STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–10.
Published online in International Academic Press (www.IAPress.org)

IoT-CR: A Novel IoT-Based Approach to Address Data Sparsity in Context-Aware Recommendation Systems

Mohamed El Amine Chafiki[1], Oumaima Stitini[1,2], Soulaimane Kaloun[1]

[1] *Cadi Ayyad University, Faculty of Science and Technology, Laboratory of Computer and Systems Engineering (L2IS), Marrakesh, Morocco*
[2] *Cadi Ayyad University, Ecole Normale Superieure, Department of computer science, Marrakech, Morocco*

Abstract In recent years, recommendation systems (RS) have become an essential part of modern online services, helping users discover new products, content, and experiences that match their preferences and needs. In particular, context-aware recommendation systems (CARS) have received considerable attention because of their capacity to use contextual information to deliver relevant and personalized recommendations, compared to traditional recommendation systems that only use user preferences and item attributes to make recommendations. However, the performance and effectiveness of CARS are challenged by the rise of data sparsity, a common issue in many recommender systems. It occurs when there is an insufficient amount of user-item interactions. This study explores using varied IoT contextual data to address this issue. We present and assess an IoT-assisted Contextual Recommendation (IoT-CR) system, which is an end-to-end deep learning framework architecture that aims to incorporate rich contexts from IoT sensors seamlessly into the recommendation process. To prove this concept, we perform an extensive comparative study against a set of baseline models on four different public, context-rich datasets. We find a mixed bag of results, indicating that the performance of models depends very much on the characteristics of the datasets such as their size and sparsity. In particular, the IoT-CR framework achieves the best results on the largest dataset where there is enough data for it to learn complex interactions. On the other hand, in smaller or more sparse situations, classical collaborative filtering or tree-based models perform better. This research offers an important benchmark, stating that although supplementing data with IoT signals is a very good way forward, the effectiveness of complex models is not a general case and depends critically on the data landscape.

Keywords context-aware recommendation systems, data sparsity, internet of things, deep learning, collaborative filtering, matrix factorization, personalization

DOI: 1.19139/soic-2310-5070-2752

1. Introduction

During the last decades, the sheer amount of online information has grown exponentially, due to the ever-increasing popularity of many web services such as Amazon, Netflix, Spotify, etc. As a result, more data flooded in, and navigating through this data became a challenging task. Recommendation systems have emerged as a critical tool to address this challenge, and they have begun to take up more and more place in our lives, suggesting to users what products to buy, what music to listen to, and what articles to read. In other words, RSs are designed to filter and suggest items that match each user's unique preferences, making it easier for them to discover content tailored to their individual needs.

The RS has been developed and improved through significant research since [1] introduced the concept of collaborative filtering. Early methods focused on collaborative filtering and content-based strategies, using user-item interactions and item attributes to generate recommendations. However, these traditional methods often do not

[*] Correspondence to: M. Chafiki (Email: m.chafiki.ced@uca.ac.ma). Cadi Ayyad University, Faculty of Science and Technology, Laboratory of Computer and Systems Engineering (L2IS), B.P 549, Av.Abdelkarim Elkhattabi, Gueliz Marrakech, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

fully comprehend the vibrant and complex nature of user preferences, which can be influenced by several factors such as time, geography, and a user's surrounding social community.

To address this limitation, CARS came to life. By incorporating context in the recommendation process, CARS can provide more relevant and personalized suggestions than traditional methods. Context is described as any piece of information that features the environment of an item [2]. To provide recommendations that are appropriate for the user's present circumstance, CARS takes into account a variety of variables, including time and place, user behavior, and past interactions [3].

While CARS have achieved success in a variety of sectors and application domains, data sparsity remains the most crucial and challenging factor that limits the performance and efficiency of these systems. [4]. The problem is stated when the number of users who have visited the items is way less than the total possible number of possible user-items interaction, that's why the recommender system fails to create suggestions when there is no intersection in users' ratings [5]. Our work aims to propose a novel approach to treat data sparsity in CARS. We propose an approach called IoT-assisted Contextual Recommendation (IoT-CR) that exploits the Internet of Things (IoT) to solve this problem. By integrating dense context data from IoT sensors and connected devices, we aim to enrich the system's understanding of user preferences and deliver more accurate recommendations. This paper presents a rigorous, comparative analysis of an end-to-end deep learning implementation of the IoT-CR concept to understand the conditions under which such an approach excels.

