of Naive Bayes and logistic regression models across various MR categories reveal valuable insights into the effectiveness of different feature combinations. Notably, the hybrid approach, presented in Table 4, combining multiple models' strengths is powerful, especially in tasks like MR categories with different data types and patterns to consider. Selecting the best-trained models for each category and leveraging their strengths improves prediction accuracy and efficiency. It's like having a toolbox of different techniques and using the right tool for each specific job. This approach allows you to tailor your solution to the unique characteristics of each category, optimizing performance overall. The consistently high F1 scores across different MR categories underscore the robustness of the models, particularly in the first category, where logistic regression achieved a near-perfect score. The selection of optimal alpha values for Naive Bayes and regularization parameters (C) for logistic regression demonstrates the careful consideration of hyperparameters, contributing to the models' stability and reliability.

Table 4. Hybrid approach

MR category	Algorithm	Features	Best cross-validated F1 score	Best F1 score on test set	Alpha/ C
#1 MR category	Naive Bayes	Goal+Title+Category+Input+Output	0.99	1	0.01
#2 MR category	Logistic regression	Goal+Title	0.98	1	1
#3 MR category	Naive Bayes	Goal+Title+Category+Input+Output+Keywords	0.99	1	0.001
#4 MR category	Logistic regression	Goal+Title+Category+Input+Output+Keywords	0.97	1	1
#5 MR category	Logistic regression	Goal+Title+Category+Input+Output+Keywords	0.99	1	1

As a result, a recommender system was meticulously crafted, integrating five distinct models designed to accurately forecast the assignment of five categorical labels to novel problem instances. Following the system's development, rigorous evaluation procedures were employed to gauge its predictive efficacy. Expert opinion was solicited and meticulously analyzed to assess the system's performance in predicting the labels of newly encountered problems. The outcome of this evaluation revealed promising results, affirming the effectiveness of the recommender system in its predictive capabilities.

However, it is crucial to acknowledge certain limitations in the study. Although expanded through paraphrasing, the dataset remains relatively small, and its generalizability to a broader range of problem instances warrants further investigation. Additionally, the study primarily focuses on binary classification, and extending the approach to handle multi-class scenarios could be an avenue for future research. Despite these limitations, the comprehensive evaluation and hybrid approach presented in this study offer a valuable contribution to the domain of recommender systems for single-machine problems adapting to distributed systems in MR.

5. CONCLUSION

In conclusion, this research significantly advances the understanding of recommender systems in the context of adapting single-machine problems to distributed systems within the MR framework. The hybrid approach, seamlessly integrating Naive Bayes and logistic regression models, emerges as a robust and adaptable strategy for tailoring predictive frameworks to distinct MR categories. The consistently high F1 scores and thorough optimization of hyperparameters underscore the efficacy of the proposed approach.

Furthermore, amalgamating five distinct models, the developed recommender system demonstrates promising predictive efficacy in assigning categorical labels to novel problem instances. Expert evaluation further substantiates the system's capabilities, affirming its utility in real-world applications. However, it's crucial to acknowledge the study's limitations, including the relatively small dataset size and the focus on binary classification.

Future research efforts may explore enlarging the dataset and extending the approach to handle multi-class scenarios, enhancing the system's versatility and applicability. Nevertheless, despite these limitations, the comprehensive methodology and hybrid approach presented herein significantly contribute to the domain of recommender systems for single-machine problems adapting to distributed systems in MR. This work provides an effective methodology for guiding users toward relevant solutions and sets the stage for further advancements in MR problem classification. It underscores the importance of rigorous methodology in data-driven scientific research, paving the way for scalable and refined recommender systems tailored to a broader range of problem instances.

ACKNOWLEDGEMENTS

This research has been funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23483919).

REFERENCES

[1] S. Maitrey and C. K. Jha, "Handling Big Data Efficiently by Using Map Reduce Technique," in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, IEEE, Feb. 2015, pp. 703–708, doi: 10.1109/CICT.2015.140.

