Computers in Industry 148 (2023) 103884

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Reinforcement learning based trustworthy recommendation model for digital twin-driven decision-support in manufacturing systems

Flávia Pires [a,*,1], Paulo Leitão [a,b,2], António Paulo Moreira [c,3], Bilal Ahmad [d,4]

[a] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus Santa Apolónia, Bragança 5300-253, Portugal
[b] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Campus Santa Apolónia, Bragança 5300-253, Portugal
[c] INESC TEC - INESC Technology and Science, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, Porto 4200-465, Portugal
[d] University of Warwick, Warwick Manufacturing Group (WMG), 6 Lord Bhattacharyya Way, Coventry CV4 7AL, United Kingdom

A R T I C L E I N F O

Keywords:
Digital twin
Decision-support
Recommendation systems
Similarity measures
Trust-based model

A B S T R A C T

Digital twin is one promising and key technology that emerged with Industry 4.0 to assist the decision-making process in multiple industries, enabling potential benefits such as reducing costs, and risk, improving efficiency, and supporting decision-making. Despite these, the decision–making approach of carrying out a what-if simulation study using digital twin models of each and every possible scenario independently is time-consuming and requires significant computational resources. The integration of recommendation systems within the digital twin-driven decision-support framework can support the decision-making process by providing targeted scenario recommendations, reducing the decision-making time and imposing decision- making efficiency. However, recommendation systems have inherent challenges, such as cold-start, data sparsity, and prediction accuracy. The integration of trust and similarity measures with recommendation systems alleviates the challenges mentioned earlier, and the integration of machine learning techniques enables better recommendations through their ability to simulate human learning. Having this in mind, this paper proposes a trust-based recommendation approach using a reinforcement learning technique combined with similarity measures, which can be integrated within a digital twin-based what-if simulation decision-support system. This approach was experimentally validated by performing accurate recommendations in an industrial case study of a battery pack assembly line. The results show improvements in the proposed model regarding the accuracy of the prediction about the user rating of the recommended scenarios over the state-of-the-art recommendation approaches, particularly in cold-start and data sparsity scenarios.

1. Introduction

In the last few years, the increasing level of digitalisation of the manufacturing sector by adopting technologies linked to the fourth industrial revolution, e.g., Internet-of-Things (IoT) and Artificial Intelligence (AI), has led to a significant increase in the data that is available for performing decision-making (Ahuett-Garza and Kurfess, 2018). The use of the digital twin concept facilitated the user decision-support through the virtual representation of the physical system being updated with real-time data and analysed using simulation (e.g., what-if simulation, allowing testing different "what-if" scenarios, as different design scenarios) and/or advanced data analytics (Pires et al., 2021b). The concept behind the digital twin technology has evolved since it was first proposed in 2002 by Michael Grieves (Grieves and Vickers, 2017), presenting now the capabilities of real-time control, adaptation, optimisation, and decision-support (Rasheed et al., 2020).

Typically, the user's manual decision-making process from the results provided by the digital twins can be very time-consuming (Pires et al., 2021b). In addition to the collected data from sensors, inferred data from the digital twin what-if simulations, and the information

* Corresponding author.
E-mail addresses: fpires@ipb.pt (F. Pires), pleitao@ipb.pt (P. Leitão), amoreira@fe.up.pt (A.P. Moreira), B.Ahmad@warwick.ac.uk (B. Ahmad).
[1] ORCID: 0000-0001-7899-3020
[2] ORCID: 0000-0002-2151-7944
[3] ORCID: 0000-0001-8573–3147
[4] ORCID: 0000-0002-2752-9161

https://doi.org/10.1016/j.compind.2023.103884
Received 2 November 2022; Received in revised form 22 February 2023; Accepted 25 February 2023
Available online 29 March 2023

provided by the data analysis, the user knowledge of the system plays a vital role in the decision- making process. The recommendation systems can support manual decision-making by accurately estimating the degree to which a user will prefer a particular scenario and the recommendation of a top-N list of scenarios that may interest the user. These two functionalities allow the user to handle a large amount of information rapidly, e.g., from hundreds of possible scenarios to be recommended only presented three, consequently decreasing the decision-making time (Isinkaye, Folajimi and Ojokoh, 2015).

Besides reducing decision-making time, integrating digital twin technology with recommendation systems improves decision-support efficiency (Isinkaye et al., 2015). This combination mainly provides a broader knowledge foundation, the capability to explore and assess alternative scenarios, the use of simulation models of manufacturing systems, the prediction of advantages and disadvantages of different solutions, provide recommendations regarding individual and changing company goals, and allows the introduction of user feedback into the recommendation cycle to assess user trust in the system and recommendations (Kunath and Winkler, 2018; Sala et al., 2019).

The decision-making process based on a digital twin depends on the user's trust in the system (Chaplin et al., 2020), being necessary to develop new recommendation systems approaches. These new approaches should be capable of learning and improving the user trust in the decision-support system and overcoming the current challenges traditional recommendation systems face. The most relevant challenges are related to the cold-start problem (Son, 2016), data sparsity (Reshma et al., 2016), and improving the recommendation accuracy through the prediction quality (Meyer, 2012). The cold-start problem relates to the lack of rating information to perform predictions, both in terms of users and scenarios, and the data sparsity problem is the irregular, insufficient or highly varying user rating information resulting in a sparse rating matrix (Sharma and Singh, 2016).

Cold-start and data sparsity problems are common and prevalent in many real-world applications, such as music and video streaming services (Velankar and Parag, 2023), e-commerce (Yin and Luo, 2021), healthcare (Tan et al., 2022), and financial services (Pereira and Varma, 2019). In the case of the manufacturing industry, many real-world applications may arise, such as maintenance, quality control, supply chain, shop-floor design, and assembly of complex products. For example, in the case of workflow optimisation, the main goal is to improve the shop-floor productivity by applying a recommendation system that is able to perform recommendations to a new user with limited or no historical data about the performance of new workflows or processes, or the user feedback. Another example is the application of a recommendation system for the execution of complex maintenance tasks, where recommend actions are required to support a new maintenance technician, based on few historical data associated to an unusual situation.

In the literature, several works (Abdul-Raham and Hailes, 1998; Guo, 2013; Massa and Avesani, 2007; Jamali and Ester, 2010, 2009) have already addressed the use of trust in recommendation systems. For example, trust has been used to improve the predictive performance of the traditional recommendation systems (Guo, 2013) and to alleviate data sparsity and cold-start problems (Massa and Avesani, 2007). The integration of a machine learning (ML) techniques, namely reinforcement learning (RL), with a recommendation system was proposed to enable decision-making in data-deprived situations, being able to tackle the data sparsity problem (Rasheed et al., 2020), and the similarity measures can also be integrated within the recommendation systems to alleviate the cold-start problems (Jain and Mahara, 2019). However, one current limitation of these approaches is the lack of integration, which needs to be efficiently addressed to offer a complete end-to-end solution to industry applications.

Having this in mind, this paper introduces a trust-based recommendation approach that considers an RL algorithm and similarity measures to be included in a digital twin-based what-if simulation decision-making process. The proposed trust-based recommendation model aims to overcome the previously stated problems, particularly cold-start and data sparsity problems. The proposed approach was experimentally validated in a battery pack assembly line case study, which exhibits cold-start and data sparsity problems, and the results were compared with state-of-the-art recommendation models in terms of recommendation accuracy. The importance of applying the proposed approach within a digital twin architecture in the case study is the performance of fast and accurate recommendations regarding the best logistical scenarios to be applied in the battery pack assembly line according to the simulation results, the human trust and knowledge of the system.

The rest of the paper is organised as follows. Section 2 presents an overview of the state-of-the-art recommendation system approaches, comparing their characteristics and limitations. Section 3 formalises the proposed trust-based recommendation model for the digital twin decision-making process. Section 4 describes the experimental validation, including the description of the case study, the evaluation metrics, and the analysis of experimental results. Lastly, Section 5 presents the conclusions and future work.

2. Overview of recommendation systems'

The application of digital twins in the manufacturing environment is directly connected to the system performance improvement by implementing decision support approaches, as is the case of recommendation systems (Pires et al., 2021b). Recommendation systems emerged in the 90s, being defined as systems capable of providing individual recommendations to users (Sharma and Singh, 2016). They can be considered as information filtering systems that deal with the information overload problem, being able to filter information based on user's preferences (Malik et al., 2020; O'Donovan and Smyth, 2005).

The recommendation approaches can be classified as traditional and social recommendation systems. Traditional recommendation systems assume that the users are independent and identically distributed, ignoring the social interactions or connections among them, basing the recommendation and prediction of ratings on the rating data. In the case of social recommendation systems, it is assumed that users are correlated, establishing measurable social relationships, combining the rating data, trust data and social information (Ma et al., 2008; Tang et al., 2013). The following subsections will further elaborate on the different traditional and social recommendation system approaches.

2.1. Collaborative filtering

Collaborative filtering (CF) is a domain-independent recommendation approach that belongs to the traditional recommendation systems group and was one of the first techniques used in recommendation systems (Isinkaye et al., 2015). The concept behind the CF is that users sharing similar interests in one area tend to be interested in similar items in other areas (Sharma and Singh, 2016). This approach considers a user-item matrix that allows for identifying the user preferences for items. The matching between users is performed considering the interests and preferences, calculating similarities between their profiles and making recommendations (Isinkaye et al., 2015). One of the first implementations of this technique was the *UserCF* which is based on the users' ratings and similarity of preferences. The main idea behind this model is to find similar users to the target user and use their preferences to generate recommendations. This model is affected by cold-start and data sparsity problems (Resnick et al., 1994).