In the remaining sections of the paper, A literature review is included in section 2. The next section explains our proposed IoT-CR framework. The fourth section presents the experimental setup and evaluation protocol. Section 5 summarizes the research results and findings, and the concluding remarks are offered in the final section.

2. Literature Review

During the last few years, the field of CARS has witnessed extensive research aimed at refining the methods of incorporation and management of contextual information, increasing the performance and efficiency of CARS, as well as pleasing the user in order to provide accurate recommendations.Context can be represented either implicitly or explicitly [7]. When it comes to integrating contextual information, three methods have been proposed which are pre-filtering, post-filtering, and contextual modeling [8]. Several approaches have been set to incorporate contextual data in the recommendations process.

A popular area of research in CARS is matrix factorization (MF), which learns hidden patterns from just the ratings data. For example [9] discussed Contextual Bias MF (CBMF), a method that builds upon matrix factorization and calculates rating forecasts using mean rating, user bias, item bias, and user-item affinity. Furthermore, [10] suggested Kernel Context Aware MF that incorporates implicit feedback related to users and items, a novel kernel loss function, and stochastic gradient descent optimization with weighted regularization to make context-aware recommendations.

Factorization Machines (FM) are another significant advancement in CARS, extending beyond traditional matrix factorization techniques. [11] proposed Convolutional Factorization Machines (CFM), which combine the strengths of factorization machines, Convolutional Neural Networks (CNN), and attention mechanism to enhance context-aware recommendations. Another direction of research is the Markov model, [12] proposes Hierarchical Hidden Markov Model. This method models each user context as a latent variable influenced by user feedback over time, then identifies common contextual change patterns and employs them to forecast the subsequent user context.

Building upon the concept of factorization, the literature also explores tensor factorization (TF). Tensor-based models have proven to perform exceptionally well, exceeding traditional models, due to their capacity to incorporate contextual data through acquiring a user, item, and context tensor, rather than the conventional user and item matrix. [13]. Zhao et al. (2021) [13] introduced the time varying bias TF (TBTF), which builds upon the bias TF approach and integrates biases associated with users and items that change over time into the TF model to improve the exactitude of recommendations in CARS. Inspired by the rapid growth of adversarial learning, [14] introduced Adversarial Tensor Factorization (ATF), a novel strategy combining TF with adversarial learning techniques designed to enhance the effectiveness and performance of CARS.

Due to its effective learning capabilities, deep learning has been employed to a great extent in this area. Using deep learning to generate recommendations for collaborative filtering., [15] proposed a context aware neural CF approach that incorporates item-splitting pre-filtering approach to include contextual information in Neural CF (NCF) which is based on neural networks to create a context-aware recommender system. [16] investigated a context-aware citation recommender through a memory network approach that learns representations of citation contexts and papers employing Bidirectional Long Short-Term Memory, and then uses multilayered End-to-End Memory Network to calculate relevance at a high-level representation. While [17] studied sequential patterns and outlier reduction by proposing a Gated Recurrent Unit (GRU) attention based context aware sequential RS, a model that works by capturing long-term dependencies in sequential data utilizing the GRU architecture as well as employing the attention mechanism concentrating on significant sections of the input queue and apply different weights to items in the rating sequence. [18] employed a Two-headed Attention Fused Auto-encoder method simultaneously learning mappings from user evaluations and implicit feedback.

Another interesting direction of research is the use of graph structure for recommendation. [19] explored a knowledge graph-based multi-context RS which combines automatic rule discovery, local-domain-based feature representation, and attention-based feature fusion, to make use of the advantages of path and propagation-based techniques, to provide enhanced suggestions. [20] leveraged an encoder/decoder architecture, centered by graph convolution layers, to propose a Graph Convolution Machine (GCM), designed to improve context-aware recommendations by leveraging graph convolutions.