[2] K. Orynbekova, A. Bogdanchikov, S. Cankurt, A. Adamov, and S. Kadyrov, "MapReduce Solutions Classification by Their Implementation," *International Journal of Engineering Pedagogy (iJEP)*, vol. 13, no. 5, pp. 58–71, Jul. 2023, doi: 10.3991/ijep.v13i5.38867.

[3] R. Jha, A.-A. Jbara, V. Qazvinian, and D. R. Radev, "NLP-driven citation analysis for scientometrics," *Natural Language Engineering*, vol. 23, no. 1, pp. 93–130, Jan. 2017, doi: 10.1017/S1351324915000443.

[4] V. B. P. Tolety and E. V. Prasad, "Hybrid content and collaborative filtering based recommendation system for e-learning platforms," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1543–1549, Jun. 2022, doi: 10.11591/eei.v11i3.3861.

[5] M. S. B. M. Omar, M. Ismail, N. M. Diah, S. Ahmad, and H. A. Rahman, "Modelling the recommendation technique for achieving awareness in serious game for obesity," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1418–1424, Dec. 2019, doi: 10.11591/eei.v8i4.1627.

[6] Z. Cao, X. Qiao, S. Jiang, and X. Zhang, "An Efficient Knowledge-Graph-Based Web Service Recommendation Algorithm," *Symmetry*, vol. 11, no. 3, p. 392, Mar. 2019, doi: 10.3390/sym11030392.

[7] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 239–249, Apr. 2019, doi: 10.1007/s11257-019-09231-w.

[8] A. A. Tawfik, H. Alhoori, C. W. Keene, C. Bailey, and M. Hogan, "Using a Recommendation System to Support Problem Solving and Case-Based Reasoning Retrieval," *Technology, Knowledge and Learning*, vol. 23, no. 1, pp. 177–187, Apr. 2018, doi: 10.1007/s10758-017-9335-y.

[9] D. P., S. Chakraborti, and D. Khemani, "Query Suggestions for Textual Problem Solution Repositories," *European Conference on Information Retrieval*, 2013, pp. 569–581, doi: 10.1007/978-3-642-36973-5_48.

[10] Y. Betancourt and S. Ilarri, "Use of Text Mining Techniques for Recommender Systems," in *Proceedings of the 22nd International Conference on Enterprise Information Systems*, SCITEPRESS - Science and Technology Publications, 2020, pp. 780–787, doi: 10.5220/0009576507800787.

[11] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, Jan. 2020, doi: 10.1145/3285029.

[12] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 1–37, Jun. 2019, doi: 10.1007/s10462-018-9654-y.

[13] S. Jain, H. Khangarot, and S. Singh, "Journal Recommendation System Using Content-Based Filtering," in *Recent Developments in Machine Learning and Data Analytics*, 2019, pp. 99–108, doi: 10.1007/978-981-13-1280-9_9.

[14] B. Towle and C. Quinn, "Knowledge based recommender systems using explicit user models," *Proceedings of the AAAI Workshop on Knowledge-Based Electronic Markets*, vol. 1, no. 1, pp. 74–77, 2000.

[15] M. Zhu, D. Zhen, R. Tao, Y. Shi, X. Feng, and Q. Wang, "Top-N Collaborative Filtering Recommendation Algorithm Based on Knowledge Graph Embedding," *International Conference on Knowledge Management in Organizations*, 2019, pp. 122–134, doi: 10.1007/978-3-030-21451-7_11.

[16] J. Chicaiza and P. Valdiviezo-Diaz, "A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions," *Information*, vol. 12, no. 6, p. 232, May 2021, doi: 10.3390/info12060232.

[17] Q. Guo *et al.*, "A Survey on Knowledge Graph-Based Recommender Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022, doi: 10.1109/TKDE.2020.3028705.