CF can be divided into two categories, memory-based and model-based (Malik et al., 2020). The memory-based filtering is also divided into two classes, user-based and item-based. The user-based CF calculates the similarity between the users by comparing ratings for the same item and calculating the predicted rating through a weighted average. The item-based CF predicts an item by using the similarity between the item and the selected item by the user. The model-based CF is a learning technique based on a prediction model that uses the user-item matrix information, namely the user rating data (Malik et al., 2020). The CF approach can also be divided into two disciplines, namely the neighbourhood approach and latent factor models (Koren, 2008). The neighbourhood approach focuses on using the relationships between items or, in the alternative, between users. In the item-oriented approach, a user's preference towards an item is determined based on the rating of similar items by the same user. The latent factor models transform items and users to the same latent factor space, making them directly comparable. In this case, a user's preference is measured by each item's factor. The *SVD++*, an extension of the Singular Value Decomposition (SVD), is a latent factor model that elaborates the recommendations based on the matrix factorisation, including a set of characteristics that model the item-item relations and the users' implicit feedback. The algorithm considers the latent factors of the user and item as vectors modelled in a low-dimensional space and accounts for the overall average of all the used items by the user (Koren, 2010).

2.2. Content-based filtering

Content-based filtering (CBF) is a domain-dependent recommendation technique based on the analysis of features of items, in which the recommendations are performed based on the user profiles by using the features extracted from the content of the items that have been previously evaluated by the user (Sharma and Singh, 2016).

The CBF follows two strategies to recommend items to users: the classifier-based and neighbour methods (Weng, 2008). The classifier-based method uses a classifier that decides if the item should be recommended or not, depending on its content. In the second strategy, the items the user has rated are stored, and the constructed network of items is used to uncover the user's interest in a new item (Portugal et al., 2018). The models that CBF mainly uses are learning models that can make use of statistical analysis or ML techniques, such as the Vector Space Model (e.g., Term Frequency Inverse Document Frequency (TF/IDF)), or Probabilistic Models (e.g., Naive Bayes Classifier, Decision Trees, and Neural Networks) (Isinkaye et al., 2015).

2.3. Hybrid-based recommendation

The main premise of the hybrid-based recommendation (HBR) system is the combination of two or more approaches in order to obtain better performance (Sharma and Singh, 2016). The HBR can be implemented in various forms, e.g., implementing collaborative and content-based methods independently and aggregating their predictions, integrating characteristics from a CBF model into a CF model, and building a new consolidated model that incorporates aspects of both CBF and CF (B. Thorat et al., 2015). One of the first HBR systems was a website recommendation, named Fab system, which used CF to find a user with similar website preferences and the CBF for finding similar website contents (Balabanović and Shoham, 1997).

In addition to combining traditional recommendation approaches, recently data mining and ML techniques have been used to build HBR systems, namely Neural Networks, Fuzzy Logic, SVD, Bayesian techniques, and RL (Urdaneta-Ponte et al., 2021; Lin et al., 2021; Çano and Morisio, 2017). This combination, e.g., with RL, presents several advantages, namely, the recommendation strategies can be updated during interactions, the long-term cumulative reward from the users' feedback is maximised, the exploration and exploitation of recommendations are balanced, and the continuous learning capability allows the update of the recommendations according to the changes of the user interests (Lin et al., 2021).

2.4. Trust-based recommendation

Trust has become a key factor in the decision-making process in highly dynamic and decentralised environments (Selmi et al., 2016). The trust concept can be divided into two categories, namely context-specific interpersonal trust, which is the user trust in another user regarding a specific situation, and system-impersonal trust, which describes the user trust over the system itself (Abdul-Raham and Hailes, 1998).

Trust-based recommendation (TBR) systems are known for including the knowledge from the social trust network created by users to generate individual recommendations (Victor et al., 2011), being included in the social recommendation systems category. TBR systems can be defined as collaborative systems based on the trust concept as a quantifier of the user relationships (Ma et al., 2009; Massa and Avesani, 2007; O'Donovan and Smyth, 2005).

The social recommendation systems have been explored, and several models have been implemented, such as the *SocialRec* that integrates social regularisation into a matrix factorisation model, using a user-feature matrix factorised by ratings and trust. This model uses social relationships between the users to generate more accurate and personalised recommendations. As other models the performance of this model can be affected by the cold-start model, this is also a very computational heavy algorithm specially for large datasets, not being well-suited for real-time scenarios (Ma et al., 2008). The *SocialRSTE* is a social trust ensemble method to linearly combine a basic matrix factorisation model and a trust-based neighbourhood model. The main idea behind this algorithm is use the *SVD* to learn the latent factors, capturing the preferences of users and characteristics of the items, and incorporating the social relationships into the factorisation (Ma et al., 2009). The same authors have proposed the *SocialReg*, which uses the active user-specific vector to calculate the average of its trusted users, and used it as regularisation to form a new matrix factorisation model. This model leverages from using the social relationships between users to improve the performance of the factorisation matrix (Ma et al., 2011). The *SocialMF* model is built upon the *SocialRec* principles, reformulating the use of trusted users to the formation of the active user's user-specific vector and enabling the trust propagation property. In this model the features of a user are dependent on the features of its direct neighbours, and recursively the features of the direct neighbours are also dependent on its direct neighbours. This method combines the matrix factorisation with the trust propagation in order to produce recommendations (Jamali and Ester, 2010). The *TrustSVD* model is an extension of the *SVD++* model that includes a trust-based matrix factorisation technique, which uses rating explicit and implicit feedback, and the explicit and implicit user social trust data. The model was adapted with a weighted regularisation to regularise the latent feature vectors of the user and items (Guo et al., 2015). Lastly, the *TrustWalker* model is based on a random walk model that combines an item-based ranking method and a trust-based nearest neighbour model. The model considers the ratings of the target item and of the similar items, the probability of using the rating of the similar item is directly affected by the length of the walk. With the *TrustWalker* is possible to calculate the confidence of the made predictions (Jamali and Ester, 2009).

2.5. Summary of challenges and gaps

Throughout time several comparative studies have been performed comparing the different recommendation approaches, such as Al Fararni et al. (2020); Sorde and Deshmukh (2015); and Pandya and Pathar (2020). Table 1 summarises the characteristics and limitations of the above mentioned approaches, providing a comparative analysis.

The CF approach presents several advantages, such as being a domain-independent technique that enables to filter of any item only based on the historical information about a given user preference (Kim et al., 2010). This approach works very well for recommendation environments with large amounts of data. Another key advantage is that recommendations are only based on the user rating. The memory-based CF makes the recommendation system easy to manage due to the ease of adding new data incrementally. In the case of the model-based CF, the

Table 1
Comparative analysis of the recommendation systems' approaches.

Approach	Characteristics	Limitations
Collaborative Filtering	• Domain-independent technique; • Performs better for a considerable amount of available data; • Performs new recommendations based on the exploration of results.	• Lack of performance for cold-start and data sparsity problems; • Dependent on the available user rating data; • Requires significant computer power for performing the recommendations.
Content-Based Filtering	• Does not rely on other users' preferences and allows comparison between items; • Does not suffer from the new items cold- start; • Explains how the system performs the recommendation; • Recommendation quality increases over time, and user usage.	• Hard to perform for a dense user dataset or with varying preferences; • Suffers from a cold-start problem related to new users; • Not possible to perform an exploration of different recommendations; • Knowledge of the field is often necessary.
Hybrid-Based Recommendation	• Overcomes some problems of previous individual recommendation approaches; • Approaches combination leads to improvements in prediction performance.	• Hard to compare recommendation approaches; • Increased implementation complexity; • Difficult to provide a recommendation explanation.
Trust-Based Recommendation	• Alleviation of data sparsity and cold-start; • Increase recommendation coverage and predictive accuracy based on the number of users.	• Limited for the new item cold-start problem; • Accuracy can decrease depending on the number of connections to the source user.

main advantage is the improvement of the prediction performance (B. Thorat et al., 2015). Despite the popularity of this kind of technique, it presents limitations regarding data sparsity and cold-start problems and scalability, requiring considerable computational power to make recommendations for big datasets (B. Thorat et al., 2015; Çano and Morisio, 2017).

The CBF approach presents some advantages relative to the CF approach, such as the ability to make recommendations even with no available ratings, to adjust the recommendations shortly after the change of the user's preferences, and to provide explanations on how the recommendations were generated (Isinkaye et al., 2015; B. Thorat et al., 2015). This approach does not suffer from new items cold-start since the recommendations are performed based on the items' descriptions and not on their user ratings. On the other hand, this technique requires a detailed description of item features, and it has difficulties performing recommendations when the users vary their preferences in a short time. This technique often suffers from the new user cold-start problem since it is challenging to perform the first recommendations accurately. The CBF approach restricts the recommendations since the approach promotes content over specialisation, focusing the recommendations on the preferred content (B. Thorat et al., 2015).

The combination of two recommendation approaches, in the HBR approach, enables the improvement of the recommendation process's accuracy and efficiency by overcoming the combined techniques' problems such as cold-start, over specialisation and data sparsity (B. Thorat et al., 2015). Despite the advantages of combining the different approaches, comparing the used recommendation techniques is complex, the complexity of implementation increases, and the recommendation explanation is problematic.

In TBR approaches, the combination of similarity and trust between users improves the recommendation accuracy (Isinkaye et al., 2015) and coverage, which means that the system will consider the entire items list in the recommendation process (Jamali and Ester, 2009). This can lead to an alleviation of the data sparsity and cold-start problems presented in the CF techniques. As an example, in O'Donovan and Smyth, 2005, the use of trust information is incorporated into the recommendation process, demonstrating a positive impact on the recommendation quality. However, TBR is limited by the definition of a social trust network between users and for the new item cold-start problem. The trust between users is also a limitation, decreasing the accuracy depending on the number of connections of the source user used for the trust calculation. There are also several open research challenges involving the trust theme, such as the alleviation of the trust-based cold-start problem, visualisation of the trust-enhanced recommendation system, theoretical foundations for trust-based research, and introduction of distrust in the recommendation process (Victor et al., 2011).