While collaborative filtering (CF) and deep learning approaches dominate CARS research, other techniques like kernel mapping and meta-learning offer promising alternatives. For example, Context Aware Kernel Mapping Recommender proposed in [21] uses kernel mapping techniques to learn the structure of the data and leverage different user and item contexts when making recommendations. [22] proposed Contextual Modulation Meta Learning, a structure that incorporates a context encoder, a hybrid context generator, and a context modulation network to address cold-start problems using meta-learning principles.

Recommender algorithms that can be customized to fit different application areas can reveal the fact that the recommendation problem is not a one-size-fits-all issue. The wealth of different domains should be one of the features of the choice of recommender algorithm embedded in CARS, and the exact domain it belongs to is the main factor. Most approaches are built around CF, essentially relying on user and item interactions given in a large set, and therefore, data sparsity is a significant hurdle in advancing CARS. As a result, researchers are actively exploring strategies to address this challenge.

Contextual feature selection is one of those strategies. [23] came up with a context-aware recommendation approach based on embedded feature selection, which is used to select a minimal subset of relevant contexts and remove the irrelevant features. Genetic algorithms (GA) are another proposed strategy. [24] proposed a feature selection-based approach based on GA that aims to enhance context-aware recommendation by selecting contextual feature subsets. Similarly, [25] proposes a Real-coded GA (RCGA) based CARS that learns individual context feature weights using RCGA, then Context-aware Collaborative Filtering (CACF) is applied with contextual pre-filtering and modeling using learned weights. Finally, Effective Missing Value Prediction (EMVP) algorithm is incorporated to predict missing contextual ratings and alleviate sparsity.

Within the domain of session-based CARS, [26] proposed a hybrid contextual recommender system called CHAMELEON - which integrates content and context features - that includes the use of CNN for feature extraction and Recurrent Neural Networks (RNN) (specifically GRU) for sequence modeling in session-based news recommendation. [27] proposes an RNN-based Session-based context-aware recommendation model that reduces the effect of sparsity.

[28] proposed a solution leveraging the graph structure to capture relationships between users, items, and context nodes. A Personalized PageRank algorithm is used to traverse the graph and identify relevant items for each user based on their historical interactions and the current context. This solution is less prone to data sparsity and cold start.

A variant of CARS is Area of Interest (AOI) recommendation that is based on geo-location data. [5] has proposed a solution to generate dynamic personalized AOIs while handling the challenges posed by data sparsity and cold-start problems, which will involve adding multidimensional data's added value to the AOI recommendation process through a hybrid similarity measurement approach.

3. Methodology: The IoT-CR Framework

This section details our proposed IoT-CR approach, which is implemented as an end-to-end deep learning framework. This architecture is designed to holistically learn complex, non-linear interactions among users, items, static context, and dynamic IoT sensor data simultaneously, as illustrated in Figure 1.

The IoT-CR framework is formally defined as a parameterized function f_θ that maps a composite feature vector to a scalar rating prediction. The architecture is designed to first learn low-dimensional representations of sparse inputs and then model high-order interactions between all available features. Let an individual interaction instance be denoted by j. The input to the model consists of a vector of integer-encoded categorical features $\mathbf{x}_{j,cat} \in \mathbb{N}^{N_{cat}}$ and a vector of standardized continuous features $\mathbf{x}_{j,cont} \in \mathbb{R}^{N_{cont}}$.