[18] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems*, vol. 69, no. 2, pp. 175–186, 2000.

[19] S. K. Raghuwanshi and R. K. Pateriya, "Recommendation Systems: Techniques, Challenges, Application, and Evaluation," in *Soft Computing for Problem Solving: SocProS*, 2019, pp. 151–164, doi: 10.1007/978-981-13-1595-4_12.

[20] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, p. 59, Dec. 2022, doi: 10.1186/s40537-022-00592-5.

[21] S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, "A Review of Text-Based Recommendation Systems," *IEEE Access*, vol. 9, pp. 31638–31661, 2021, doi: 10.1109/ACCESS.2021.3059312.

[22] H. K. Omar, M. Frikha, and A. K. Jumaa, "Big data cloud-based recommendation system using NLP techniques with machine and deep learning," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 5, pp. 1076–1083, Oct.

2023, doi: 10.12928/telkomnika.v21i5.24889.

[23] X. Kong, M. Mao, W. Wang, J. Liu, and B. Xu, “VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation,” *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 226–237, Jan. 2021, doi: 10.1109/TETC.2018.2830698.

[24] G. R. Chughtai, J. Lee, M. Shahzadi, A. Kabir, and M. A. S. Hassan, “An efficient ontology-based topic-specific article recommendation model for best-fit reviewers,” *Scientometrics*, vol. 122, no. 1, pp. 249–265, Jan. 2020, doi: 10.1007/s11192-019-03261-2.

[25] P. Pérez-Núñez, J. Díez, A. Bahamonde, and O. Luaces, “Text-based recommender system with explanatory capabilities,” *Research Square (preprint)*, 2022, doi: 10.21203/rs.3.rs-1536768/v2.

[26] A. Klašnja-Milićević, M. Ivanović, and A. Nanopoulos, “Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions,” *Artificial Intelligence Review*, vol. 44, no. 4, pp. 571–604, Dec. 2015, doi: 10.1007/s10462-015-9440-z.

[27] J. Buder and C. Schwind, “Learning with personalized recommender systems: A psychological view,” *Computers in Human Behavior*, vol. 28, no. 1, pp. 207–216, Jan. 2012, doi: 10.1016/j.chb.2011.09.002.

[28] T. Y. Tang, B. K. Daniel, and C. Romero, “Recommender systems in social and online learning environments,” *Expert Systems*, vol. 32, no. 2, pp. 261–263, Apr. 2015, doi: 10.1111/exsy.12058.

[29] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, and L. C. Jain, “Recommender Systems in E-Learning Environments,” in *E-Learning Systems Intelligent Systems Reference Library*, 2017, pp. 51–75, doi: 10.1007/978-3-319-41163-7_6.

[30] C. Liu, C. Chang, and J. Tseng, “The effect of recommendation systems on internet-based learning for different learners: A data mining analysis,” *British Journal of Educational Technology*, vol. 44, no. 5, pp. 758–773, Sep. 2013, doi: 10.1111/j.1467-8535.2012.01376.x.

[31] A. A. Tawfik, K. Kim, and D. Kim, “Effects of case library recommendation system on problem solving and knowledge structure development,” *Educational Technology Research and Development*, vol. 68, no. 3, pp. 1329–1353, Jun. 2020, doi: 10.1007/s11423-020-09737-w.

[32] L. Cheng-hua, Z. Xin-fang, J. Hai, and X. Wen, “MapReduce: A new programming model for distributed parallel computing,” *Computer Engineering & Science*, vol. 33, no. 3, p. 129, 2011, doi: 10.3969/j.issn.1007130X.2011.

[33] V. Vijayalakshmi, A. Akila, and S. Nagadivya, “The survey on MapReduce,” *International Journal of Engineering Science and Technology (IJEST)*, vol. 4, no. 7, p. 2012, 2012.