3. Trust-based recommendation system for what-if digital twin

This paper proposes a trust-based recommendation approach combined with an RL technique that promotes learning from user feedback and introduces similarity measures. With this combination, the proposed model is potentially able to handle cold-start and data sparsity problems and allows the integration of user trust into the digital twin decision-support framework.

3.1. What-if digital twin architecture

The digital twin architecture considers integrating a trust-based recommendation system with what-if simulation models. The architecture is composed of five layers: *Communication, Data Analysis, Simulation, Decision-Support*, and *Human Trust* (Pires et al., 2021a).

Briefly, the *Communication Layer* provides communication protocols to enable the connectivity between the physical and virtual dimensions, and the *Data Analysis Layer* transforms the collected data into information that the digital twin can understand. By combining real-time monitoring and prediction, it is possible to extract important knowledge to trigger a deeper analysis to improve the physical system's behaviour. This analysis is performed by the *Simulation Layer*, which uses a what-if simulation approach to run different simulation scenarios, generated from previously defined degrees of freedom (DoFs), in the virtual model to identify improvement opportunities. The simulation results serve as input for the AI-based recommendation algorithm placed in the *Decision-Support Layer*, which provides support to the users in the decision-making process. The generated recommendations pass through an explanation node, which translates the generated recommendations into a form that can be understood and presented to the users. The users then decide if the recommendation is appropriate for implementation in the physical system and provide the correspondent feedback to the system.

Lastly, the *Human Trust Layer* aims to assess and determine the user's trust level in the digital twin-based recommendations. The assessment of user trust level is based on a trust model that considers the user trust in the recommendation and the system and the acceptability of implementing the recommendation in the physical world. To address the cold-start problem, the trust model is complemented by cold-start strategies (i.e., scenario similarity, user similarity, and user reputation), which allow for mitigating the lack of initial data regarding the trust rating feedback. The user trust data is used as input for the recommendation system embedded in the *Decision-Support Layer*.

3.2. Trust-model formalisation

The proposed model aims to overcome the limitations of traditional models, such as data sparsity and cold-start, and improve the system's performance by integrating the trust model with an AI-based algorithm and similarity metrics. The term "item" is typically used in the recommendation systems for object or news recommendations, but in this paper, the term “scenario” will be used. Fig. 1 illustrates the trust-based recommendation system, named *SimQL*, that joins the what-if simulation model with the trust-based model, following the principles of a hybrid recommendation approach.

The *SimQL* model uses the following three inputs, namely:

- User trust in the given scenario recommendation (UT_R), which can be measured by the feedback in the form of a rating given by the user;
- User trust in the recommendation system itself (UT_S), which is set initially by the user and continuously updated given the accuracy of the recommendation of the system;
- User social trust in the work network, in which each user can give a trust score to another user depending on a set of work-related factors to calculate the user’s reputation.

In Fig. 1 are presented five labels that will help to understand the steps taken within the recommendation cycle. The recommendation starts its cycle on Label 1, were the user defines the DoFs for the scenarios to be simulated on using what-if simulation based on the virtual model of the physical system (if necessary, the set of scenarios can be subjected to scenario reduction techniques). The simulation is performed on Label 2 and the simulation results are used as input for the RL algorithm (Label 3) that learns from them and from the user feedback to perform the individual recommendations for one user at a time. Based on the given recommendations, the user will provide feedback (see Label 4 in Fig. 1) in the form of a user trust rating (A_R) and its intention of applying the provided scenario recommendation (U_{Acc}). This information is sent to the trust model, which is updated and transforms the data into a user-scenario trust matrix, scenario-DoFs matrix, and a user-acceptability matrix. This information is used to calculate the user’s trust in the system. Depending on the case, the scenario similarity, user similarity, and user reputation can be used to calculate the predicted user trust as a rating (E_R). Based on this new information, the RL algorithm calculates the reward and adjusts its recommendations accordingly to user preferences (Label 5). The trust model also considers a social trust network that is the basis for calculating the user reputation.

3.2.1. Reinforcement learning algorithm

The *SimQL* uses an RL algorithm, namely Q-Learning (Sutton and Barto, 1998), based on a Q-table and Q-function. The Q-table represents the relationship of q-values ($Q(s_t, a_t) \equiv Q(s, a)$) between the trust state of the user ($s_t \equiv s$) and the actions ($a_t \equiv a$) represented by all the possible scenarios to be recommended. The Q-learning is a model-free reinforcement algorithm that is used to learn the optimal action-selection policy for a given environment. The Q-learning algorithm can be seen as a Markov Decision Process (MDP), where the states (s) are the states of the environment that belong to a state space ($s \in S$), defined as all the possible trust states, the actions (a) are the actions taken by the agent belong to an action space ($a \in A$), defined as the possible scenarios to be recommended, the transition probabilities (P) are given by the environment through a transition function specifying the probability of transitioning to a new state (s') given the current state and action, which can be represented by $P(s'|s,a)$, and the reward ($r_t \equiv r$) is given by the reward function (R) of the environment, this function assigns a real value r to each state-action pair $(s,a) : R(s,a)$. The goal of the algorithm is to find the optimal policy, which is the action selection policy that maximizes the expected future reward.

The Q-learning algorithm works by iteratively updating the Q-function according to the Bellman equation (Eq. (1)),

$$Q(s, a) = R(s, a) + \gamma \times \max_{a'} Q(s', a') \tag{1}$$

where s' is the next state, a' is the action taken in s', $R(s, a)$ is the reward for taking action a in state s, and γ is the discount factor, where $0 \leq \gamma \leq 1$, determining the importance of future rewards compared to current rewards. The goal of the Q-learning is to learn a function Q(s,a), which gives the expected reward for taking action a in states s, and following the optimal policy thereafter. In Fig. 2 is illustrated a block diagram of the MDP RL algorithm.

As is possible to observe in the figure above, the RL algorithm consists of an environment that represents the outside world, an agent that has, in this case, the Q-Learning algorithm receiving states (S_t) and performing actions (A_t) according to an established policy, the actions receive rewards (R_{t-1}) by the operators (O_t) present in the environment. The agent and the environment interact over a sequence of discrete-time steps.

When there is no historical data, the initial state of the recommendation relies on establishing the initial values of the Q-table as random and performing offline training through the performance of random actions. The training should be done through episodes and by maximising the total reward of each episode. After this, the system recommends the best scenarios and E_R. The user-scenario trust matrix, user acceptability matrix, and UT_S are updated based on the user feedback and acceptability, and the r_t is calculated. These calculations will update the Q-table for the active user[1] in iteration 1 to N (it is important to note that each user has its Q-table, which contains the q-values for each scenario).

3.2.2. Reward calculation

After this initial state, the Q-tables are already filled with values for the existing scenarios, being possible to apply the described approach to calculate the reward (r_t) value for each scenario recommendation, which is the output of the trust model, calculated according to Eq. (2).

$$r_t = W_1 \times UT_R + W_2 \times U_{Acc} + W_3 \times UT_S \tag{2}$$

This equation aims to reward trustworthy scenarios and penalise untrustworthy scenarios, calculated by a multicriteria function where the three components, which should be normalised, are weighted with W_1, W_2 and W_3 according to the system properties. The first component is the UT_R, which ranges from $[-V_{\min}, V_{\max}]$; note that this scale is symmetrical, which means that absolute values for $V_{\min}$ and $V_{\max}$ are the same. The second component is the U_{Acc}, and the last component is the UT_S. The reward value is sent to the RL algorithm, updating it for future recommendations.

For the first recommendation, the values for the reward calculation are provided by the user. However, for the next iterations, the UT_S value is updated according to the performance of the recommendation system. From the evaluation of the recommended scenarios, the $UT_R = A_R$ and U_{Acc} are obtained, and from the system, the E_R value is calculated, which is sent to the trust model. The user trust in the system, UT_S, is calculated by the following equation (Eq. (3)),

$$UT_S = \begin{cases} C_{UT} - \dfrac{|E_R - A_R|}{|E_R|}, & E_R > A_R \vee E_R < A_R \\ C_{UT} + \theta, & E_R = A_R \end{cases} \tag{3}$$

where the C_{UT} is the current value of the user trust in the system, which is established initially by the user and continuously updated at the end of each iteration ($C_{UT} = UT_S$), the value of θ represents a positive value to be defined by the user, determining how much a correct rating prediction gave the user trust in the system.

3.2.3. Recommendation module

A recommendation module was defined for the proposed model with the recommendations being performed based on a recommendation value, R_{Value}, calculated by the Eqs. (4) and (5). For both equations a_t, represents the possible recommended scenario.

[1] Active user refers to the user currently using the recommendation system.