Embedding Layer. To handle the high cardinality and sparse nature of categorical inputs (such as user and item IDs), the framework first transforms them into dense, real-valued vector representations. Each of the N_{cat} categorical features is associated with a unique, learnable embedding matrix $E_k \in \mathbb{R}^{d_k \times m}$, where $k \in \{1, \ldots, N_{cat}\}$, d_k is the vocabulary size (cardinality) of the k-th feature, and m is the uniform embedding dimension, a key hyperparameter. The embedding lookup operation maps the integer index $x_{j,cat,k}$ to its corresponding dense vector $\mathbf{e}_{j,k} \in \mathbb{R}^m$. All resulting embedding vectors are then concatenated:

$$\mathbf{v}_{j,emb} = \text{concat}(\mathbf{e}_{j,1}, \mathbf{e}_{j,2}, \ldots, \mathbf{e}_{j,N_{cat}}) \tag{1}$$

Feature Fusion Layer. The output from the embedding layer, $\mathbf{v}_{j,emb}$, is subsequently flattened and fused with the continuous feature vector $\mathbf{x}_{j,cont}$ through concatenation. This operation creates a single, unified vector $\mathbf{v}_{j,0}$ which serves as the input to the Multi-Layer Perceptron.

$$\mathbf{v}_{j,0} = \text{concat}(\text{flatten}(\mathbf{v}_{j,emb}), \mathbf{x}_{j,cont}) \in \mathbb{R}^{(N_{cat} \times m) + N_{cont}} \tag{2}$$

Multi-Layer Perceptron (MLP). The core of the interaction modeling occurs in the MLP, which processes the fused vector $\mathbf{v}_{j,0}$ through a stack of L hidden layers. For each layer $l \in \{1, \ldots, L\}$, the forward propagation is defined recursively:

$$\mathbf{v}_{j,l} = f_l(\mathbf{v}_{j,l-1}) = \text{Dropout}(\sigma(\text{BN}(W_l \mathbf{v}_{j,l-1} + \mathbf{b}_l))) \tag{3}$$

where W_l and $\mathbf{b}_l$ are the learnable weight matrix and bias vector for layer l. The function f_l is a composition of a linear transformation followed by Batch Normalization, a non-linear activation, and Dropout.

Batch Normalization (BN). To stabilize the training of a deep network, BN is applied to the affine transformation of each layer. It addresses the problem of internal covariate shift by normalizing the activations within each mini-batch $\mathcal{B}$. For an activation $x_i \in \mathcal{B}$, the normalization is:

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{4}$$

where $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}^2$ are the mean and variance of the mini-batch, and ϵ is a small constant for numerical stability. To preserve the representational capacity of the network, this is followed by a learnable affine transformation: $\text{BN}(x_i) = \gamma \hat{x}_i + \beta$, where γ and β are learnable scale and shift parameters.