[34] S. Sakr, “General-Purpose Big Data Processing Systems,” in *Big Data 2.0 Processing Systems. SpringerBriefs in Computer Science*, 2016, pp. 15–39, doi: 10.1007/978-3-319-38776-5_2.

[35] C. Lam, “Hadoop in Action,” in *Manning Publications*, 2010.

[36] D. Miner and A. Shook, “MapReduce design patterns,” in *O’Reilly Media, Inc.*, 2012.

[37] P. Gupta, P. Kumar, and G. Gopal, “Sentiment analysis on hadoop with hadoop streaming,” *International Journal of Computer Applications*, vol. 121, no. 11, pp. 4–8, 2015.

[38] T. H. Sardar and Z. Ansari, “Partition based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions,” *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 247–261, Dec. 2018, doi: 10.1016/j.fcij.2018.06.002.

[39] W. Dai and W. Ji, “A mapreduce implementation of C4. 5 decision tree algorithm,” *International journal of database theory and application*, vol. 7, no. 1, pp. 49–60, 2014.

[40] S.-Y. Choi and K. Chung, “Knowledge process of health big data using MapReduce-based associative mining,” *Personal and Ubiquitous Computing*, vol. 24, no. 5, pp. 571–581, Oct. 2020, doi: 10.1007/s00779-019-01230-3.

BIOGRAPHIES OF AUTHORS

Kamila Orynbekova is a Senior Lecturer in the Department of Computer Sciences, Faculty of Engineering and Natural Sciences, and a head of the Distributed Systems and Computing Laboratory at SDU University, Kaskelen, Almaty, Kazakhstan. Also, she is a Ph.D. student in the Computer Sciences educational program. Upon completing secondary education, she matriculated at Suleyman Demirel University, where she was awarded a Bachelor's degree in Computing Systems and Software in 2013. She furthered her academic pursuits at the same University, culminating in attaining a Master’s in Technical Sciences in 2017, focusing on the same specialization. She can be contacted at email: kamila.orynbekova@sdu.edu.kz.

Shirali Kadyrov received the Ph.D. and M.Sc. degrees from Ohio State University, USA in 2010 and 2009, respectively, and the B.Sc. degree from Bogazici University, Turkey in 2004. Currently, he is working as the Associate Professor of Mathematics at New Uzbekistan University. He has held impactful academic positions throughout his career. His prior roles include a dean of Faculty at Oxus University, Uzbekistan, full professorship at SDU University, Kazakhstan, and positions such as a research assistant at ETH Zurich, Switzerland, an EPSRC postdoctoral resercher at University of Bristol, UK, and a full-time faculty member at Nazarbayev University, Kazakhstan. His research interests encompass dynamical systems, ergodic theory, number theory, and data science, reflecting a diverse and prolific scholarly profile. He can be contacted at email: sh.kadyrov@newuu.uz.

Andrey Bogdanchikov research interests are big data distributed systems and taught courses about big data on Ph.D., B.Sc., and M.Sc. levels. Now he holds the title of Associate Professor at SDU University's Faculty of Engineering and Natural Sciences, within the Department of Information Systems, and Vice-Rector of Academic Affairs situated in Abylai khan street 1/1, Kaskelen, Kazakhstan. He obtained his Doctor of Philosophy degree in 2014 from Suleyman Demirel University, Kazakhstan. His areas of expertise include the fields of distributed systems, parallel computing, and programming languages. He can be contacted at email: andrey.bogdanchikov@sdu.edu.kz.

Saidakmal Oktamov received a Bachelor's degree in Information Systems from SDU University in 2023. He is currently a Software Developer at Andersen Lab, where he has been contributing since 2023, focusing on developing scalable and efficient software solutions. Hi is also pursuing a Master's degree in Information Systems at SDU University, expected to complete in 2025. His research interests include blockchain, machine learning, artificial intelligence, and big data systems. He is passionate about leveraging these technologies to solve complex problems. He can be contacted at email: oktamovsaidakmal@gmail.com.