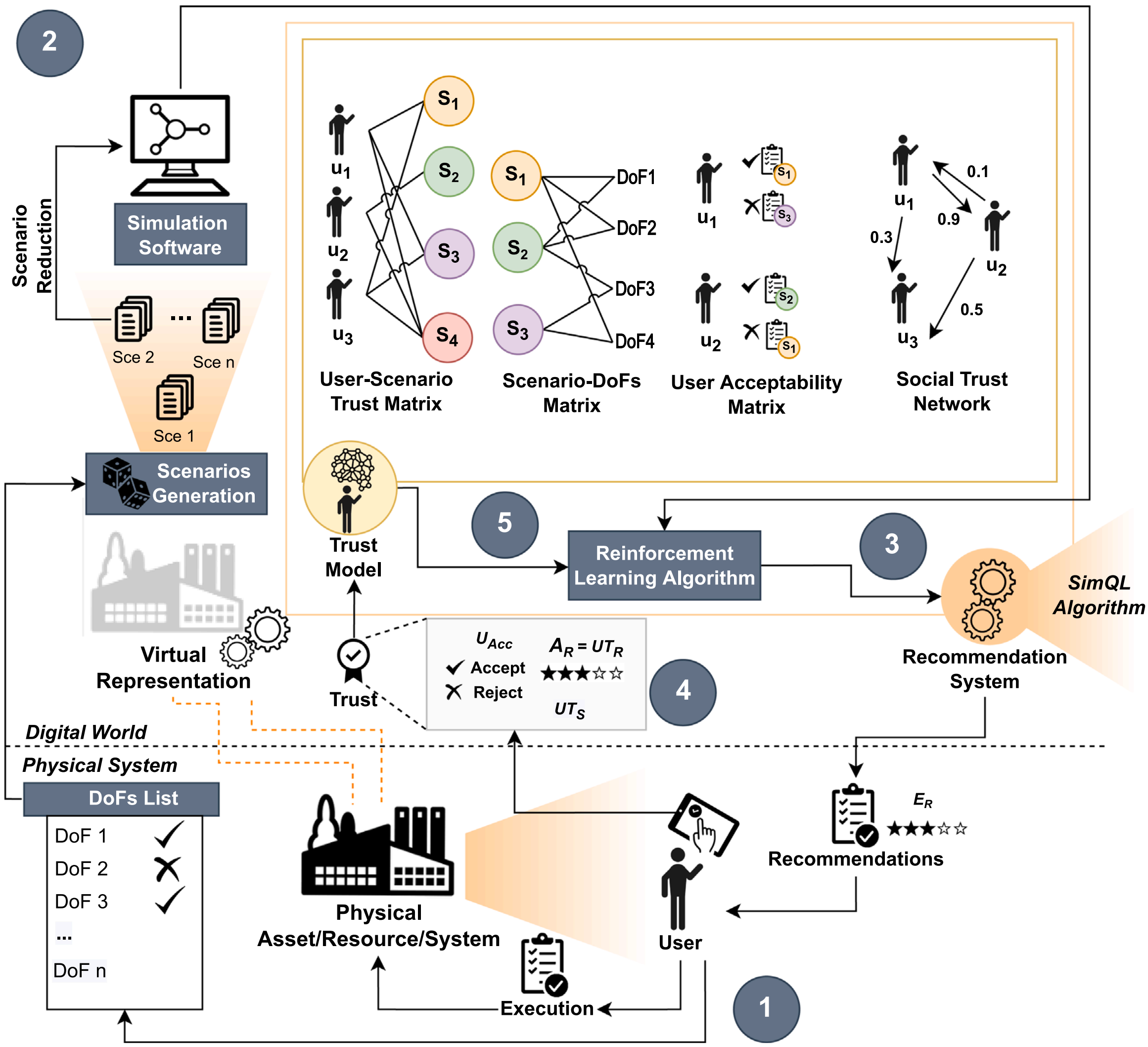

Fig. 1. What-if simulation model and trust model structure.

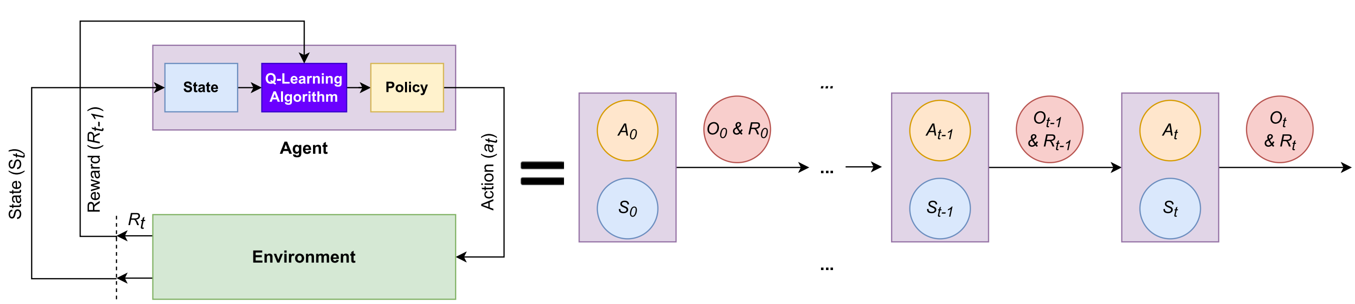

Fig. 2. Block diagram of the RL algorithm MDP (Based on Geravanchizadeh and Roushan, 2021).

$$Trust_P = \frac{|Q(3, a_t) + Q(2, a_t) + Q(1, a_t)|}{3} \tag{4}$$

This translates into an average value for positive trust values assigned to a specific scenario.

$$Trust_N = \frac{|Q(0, a_t) + Q(-1, a_t) + Q(-2, a_t) + Q(-3, a_t)|}{4} \tag{5}$$

This translates into an average value for negative trust values assigned to a specific scenario. The $Q(s_t, a_t)$ variable represents the q-value for the s_t rating state, positive/negative trust rating values, for the recommended scenario, a_t. The R_{Value} is calculated based on Eq. (6).

$$R_{Value} = Trust_P - Trust_N \tag{6}$$

which intends to penalise the negative trust values, also known as untrustworthy behaviour. The output of the recommendation model is a list of test scenarios ordered according to R_{Value} parameters.

3.2.4. Rating the prediction calculation

The recommendation module considers whether the active user has already rated the recommended scenarios or not, implying different forms of performing the calculation of the E_R.

The calculation of the expected rating or rating prediction, E_R, is dynamically changed according to the cases presented to the recommendation system. The different variants of calculating E_R are visually separated in Fig. 3, in the following text are going to be described in detail the different cases and equations.

If there is enough information regarding user and scenario trust rating, the rating prediction is calculated by Eq. (7).

$$E_R = \frac{\bar{U_{Acc}}(u) + \bar{UT_R}(a_t)}{\bar{UT_R}(u)} \tag{7}$$

This equation considers the average user acceptability, with the average trust rating of the scenario divided by the average user trust rating of the active user. Significant measures are introduced when there is no historical information regarding new scenarios or new users. A scenario can be defined as a new scenario in two situations, when this has never been rated by a specific user, being new to that user, or when any user has not rated it.

In the case of a new scenario that any user in the system never rated, scenario similarity is used to get the test data from a similar scenario. For this purpose, the scenario similarity, $sim(a_t, a_j)$ is calculated by using the cosine similarity function based on the DoFs values for the tested scenarios through Eq. (8). It should be considered that a_t represents the recommended scenario, and a_j represents the other scenarios that already were recommended. For this equation is also taken in consideration a variable k that represents the unique identifier for each DoFs involved in the calculation going from 1 to m.

$$sim(a_t, a_j) = \begin{cases} \dfrac{\sum_{k=1}^{m} V_{DoF(a_t,k)} \cdot V_{DoF(a_j,k)}}{\sqrt{\sum_{k=1}^{m} V^2 DoF(a_j,k)} \cdot \sqrt{\sum_{k=1}^{m} V^2 DoF(a_j,k)}}, & N_{DoF(a_t)} = N_{DoF(a_j)} \\ 0, & N_{DoF(a_t)} \neq N_{DoF(a_j)} \end{cases} \tag{8}$$

which uses the DoF value ($V_{DoF(a,k)}$) for each scenario, is only calculated if the number of DoFs ($N_{DoF(a)}$) are the same for both scenarios, and they are correlated; otherwise the similarity is zero. In this case, the q-value to be used in the Q-Table for the recommendation calculation will be the q-value from the most similar scenario rated by the active user. For the rating prediction, if the scenario similarity value is greater than 0.5, the $E_R$is calculated using Eq. (9).

$$E_R = \bar{UT_R}(u) + \frac{(\bar{UT_R}(u, a_t) + \bar{UT_R}(u, s_j)) \times sim(a_t, a_j)}{sim(a_t, a_j)} \tag{9}$$

which considers the average rating trust of the user, $\bar{UT_R}(u)$, plus the average trust of the user in scenario t, $\bar{UT_R}(u, a_t)$, and the average trust of the user in the most similar scenario $\bar{UT_R}(u, a_j)$.

In the case of a new user, user similarity $sim(u, v)$ is used by the recommendation system to recommend a scenario, given that other users have already rated the scenario. The Pearson Correlation Coefficient (PCC) equation performs the user similarity calculation using Eq. (10).

$$sim(u, v) = \frac{\sum_{i}^{n} (UT_R(u, a_t) - \bar{UT_R}(u)) \cdot (\bar{UT_R}(v, a_t) - \bar{UT_R}(v))}{\sqrt{\sum_{1}^{n} (UT_R(u, a_t) - \bar{UT_R}(u))^2 \cdot \sum_{i}^{n} (UT_R(v, a_t) - \bar{UT_R}(v))^2 + C}} \tag{10}$$

In this equation, the user bias is minimised using the PCC, and the support problem is also minimised by adding the shrinking term. The user bias problem can be defined as the fact that some users give a higher rating than others, favouring some scenarios over others. The support problem can be defined as the balance between having the similarity calculated based on a few or a large amount of data and normalising the similarity value. In this case, the q-values used by the recommendation module to build the top-N list are from the most similar user to the active user.

If the user similarity value is greater than 0.5, the rating prediction is calculated using the Eq. (11).

$$E_R = \bar{UT_R}(u) + \frac{(UT_R(u, a_i) - \bar{UT_R}(u)) \times sim(u, v)}{sim(u, v)} \tag{11}$$

which considers the average rating of the user trust, $\bar{UT_R}(u)$, and how much the user trusts that the scenario will work in the physical system $(UT_R(u, a_t) - \bar{UT_R}(u))$.