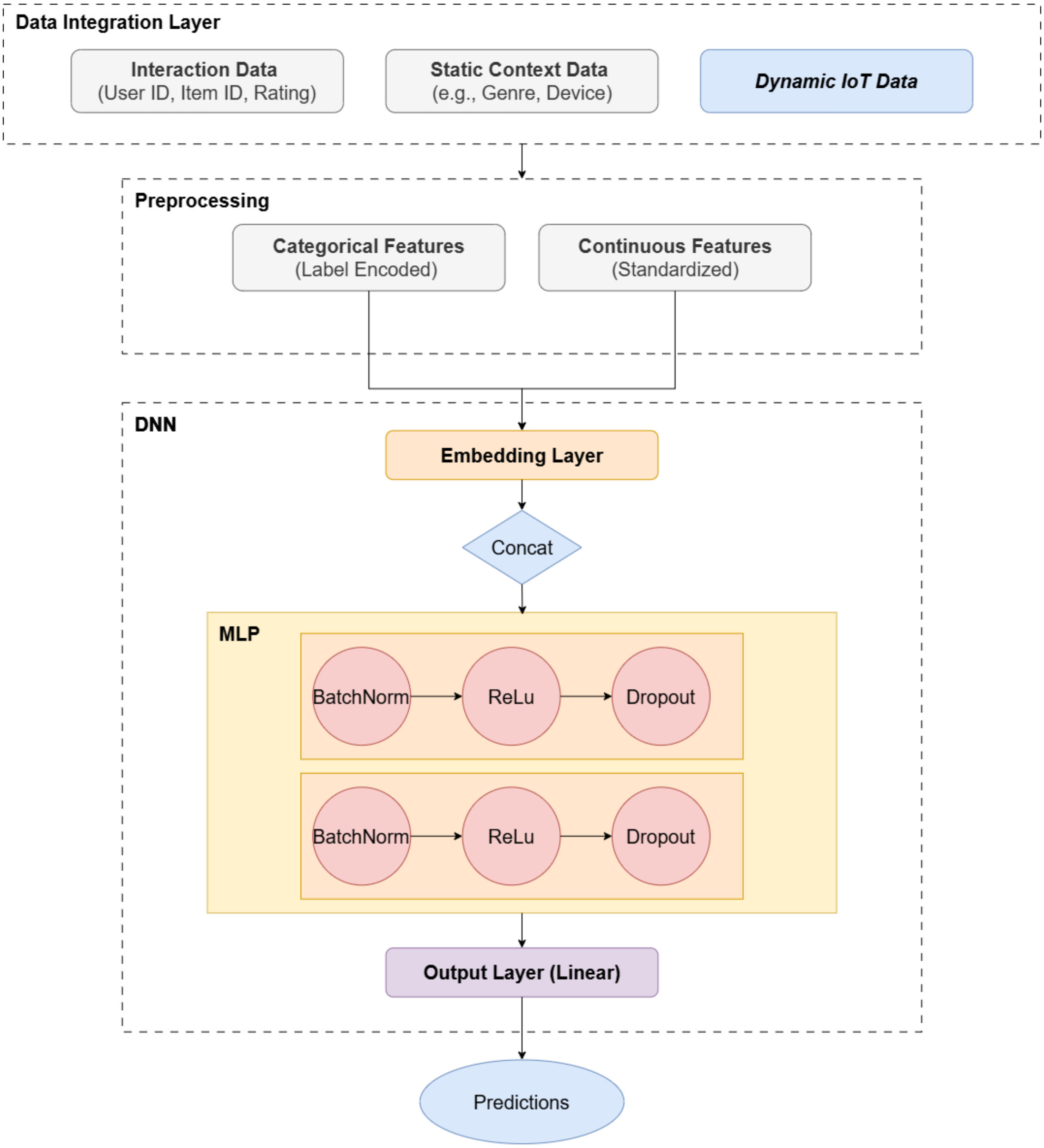

Figure 1. Overall high level architecture of IoT-CR

ReLU Activation (σ). The Rectified Linear Unit, defined as $\sigma(z) = \max(0, z)$, is employed as the non-linear activation function. Without non-linearity, a stack of linear layers would collapse into a single equivalent linear layer, precluding the model from learning complex patterns. ReLU is favored over traditional activations (e.g., sigmoid, tanh) for its computational efficiency and its effectiveness at mitigating the vanishing gradient problem, as its gradient does not saturate for positive inputs.

Dropout. To prevent overfitting, Dropout is used as a stochastic regularization technique during training. For each neuron in a layer, its output is set to zero with a probability p (the dropout rate). This method inhibits intricate co-adaptation among neurons, thereby encouraging the learning of a more robust and redundant feature set. At inference time, all neurons are used, but their outputs are scaled by a factor of $(1-p)$ to maintain the same expected output as during training (a technique known as inverted dropout).

Output Layer. The final predicted rating, $\hat{y}_j$, is produced by a single linear output neuron that projects the final hidden state $\mathbf{v}_{j,L}$ to a scalar value:

$$\hat{y}_j = \mathbf{w}_{out}^{\top}\mathbf{v}_{j,L} + b_{out} \tag{5}$$

Objective Function. The set of all learnable model parameters $\theta = \{E_k, W_l, \mathbf{b}_l, \gamma_l, \beta_l, \mathbf{w}_{out}, b_{out}\}$ are optimized by minimizing the Mean Squared Error (MSE) loss function over the training set $\mathcal{T}$:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{T}|}\sum_{j\in\mathcal{T}}(\hat{y}_j - y_j)^2 \tag{6}$$

where y_j is the ground-truth rating for interaction j. This objective is minimized using the AdamW optimization algorithm.

4. Experiments and Evaluation

This section delineates the detailed experimental procedure intended to systematically examine the IoT-CR framework, in addition to the baseline models, across a suite of real datasets.