A specific case relies upon when the user similarity values are the same and greater than 0.5, for which a tie-breaking measure is applied by considering the user reputations, $rep(u)$. The user reputation can be defined as how much the other users trust the active user and if the user always gives a fair trust measure to scenarios (Song et al., 2017). The user reputation is calculated using the Eq. (12).

$$rep(u) = \frac{\sum_{a_t \in S(u)} |UT_R(u, a_t) - \bar{UT_R}(a_t)|}{S(u)} + \frac{\sum_{u \in U(u)} |Trust_{(u,v)} - \bar{Trust_u}|}{U(u)} \tag{12}$$

which considers a weighted average of the UT_R as ratings given by the user to a set of scenarios and the trust that the other users have in the active user.

The trust between users is calculated based on a user social trust network built on users' trust connections and weights representing the user's trust in the other user. Fig. 4 illustrates an example of a social trust network.

The calculation of the trust between users, $Trust_{(u_i,u_j)}$ is a combination of direct and indirect trust (Chen et al., 2021), calculated according to Eq. (13).

$$Trust_{(u_i,u_j)} = \begin{cases} Dtrust_{(u_i,u_j)}, & Dtrust_{(u_i,u_j)} \neq 0 \\ Itrust_{(u_i,u_j)}, & Dtrust_{(u_i,u_j)} = 0, \quad Itrust_{(u_i,u_j)} \neq 0 \\ 0, & Dtrust_{(u_i,u_j)} = 0, \quad Itrust_{(u_i,u_j)} = 0 \end{cases} \tag{13}$$

The direct trust calculation, *Dtrust*, can be performed using the social trust network, which has weighted paths between users. The direct trust is calculated by Eq. (14).

$$Dtrust_{(u_i,u_j)} = W^k_{direct} \tag{14}$$

The indirect trust, *Itrust*, is calculated by using W^k_{direct}, which represents the trust value before user u_i reaches the user u_j, and also by using the $W(k)$ that represents the weight of k path that indirectly connects the users. $W(k)$ is calculated by multiplying the direct weight of the path according to Eq. (15).

$$W(k) = \prod_{i=1}^{l-1} Dtrust_i(x, y) \tag{15}$$

The indirect trust is calculated according to Eq. (16).

$$Itrust_{(u_i,u_j)} = \frac{\sum_{k=1}^{n} (W(k) \times W^k_{direct})}{\sum_{k=1}^{n} W(k)} \tag{16}$$

The trust value between users is calculated and provided for calculating the user's reputation through the presented formulas. The recommendation module uses the q-values of the user with a higher reputation towards the active user. In calculating the rating prediction using the user reputation, the E_R is calculated by Eq. (17).

$$E_R = \bar{UT_R}(u) + \frac{(UT_R(u, a_i) - \bar{UT_R}(u)) \times rep(u)}{rep(u)} \tag{17}$$

In this case, instead of using the similarity to calculate the rating prediction, the $rep(u)$ of the user with the higher value is used,

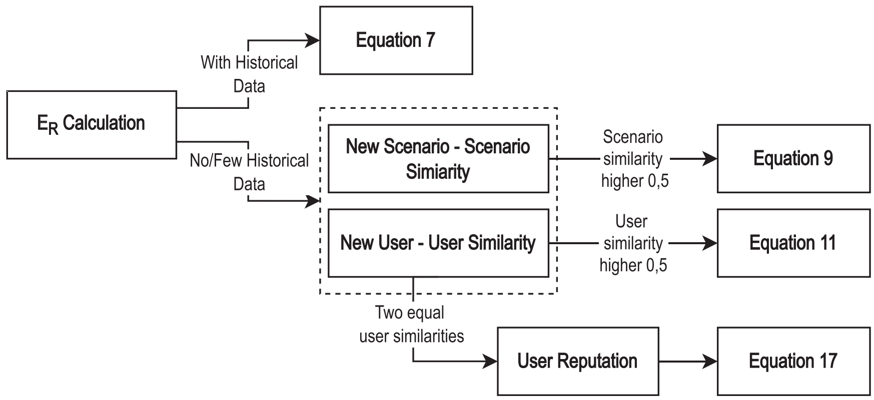

Fig. 3. Calculating E_R according to the different scenarios.

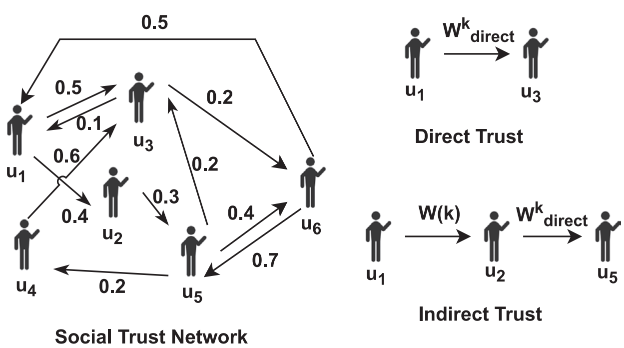

Fig. 4. Social trust network, and direct and indirect trust.

considering the average rating trust and how much the user trusts in that scenario.

Summarizing, the main innovative aspects associated with the proposed *SimQL* algorithm are the combination of an RL algorithm with similarity and trust measures to minimize the effects of cold-start and data sparsity problems and the different forms of calculating the predicted trust rating to improve the predicting rating calculation accuracy.

4. Experimental validation

The proposed approach was experimentally implemented and applied to a manufacturing case study aiming to validate its benefits and performance. The chosen case study enables the creation of different testing scenarios with different characteristics directly linked with cold-start and data sparsity problems, such as having several users and scenarios, new users and new scenarios with no historical information. A simple version of the recommendation model, hereafter called *QL*, was also implemented to serve as a comparison. The *QL* algorithm only considers the Q-learning algorithm reporting only on the user trust rate and acceptability and does not consider any similarity measures or different predicting rating calculation equations. This model was defined as an intermediate approach, with the purpose of highlighting the benefits of applying similarity measures to address the cold-start and data sparsity problems.

4.1. Description of the case study

The case study is based on a battery pack assembly line called IML (Integrated Manufacturing & Logistics), installed at WMG, University of Warwick. The battery pack assembly line consists of two semi-lines, one that pre- processes the materials and another responsible for the welding processes. Different AGVs (Autonomous Guided Vehicles) serve each semi-line for transporting the parts. The physical AGVs are MiR100, which have an average of 10 hours running time (or 20 km), reaching a maximum speed of 1.5 m/s forward and 0.3 m/s backwards. These are powered by a battery (Li-NMC, 24 V, 40Ah), taking around 4.5 hours to charge fully. The work in the assembly line follows a task sequence, where the operator requests an AGV to transport parts to the preprocessing stations in the first semi-line. The AGV travels to these stations and unloads the material to be processed. After each preprocessing station has finished, AGVs are called to transport the parts to the welding stations in the second semi-line. Lastly, the parts are sent to the storage station.

As a representation of the physical system in the digital twin, a discrete event simulation (DES) model was developed using the FlexSim software, illustrated in Fig. 5.

The DoFs in the model are the number of available AGVs per semi-line, varying between 1 and 3, and the recharge threshold[2], ranging from 30% to 70% with increments of 10%, the resume threshold[3], ranging from 40% to 90% of battery level with increments of 10%, and the shift duration (8, 16, and 24 h). Considering these DoFs, 540 alternative scenarios (configurations) for the system operation are available to be simulated.

The proposed trust-based recommendation approach is used to recommend the best logistical scenario from the set of generated scenarios. The scenarios are evaluated in terms of the best number of AGVs operating in the line considering the analysis of the results from the simulation, the user trust rating, the trust history in the recommendation system, and the user social trust network. Several advantages of applying the proposed approach to the specified case study can be highlighted, namely a faster identification of the best scenarios allowing for a near real-time intervention, a selection of the best scenarios not only based on the simulation results but also on the human knowledge of the system, and the capability of handling problems as cold-start and data sparsity which are recurrent in recommendation environments.

4.2. Evaluation methodology

The main purpose of the proposed trust-based recommendation system is to recommend the best scenarios according to the individual users' trust rating profile. The proposed approach was evaluated by comparing its rating prediction accuracy with the state-of-the-art recommendation systems approaches. The created datasets used to support this evaluation, represented in Table 2, were divided into sub-datasets with a different number of users, scenarios, density, and sparsity levels, and it was applied 5-fold cross-validation in the validation phase of the model.

These three datasets include users' feedback for different scenarios

[2] % of battery level triggers the AGV to a charging point.
[3] % of battery level that should be reached during a changing operation.