4.1. Datasets

To assure a meaningful evaluation, we selected four unique, publicly available datasets that provide the different rich contextual information. The variation in domain, data type and complexity of contextual inference provides an increasingly rigorous dataset.

- **LDOS-CoMoDa [34]:** A dataset for movie recommendations rich with context, which obtained ratings and context information from users immediately after they had viewed a movie. This dataset provides high quality, in-the-moment context. The context dimensions are extensive and include a user's mood, social context (alone, partner, friend), and physical context (location, weather).
- **Frappe [35]:** A sizable dataset of mobile application utilization logs obtained from an actual implementation of a context-aware recommendation system. Interactions are implicit (usage frequency) and the context describes common mobile scenarios, including time of day, day of week, and user location (home, work). The size and real-world provenance of this dataset makes it well-suited to testing both scalability and effectiveness on implicit feedback.
- **InCarMusic [37]:** An annotated dataset concentrating on the music preferences of users within a car environment. One of the contextual features is closely related to the driving environment by capturing features related to the type of road (city, highway), traffic levels, weather, as well as the driver's perceived mood and sleepiness. This dataset is a clear demonstration of what an IoT-based context-rich environment looks like.

Table 1 provides a quantitative summary of these datasets, highlighting their differences in scale, sparsity, and contextual richness.

4.2. Evaluation Protocol

In order to ensure a comprehensive, reproducible, and fair comparison of all models, we developed a comprehensive evaluation procedure focusing on cross-validation and an additional set of standard performance measures.

Table 1. Summary of dataset characteristics.

Dataset	Interactions	Users	Items	Sparsity (%)	Context Factors	IoT Factors
CoMoDa	2,296	121	1,232	98.46%	39	10
Frappe	96,203	957	4,082	97.54%	8	4
STS	2,534	325	249	96.87%	23	10
InCarMusic	4,012	42	139	31.28%	9	7

4.2.1. Cross-Validation and Model Training. For all experiments across all datasets, a 5-fold cross-validation (CV) method was used. We opted for 5-fold CV to give our results greater robustness and data generalizability by evaluating performance across different subsets of data, rather than performing a single, arbitrary train-test split, which would be sensitive to evaluation bias. For each of the 5 folds, the data was split into a training set (80% of the data) and a held-out test set (20%). The training set was then further split, reserving a portion (e.g., 15% of the 80%) as a validation set. The validation dataset served two main purposes for our DNN-based IoT-CR framework: (1) it was used to carry out hyperparameter optimization, and (2) it was used to provide the error signal to terminate training in the event of overfitting. For the baseline models, training was accomplished against the full data (80% training section). The final performance for all models was acquired from the held-out test dataset in each fold, and averaged across the 5 folds.

4.2.2. Performance Metrics. The performance of the model on this rating prediction task was evaluated using three common metrics from regression and added complementary regression metric. Let y_j be the true rating for interaction j, $\hat{y}_j$ be the model's prediction for the rating, $\bar{y}$ is the mean of all true ratings and N is the total number of interactions in the test set.

- *Root Mean Squared Error (RMSE):* This is the main metric to evaluate accuracy in rating prediction. RMSE is especially sensitive to large errors due to squaring the residuals from the estimated ratings to the reference ratings.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2} \tag{7}$$

- *Mean Absolute Error (MAE):* This metric provides a more direct and interpretable expression of the average magnitude of the errors. In contrast to RMSE, MAE weights the same amount of error regardless of size. This causes MAE to be less sensitive to outliers than RMSE.

$$\text{MAE} = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j| \tag{8}$$

- *R-squared (R²), also known as the Coefficient of Determinationn:* A representational metric of the variance in the true ratings that is available to be captured from the model. R² represents how much variability in the true ratings are reflective of the variability in the model ratings. An average of 1.0 indicates a perfect fit. An average of 0.0 indicates the model performs no better than a naive baseline of always predicting the average of the true ratings $\bar{y}$. An R² can be negative and can indicate the model performed worse than the baseline average of the target ratings, suggesting that even simple mean based average can be used to build a more complex model.