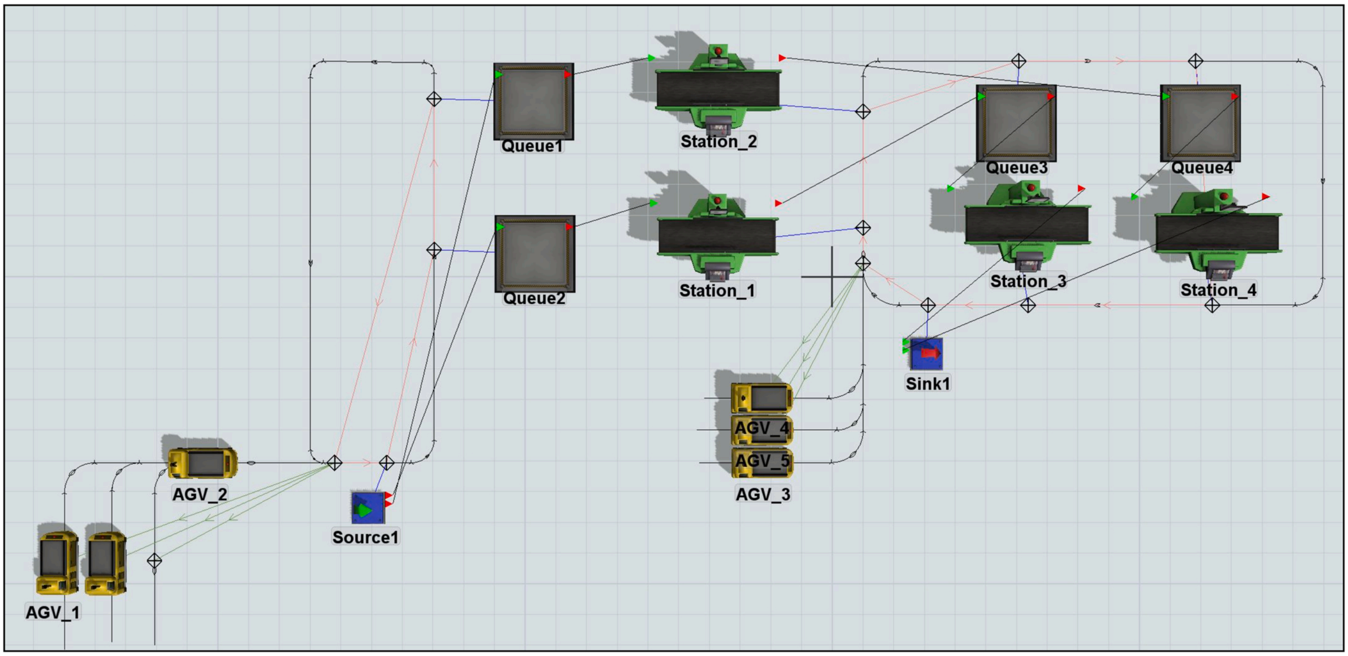

Fig. 5. DES model of the battery pack assembly line.

Table 2
Characterisation of the experimental datasets.

Dataset	Users	Scenarios	Ratings	Avg. nº Ratings/User	Avg. nº Ratings/Scenario	Density	Sparsity
#1	2	23	31	15,50	1,35	67,39%	32,61%
#2	3	26	48	16,00	1,85	61,54%	38,46%
#3	6	47	56	9,33	1,19	19,86%	80,14%

modelled according to a specific user-bot mimicking a different rating profile for each user. The datasets were constituted this way for the proposed model to be evaluated for different levels of sparsity and the number of cold-start users and scenarios. In order to attain different sparsity levels, it was necessary to establish datasets with different numbers of users, scenarios and ratings. The initial dataset had 540 scenarios, which at the time of recommendation, would become computationally heavy and time-consuming. The three datasets are the result of having subjected the original dataset to scenario reduction techniques based on the simulation results. Since it is an industrial environment, the presence of few users is a recurrent variable in these systems, being necessary for a recommendation system capable of working with a small dataset with few users and rating information.

The density level is ratio between the number of actual ratings (Act_R) and the number of possible ratings (Pos_R) that can be calculated by multiplying the number of users by the number of scenarios. The equation to calculate the density is illustrated in Eq. (18).

$$Density = \frac{Act_R}{Pos_R} \times 100 \tag{18}$$

The sparsity level can be calculated based on the density, as the sum of the two has to be 100%, being $Sparsity = 100 - Density$.

The experimental validation metrics usually used to evaluate the predictive rating accuracy for a recommendation system are the mean absolute error (MAE) and the root mean squared error (RMSE). The MAE quantifies the average distance between the scenario's actual rating and the predicted rating, being more accurate in predicting the user ratings lower the MAE value is. This metric can be computed as follows (Eq. (19)),

$$MAE = \frac{1}{N}\sum_{u,a}|p_{u,a} - r_{u,a}| \tag{19}$$

where N is the total number of ratings on the scenario set, $p_{u,a}$ is the predicted rating of user u on scenario a, and $r_{u,a}$ is the actual scenario rating. The RMSE measure emphasises the more significant absolute error, with a lower value indicating a better recommendation accuracy. The calculation of this metric is performed as follows (Eq. (20)),

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,a}|p_{u,a} - r_{u,a}|^2} \tag{20}$$

where N denotes the total number of predictions.

The *SimQL* and the *QL* approaches were implemented in Python, and compared with the state-of-the-art recommendation models, namely two merely based on ratings models, *UserCF* (Resnick et al., 1994) and *SVD++* (Koren, 2010), three early TBR models, *SocialRec* (Ma et al., 2008), *SocialRSTE* (Ma et al., 2009) and *SocialReg* (Ma et al., 2011), and three latest state-of-the-art TBR models, *SocialMF* (Jamali and Ester, 2010), *TrustWalker* (Jamali and Ester, 2009), and *TrustSVD* (Guo et al., 2015). A brief description of each model is provided in Section 2 and more details can be found in the respective references. These methods were implemented following the implementation provided by Zhang et al. (2018), using an Intel Core M-5Y71 1.20 GHz CPU with 8 GB RAM to run all the approaches on a Windows 10 Pro system.

4.3. Analysis of experimental results

The proposed *SimQL* model was experimental validated through the performance of a comparison study with the state-of-the-art recommendation models. Table 3 summarises the achieved MAE and RMSE results for the experimental tests considering the different approaches and the three datasets. Note that the 'Deviation' parameter indicates the improvement of performance of the *SimQL* approach relative to the analysed model, which is calculated through the Eq. (21).

$$Deviation = \frac{V_M - V_{SimQL}}{V_M} \times 100 \tag{21}$$

where the V_{SimQL} represents the RMSE or MAE value of the *SimQL* model, and V_M is the value for the model to be compared.

Considering the results presented in Table 3, each dataset's best and worst models are identified with the RMSE values in bold. Regarding specifically the state-of-the-art models and the recommendation accuracy, for dataset #1, the *TrustWalker* was the best-performing model with an RSME of 2,453. This method uses the rating prediction of the user-scenario rating information and the user trust data from which a trust network is built. One of the reasons this model is the best-performing model is that it performs simulated random walks on the trust network for each user, collecting information about the direct and indirect relationships between users. The use of trust measures in recommendation systems has a direct correlation with improvement in terms of performance. In this case, dataset #1 presentes only two users, making the trust network less extensive, consequently improving the model's accuracy since the model loses accuracy with the increase in the size of the trust network. In dataset #2, the best-performing model was *SocialRec,* with an RMSE of 2,469. One of the main reasons for this is that the model uses social influence in the recommendation process, considering the users' individual and friends' preferences. This is achieved by fusing the user-rating matrix with the user's social network using a probabilistic matrix factorisation. The *TrustSVD* model was the best-performing method for dataset #3, with an RMSE of 1,918. Although the implicit information about the user-scenario rating is not present in this dataset or the other two datasets, this method outperforms the other methods. Despite the lack of explicit and implicit information by the *TrustSVD* model, this method also uses a weighted regularisation technique to avoid the over-fitting of the model learning and the user trust for the rating prediction, and also uses the trust network of users considering only the direct relationships between the users. Although these methods are all social recommendation models, there are fundamental differences between them and how they perform recommendations, which results in different best-performing models for each dataset with different conditions.

In terms of worst methods, in dataset #1, the *SVD++* model was the worst performing method, with an RMSE of 3,663. For this model to work well requires a large dataset with a large set of implicit user-scenario rating information, since this information is the basic information used for the rating prediction calculation in this model. This model is also known by working better under sparse data, which does not happen in this dataset (sparsity = 32,61%). For the dataset #2, the worst method was the *SocialMF* with an RMSE of 2,917, which uses the matrix factorisation-based model for recommendation in social rating networks, incorporating trust propagation. This model needs relevant and a large amount of social information to be able to efficiently incorporate it into the recommendations, which in a dataset with three users may be difficult. One of the main characteristics of the method is its ability to reduce the RMSE significantly for cold-start users, which in dataset #2 is not present since there is a rate of 61,54% of dataset density. In dataset #3, the worst performing method is the *SocialRSTE* with an RMSE of 2,574, which uses a factorised user-scenario rating matrix and the social trust network then applied in a probabilistic framework and gradient descent objective function. One of the possible reasons for this method being the worst is the fact that this performs best in very large datasets, and since the approach scales linearly with the number of observations, the provided dataset is of a small dimension. All the recommendation models performance can be influenced by the quality, and quantity of the user-item interactions, and social information available.

In a high-level analysis, on average, the social recommendation models perform better than the traditional recommendation methods. The differences in performance start to decrease with the increase in the dataset size and number of users, scenarios and ratings. From the results presented in the table, the proposed *SimQL* consistently outperforms the tested state-of-the-art methods presenting a stable performance throughout the three datasets in terms of RMSE and MAE. For the dataset #1 the model improves 37,92% regarding the *TrustWalker*, for dataset #2 improves 30,28% relatively to *SocialRec*, and for dataset #3 improves 18,54% regarding the *TrustSVD*. In summary, the *SimQL* model presents a high percentage of performance improvement in all datasets and can alleviate the problems of data sparsity and cold-start users/scenarios. The main difference between the *SimQL* model and the other social recommendation models is the combination of an RL algorithm with similarity measures, social trust data for performing recommendations, and a dynamic system for rating prediction calculation.

After comparing the proposed model *SimQL* model with the state-of-the-art models, another study was conducted comparing *SimQL* with the *QL* model. Bearing in mind that the proposed approach is an iterative method, an experiment was performed for which the dataset ran continuously, performing recommendations for one user at a time and

Table 3
Comparison of the performance of *SimQL* model with the state-of-the-art recommendation models.