$$R^2 = 1 - \frac{\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}{\sum_{j=1}^{N}(y_j - \bar{y})^2} \tag{9}$$

4.3. Baseline Models

In order to thoroughly assess the merits of the suggested IoT-CR framework, we compared its performance to a variety of benchmarks which were constructed to represent different modeling paradigms. The benchmarks included performances of pure collaborative filtering, pure content-based filtering, and hybrid ensemble approaches that represented current state of the art. This will be useful to comprehensively assess the respective strengths and weaknesses of the IoT-CR framework.

Collaborative Filtering Baselines. These models are used only to gain a performance measure of the pure predictive power of the user-item interaction matrix in isolation, without using contextual information. They serve as the simplest and fundamental performance benchmark based purely on the collaborative signal.

- *Singular Value Decomposition (SVD) [38]:* Matrix factorization models are a foundational approach to recommendation. SVD accomplishes this by performing a factorization of the sparse user-item rating matrix into two dense, lower dimensional matrices that represent latent factors for users and items. The principal strength of SVD technologies is their ability to handle sparsity through learning embeddings that are dense, and therefore usable for predicting missing ratings.

- *Alternating Least Squares (ALS) [39]:* An alternative robust and commonly implemented matrix factorization method. This method, like SVD, seeks to discover latent user and item factors. It efficiently does this through an iterative optimization process where the user factors are fixed first, and the item factors are solved, and then the item factors are fixed and the user factors are solved. This method is also scalable, particularly in distributed computing systems.

Content-Only Baseline. This model serves as an essential control experiment, designed to evaluate the predictive value of the contextual and IoT features in isolation (i.e. without user or item IDs).

- *LGBM (Content-Only):* We use LightGBM, a highly efficient gradient boosting framework. By training this model exclusively on contextual data, we can establish a performance baseline that quantifies the inherent predictive power of the context itself, independent of user or item identity.

Hybrid Baselines (Tree-Based). This class encapsulates robust, non-deep learning alternatives that can readily incorporate collaborate signals (user/item ids) and contextual features. They are strong benchmark models for performance on structured, tabular data, and have established a reputation for capturing complex feature interactions.

- *Random Forest:* A bagging-based, ensemble approach. It creates many decision tree classifiers during training and outputs the average prediction of the individual trees is. Its inherent randomisation makes it robust to overfitting and efficient in handling heterogeneous features and outcome levels without extensive pre-processing.

- *XGBoost (Extreme Gradient Boosting):* An advanced state-of-the-art implementation of gradient boosting. It sequentially builds trees such that each tree is trained to correct the errors of previous trees. This is one of the most highly regarded gradient boosting algorithms in terms of predictive accuracy and computational efficiency on tabular datasets.

5. Results and Discussion

This section presents the aggregated results from our 5-fold cross-validation experiments and offers a detailed analysis of the findings.

5.1. Quantitative Results

The experimental results show a diverse performance scenario where no one models outperforms all others across all datasets. Table 2 shows an overall summary of all three metrics for each model in the four datasets. The best result for each metric in each dataset is highlighted in bold indicating the influence of datasets on model performance.

Table 2. Comprehensive summary of average performance metrics across 5 folds. The best result for each metric within each dataset is highlighted in bold.

	CoMoDa			**Frappe**			**STS**			**InCarMusic**		
Model	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2
SVD	1.0563	0.7896	-0.0005	1.3852	1.0577	0.3419	1.1530	0.8840	0.2043	**1.0100**	0.6912	**0.5344**
ALS	1.1643	0.8441	-0.2178	1.4435	1.1031	0.2854	1.2308	0.9081	0.0892	1.0397	**0.6478**	0.5055
LGBM-Content	0.9425	**0.6594**	0.2042	1.5026	1.2111	0.2257	1.2340	0.9534	0.0883	1.4418	1.1516	0.0507
Random Forest	**0.9268**	0.6599	**0.2300**	1.2242	0.9528	0.4860	**1.1368**	**0.8437**	**0.2256**	1.0977	0.7605	0.4498
XGBoost	0.9471	0.6668	0.1959	1.2123	0.9382	0.4960	1.1920	0.8607	0.1483	1.1316	0.7807	0.4151
DNN (IoT-CR)	1.1643	0.8772	-0.2207	**1.1094**	**0.8300**	**0.5779**	1.1955	0.9025	0.1420	1.1445	0.8283	0.4013