	Model	Dataset #1		Dataset #2		Dataset #3	
		RMSE Deviation	MAE Deviation	RMSE Deviation	MAE Deviation	RMSE Deviation	MAE Deviation
Proposed Model	***SimQL***	**1,523**	**1,004**	**1,734**	**1,295**	**1,562**	**1,367**
Traditional	*UserCF*	3,569	2,989	2,565	1,932	2,240	1,639
		57,33%	56,66%	32,90%	32,96%	30,24%	16,63%
	SVD++	**3,663**	3,117	2,734	2,238	2,474	2,066
		58,42%	56,14%	37,04%	42,12%	36,85%	33,87%
Social	*SocialRec*	3,031	2,504	**2,469**	1,939	2,056	1,603
		49,76%	48,27%	**30,28%**	33,19%	24,00%	14,74%
	SocialRSTE	3,057	2,724	2,882	2,574	2,574	2,275
		50,18%	52,45%	40,28%	49,68%	39,30%	39,93%
	SocialMF	3,637	3,123	**2,917**	2,468	2,502	2,118
		58,13%	58,53%	**41,00%**	47,52%	37,54%	35,49%
	SocialReg	3,191	2,697	2,695	2,200	2,212	1,814
		52,27%	51,97%	36,13%	41,12%	29,38%	24,65%
	TrustWalker	**2,453**	2,113	2,695	2,298	2,484	2,103
		37,92%	38,70%	36,13%	43,62%	37,10%	35,00%
	TrustSVD	2,821	2,428	2,560	2,067	**1,918**	1,498
		46,02%	46,64%	32,76%	37,35%	**18,54%**	8,80%

observing the performance of the models in terms of rating prediction accuracy. Fig. 6 illustrates the results of the RMSE for the *SimQL* and *QL* models in twenty-five iterations and considering the dataset #1 and #3, the datasets with the lower and higher levels of sparsity, and presenting the lower and higher number of cold-start users/scenarios. This allows to compare on how the integration of the similarity measures and the dynamic predicting of trust rating changes the performance of the proposed algorithm.

In the RMSE graph for the dataset #1, it is possible to observe a significant increase in the RMSE value at the fifth and from the ninth to the twelfth iterations. In the fifth iteration, this is due to the first change of user, introducing a cold-start user with no previous rating history and starting to rate never rated scenarios. The system recurs to the scenario similarity measure, but the scenario may be too different, influencing a less accurate prediction. The first scenario, already rated by another user, is introduced in the ninth iteration. The user similarity is calculated, and since there is little historical information for both users, similarity calculation may not be very accurate, changing the following predictions based on this result, which can be classified as an outlier. The significant increase in the RMSE measure may be due to its susceptibility to outliers. In the twentieth iteration, there is again a user similarity calculation, but now with more historical information, which translates into an insignificant increase in the RMSE.

In the dataset #3 graph, the introduction of the users is performed until the twelfth iteration, which means that the algorithm will have more basic information to perform the recommendations. The increases in the RMSE from the third to the sixth iteration may be due to very different new scenarios from the scenarios with historic rating information. In the eighth iteration, user 3 is rating a scenario already rated by user 1, applying the user similarity and significantly increasing the RMSE.

For dataset #1, the RMSE of the *QL* model is significantly higher than the *SimQL* model, which means that the *SimQL* model performs better than the *QL*. Considering the performance of both models, the *SimQL* outperforms the *QL* model in the two situations analysed. Implementing the similarity, reputation and trust measures within an RL algorithm for recommendation significantly contributes to handling cold-start users/scenarios and data sparsity problems. Particularly, *SimQL* model, on average, performs better in a dataset with more users and scenarios and can handle the data sparsity problem without sacrificing the performance.

Regarding the datasets #1 and #3, there are significant differences in their constitution, with dataset #1 having only two users and dataset #3 having six users; the number of scenarios to be evaluated is also different, being 23 and 47 respectively, and consequently, the sparsity levels are also quite different, 32% and 80%. These differences are also noted in the evolution of the RMSE over the iterations, and for dataset #3, the values of the *SimQL* model stabilise earlier. Although the dataset #3 has more users, which means it is more likely to be subjected to cold-start users, and the sparsity level is higher, the fact that all users are introduced early in the recommendation cycle until the twelfth iteration makes the system more stable and on a path of continuous improvement.

5. Conclusions and future work

In manufacturing, digital twin technology can simulate the performance and behaviour of physical systems, such as production lines, equipment, and machinery. Through the combination of digital twin technology with a recommendation system, it is possible to model a manufacturing system's performance and make recommendations for how to improve it. However, there needs to be integrated approaches capable of learning and improving user trust in the decision-support system to overcome the traditional challenges that recommendation systems face, such as cold-start and data sparsity problems.

This paper proposes an innovative digital twin decision support framework that integrates recommendation systems with the RL algorithm, trust and similarity measures, aiming to provide more accurate and reliable recommendations that can improve efficiency, reduce downtime and increase production output, contributing to optimising the manufacturing process. For this purpose, the *SimQL* model was formally specified to combine a trust model, an RL algorithm and similarity measures in an effective solution for addressing the common problems usually associated with recommendation systems, namely the user trust in the recommendation system, the time for making a decision, and the cold-start and data sparsity problems.

The proposed model was experimentally validated in a manufacturing case study of a battery pack assembly line. Initially, the *SimQL* model was compared with some state-of-the-art recommendation models, being possible to verify the effectiveness of the proposed solution that outperforms the existing models in terms of accuracy. In the second set of experiments, it was assessed if the integration of trust and similarity measures would impact the performance of the *SimQL* model, for which it was compared with the *QL* model that does not implement any trust or similarity measures. The results showed that the integration of these measures has a positive impact on the performance of the proposed solution, improving its performance and accuracy.

Future work will be devoted to further improving the recommendation model by incorporating tie-breaking measures for a recommendation scenario with equal scenario similarities and including an exploration module that enables the recommendation of scenarios outside the scope of the user preferences, which can bring diversity to the system. Additionally, other methods than PCC and cosine and new

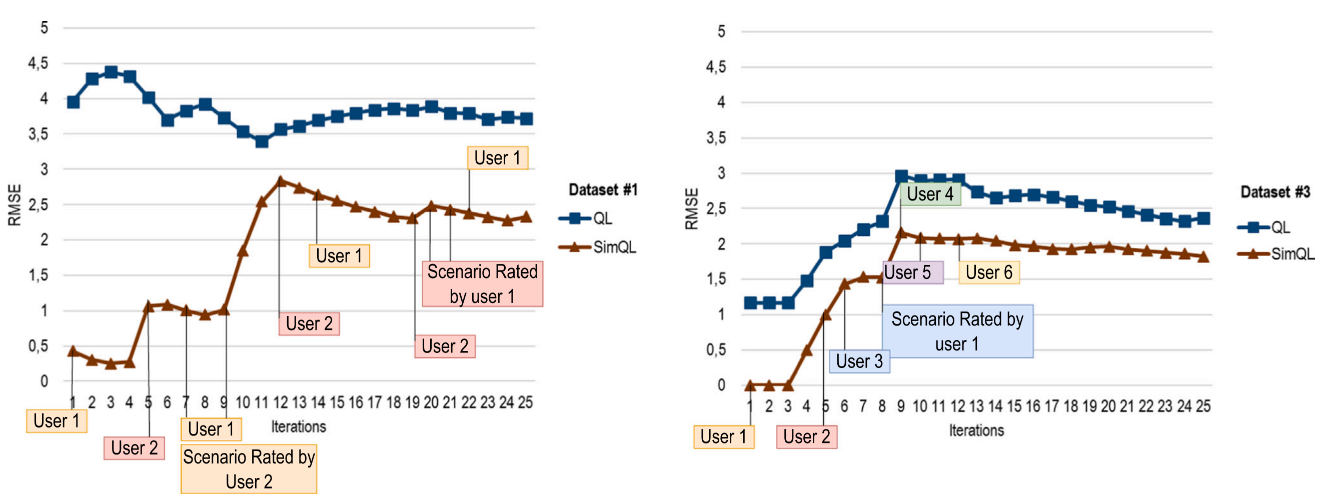

Fig. 6. Comparison of the performance of *SimQL* model with *QL* model.

similarity measures should be tested to ensure more efficient management of the data sparsity problem.

Funding

The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021). The author Flávia Pires thanks the Fundação para a Ciência e Tecnologia (FCT), Portugal for the PhD Grant SFRH/BD/143243/2019.

CRediT authorship contribution statement

Flávia Pires: Conceptualisation of this study, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualisation. **Paulo Leitão:** Conceptualisation of this study, Methodology, Validation, Formal analysis, Writing – review & editing. **António Paulo Moreira:** Conceptualisation of this study, Methodology, Validation, Formal analysis, Writing – review & editing. **Bilal Ahmad:** Validation, Software, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

References

Abdul-Raham, A., Hailes, S., 1998. Distributed trust model. Proc. 1997 Workshop N. Secur. Paradig. 48–60.

Ahuett-Garza, H., Kurfess, T., 2018. A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing. Manuf. Lett. 15, 60–63. https://doi.org/10.1016/j.mfglet.2018.02.011.

Al Fararni, k, Aghoutane, B., Riffi, J., Sabri, A., Yahyaouy, A., 2020. Comparative study on approaches of recommendation systems. Adv. Intell. Syst. Comput. 1076, 753–764. https://doi.org/10.1007/978-981-15-0947-6_72.

B. Thorat, P., M. Goudar, R., Barve, S., 2015. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. Int. J. Comput. Appl. 110, 31–36. https://doi.org/10.5120/19308-0760.

Balabanovíc, M., Shoham, Y., 1997. FabContent-based, collaborative recommendation. Commun. ACM 40, 66–72. https://doi.org/10.1145/245108.245124.

Çano, E., Morisio, M., 2017. Hybrid recommender systems: a systematic literature review. Intell. Data Anal. 21, 1487–1524. https://doi.org/10.3233/IDA-163209 arXiv:1901.03888.

Chaplin, J.C., Martinez-arellano, G., Mazzoleni, A., 2020. Digital twins and intelligent decision making. Digit. Manuf. SMEs. 159–187.

Chen, H., Sun, H., Cheng, M., Yan, W., 2021. A recommendation approach for rating prediction based on user interest and trust value. Comput. Intell. Neurosci. 2021, 1–9. https://doi.org/10.1155/2021/6677920.

Geravanchizadeh, M., Roushan, H., 2021. Dynamic selective auditory attention detection using RNN and reinforcement learning. Sci. Rep. 11, 1–12. https://doi.org/10.1038/s41598-021-94876-0.

Grieves, M., Vickers, J., 2017. Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches. Springer, pp. 85–113. https://doi.org/10.1007/978-3-319-38756-7_4.

Guo, G., 2013. Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems. Proc. 7th ACM Conf. Recomm. Syst. 451–454. https://doi.org/10.1145/2507157.2508071.

Guo, G., Zhang, J., Yorke-Smith, N., 2015. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. Proc. AAAI Conf. Artif. Intell. 123–129. https://doi.org/10.1609/aaai.v29i1.9153.

Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A., 2015. Recommendation systems: principles, methods and evaluation. Egypt. Inform. J. 16, 261–273. https://doi.org/10.1016/j.eij.2015.06.005.

Jain, G., Mahara, T., 2019. An efficient similarity measure to alleviate the cold-start problem. 2019 15th Int. Conf. Inf. Process.: Internet Things (ICINPRO) 1–8. https://doi.org/10.1109/ICInPro47689.2019.9092250.

Jamali, M., Ester, M., 2009. TrustWalker: A random walk model for combining trust-based and item-based recommendation. Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 397–405. https://doi.org/10.1145/1557019.1557067.

Jamali, M., Ester, M., 2010. A matrix factorization technique with trust propagation for recommendation in social networks. Proc. 4th ACM Conf. Recomm. Syst. 135–142. https://doi.org/10.1145/1864708.1864736.

Kim, H.N., Ji, A.T., Ha, I., Jo, G.S., 2010. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electron. Commer. Res. Appl. 9, 73–83. https://doi.org/10.1016/j.elerap.2009.08.004.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 426–434. https://doi.org/10.1145/1401890.1401944.

Koren, Y., 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Trans. Knowl. Discov. Data 4, 1–24. https://doi.org/10.1145/1644873.1644874.

Kunath, M., Winkler, H., 2018. Integrating the Digital Twin of the manufacturing system into a decision support system for improving the order, management process. 51st CIRP Conference on Manufacturing Systems. Elsevier B.V., pp. 225–231. https://doi.org/10.1016/j.procir.2018.03.192

Lin, Y., Liu, Y., Lin, F., Zou, L., Wu, P., Zeng, W., Chen, H., Miao, C., 2021. A Survey on Reinforcement Learning for Recommender Systems 14, 1–21. arXiv:2109.10665.

Ma, H., Yang, H., Lyu, M.R., King, I., 2008. SoRec: social recommendation using probabilistic matrix factorization. Proc. 17th ACM Conf. Inf. Knowl. Manag. 931–940. https://doi.org/10.1145/1458082.1458205.

Ma, H., King, I., Lyu, M.R., 2009. Learning to recommend with social trust ensemble. Proc. 32nd Annu. Int. ACM, SIGIR Conf. Res. Dev. Inf. Retr. 203–210. https://doi.org/10.1145/1571941.1571978.

Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I., 2011. Recommender systems with social regularization. Proc. 4th ACM Int., Conf. Web Search Data Min. 287–296. https://doi.org/10.1145/1935826.1935877.

Malik, S., Rana, A., Bansal, M., 2020. A survey of recommendation systems. Inf. Resour. Manag. J. 33, 53–73. https://doi.org/10.4018/IRMJ.2020100104.

Massa, P., Avesani, P., 2007. Trust-aware recommender systems. Proc. 2007 ACM Conf. Recomm. Syst. 17–24. https://doi.org/10.1145/1297231.1297235.

Meyer, F., 2012. Recommender systems in industrial contexts. Ph.D. thesis. URL: ⟨http://arxiv.org/abs/1203.4487⟩, arXiv:1203.4487.

O'Donovan, J., Smyth, B., 2005. Trust in recommender systems. Proc. 10th Int. Conf. Intell. Use Interfaces 167–174. https://doi.org/10.1145/1040830.1040870.

Pandya, K., Pathar, S., 2020. Comparison of various technique of recommendation systems. J. Emerg. Technol. Innov. Res. 7, 560–565. ISSN-2349-5162.

Pereira, N., Varma, S., 2019. Financial planning recommendation system using content-based collaborative and demographic filtering. In: Smart Innov. Commun. Comput. Sci., 2019. Springer Singapore, pp. 141–151. https://doi.org/10.1007/978-981-10-8968-8_12.

Pires, F., Ahmad, B., Moreira, A.P., Leitao, P., 2021a. Digital twin based what-if simulation for energy management. 2021 4th IEEE, Int. Conf. Ind. Cyber-Phys. Syst. (ICPS) 309–314. https://doi.org/10.1109/ICPS49255.2021.9468224.

Pires, F., Ahmad, B., Moreira, A.P., Leitao, P., 2021b. Recommendation system using reinforcement learning for what-if simulation in digital twin. IEEE Int. Conf. Ind. Inform. (INDIN) 1–6. https://doi.org/10.1109/INDIN45523.2021.9557372.

Portugal, I., Alencar, P., Cowan, D., 2018. The use of machine learning algorithms in recommender systems: a systematic review. Expert Syst., Appl. 97, 205–227. https://doi.org/10.1016/j.eswa.2017.12.020 arXiv:1511.05263.

Rasheed, A., San, O., Kvamsdal, T., 2020. Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access 8, 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143.

Reshma, R., Ambikesh, G., Santhi Thilagam, P., 2016. Alleviating data sparsity and cold start in recommender systems using social behaviour. 2016 International Conference on Recent Trends in Information Technology (ICRTIT). IEEE,, pp. 1–8. https://doi.org/10.1109/ICRTIT.2016.7569532.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. GroupLens: an open architecture for collaborative filtering of netnews. Proc. 1994 ACM Conf. Comput. Support. Coop. Work 175–186. https://doi.org/10.1145/192844.192905.

Sala, R., Pezzotta, G., Pirola, F., Huang, G.Q., 2019. Decision-support system-based service delivery in the product-service system context: Literature review and gap analysis. 11th CIRP Conference on Industrial Product-Service Systems. Elsevier B.V., pp. 126–131. https://doi.org/10.1016/j.procir.2019.03.140

Selmi, A., Brahmi, Z., Gammoudi, M.M., 2016. Trust-based recommender systems: an overview. Proc. 27th Int. Bus., Inf. Manag. Assoc. Conf. (IBIMA) 371–380.

Sharma, R., Singh, R., 2016. Evolution of recommender systems from ancient times to modern era: a survey. Indian J. Sci. Technol. 9, 1–12. https://doi.org/10.17485/ijst/2016/v9i20/88005.

Son, L.H., 2016. Dealing with the new user cold-start problem in recommender systems: a comparative review. Inf. Syst. 58, 87–104. https://doi.org/10.1016/j.is.2014.10.001.

Song, H., Pei, Q., Xiao, Y., Li, Z., Wang, Y., 2017. A novel recommendation model based on trust relations and item ratings in social networks. 2017 Int. Conf. Netw. Netw. Appl., (NaNA) 17–23. https://doi.org/10.1109/NaNA.2017.17.

Sorde, R.K., Deshmukh, S.N., 2015. Comparative study on approaches of recommendation systems. Int. J. Comput. Appl. 118, 10–14. https://doi.org/10.1007/978-981-15-0947-6_72.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. MIT Press. https://doi.org/10.1108/k.1998.27.9.1093.3.

Tan, Y., Yang, C., Wei, X., Chen, C., Liu, W., Li, L., Zhou, J., Zheng, X., 2022. MetaCare++: meta-learning with hierarchical subtyping for cold-start diagnosis prediction in healthcare data. Proc. 45th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. 449–459. https://doi.org/10.1145/3477495.3532020.

Tang, J., Hu, X., Liu, H., 2013. Social recommendation: a review. Soc. Netw. Anal. Min. 3, 1113–1133. https://doi.org/10.1007/s13278-013-0141-9.

Urdaneta-Ponte, M.C., Mendez-Zorrilla, A., Oleagordia-Ruiz, I., 2021. Recommendation systems for education: systematic review. Electronics 10, 1611. https://doi.org/10.3390/electronics10141611.

Velankar, M., Parag, K., 2023. Music recommendation systems: overview and challenges. In: Adv. Speech Music Technol. Comput. Asp. Appl., 2023. Springer International Publishing, pp. 51–69. https://doi.org/10.1007/978-3-031-18444-4_3.

Victor, P., De Cock, M., Cornelis, C., 2011. Trust and Recommendations. Recommender Systems Handbook. Springer, pp. 645–675. https://doi.org/10.1007/978-0-387-85820-3_20.

Weng, L.T., 2008. Information Enrichment for Quality Recommender Systems. Ph.D. thesis.

Yin, S., Luo, X., 2021. A survey of learning-based methods for cold-start, social recommendation, and data sparsity in e-commerce recommendation systems (Chengdu, China). 16th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE) 276–283. https://doi.org/10.1109/ISKE54062.2021.9755389.

Zhang, H., Liu, G., Wu, J., 2018. Social collaborative filtering ensemble. Pac. Rim Int. Conf. Artif. Intell. 1005–1017. https://doi.org/10.1007/978-3-319-97304-3_77.