5.2. Performance Analysis and Discussion

The findings derived from the empirical inquiry indicate that the best modeling approach varies considerably based on the data characteristics, which include the scale, sparsity, and format of contextual data. These findings provide more understanding on when each model type is preferred.

On the Frappe dataset, which was the largest dataset by a significant margin, our tuned IoT-CR (DNN) framework was a clear winner across all three metrics (RMSE, MAE, and R^2). This is exactly the strength of deep learning: using a lot of data to learn complex patterns. The near 100,000 interactions provides a sufficiently dense signal to meaningfully train millions of parameters in the DNN with minimal overfitting. The model could learn useful, high quality user and item embeddings and importantly, the complex non-linear interaction between these embeddings and the contextual features, which other methods were less capable of doing. This example confirms the central premise of the IoT-CR method in a data-rich environment.

The results on the CoMoDa dataset are strikingly different. Here, the collaborative filtering models (SVD, ALS) and the DNN implementation of IoT-CR performed very poorly, evidenced by their high error metrics and negative R^2 values, which indicate a model fit worse than a simple horizontal line. The best-performing models were the tree-based hybrids (Random Forest) and the content-only model. This suggests that the user-item interaction matrix for CoMoDa is so sparse (98.46%) that the collaborative signal is exceptionally weak and unreliable. In this scenario, the learned user and item embeddings are unstable and fail to generalize. Consequently, models that can pivot to rely almost entirely on the rich, explicit contextual features, such as tree-based ensembles, gain a decisive advantage.

On the STS and InCarMusic datasets, which represent a more classic recommender system scenario of moderate size and sparsity, the high-capacity IoT-CR framework was outperformed by simpler models. For STS, the Random Forest hybrid achieved the best result on all metrics. For InCarMusic, classic collaborative filtering dominated, with SVD achieving the best RMSE and R^2, and ALS securing the best MAE. In these cases, the number of interactions appears insufficient for the complex DNN to converge to a generalizable solution without overfitting. The strong inductive bias of matrix factorization (the low-rank assumption) acts as a powerful form of regularization that is perfectly suited to these conditions. This demonstrates a better bias-variance trade-off on limited data.

In conclusion, our research uncovers a pivotal insight: although the IoT-CR framework exhibits considerable potential, its efficacy is contingent upon data scale. The high parametric complexity of a deep learning model is not warranted for smaller or very sparse datasets; instead, simpler, more robust methods with stronger inductive biases are the preferred option.

6. Conclusion and Future Work

In this work, we introduced IoT-CR, a novel deep learning framework designed to mitigate the persistent challenge of data sparsity in context-aware recommender systems by leveraging rich, granular data from IoT sensors. We presented an end-to-end neural network architecture and conducted a rigorous empirical evaluation across four distinct datasets, benchmarking our approach against a strong suite of collaborative filtering, content-based, and hybrid models. Our findings reveal that the integration of IoT data is not a universal solution to data sparsity, but a powerful tool whose effectiveness is highly contingent on the underlying characteristics of the dataset. On the largest dataset, our proposed IoT-CR framework achieved state-of-the-art performance. This confirms our central hypothesis: when a sufficient baseline of user-item interactions exists, rich IoT contextual data can effectively densify the available information. The model is able to learn complex, non-linear interactions between user behavior and their environment, unlocking predictive power that simpler models cannot capture.

However, in scenarios with higher data sparsity or fewer interactions, the complexity of the IoT-CR framework was outperformed by simpler models. On datasets where the collaborative signal was exceptionally weak, the rich IoT context was not enough to prevent the deep learning model from overfitting. In these cases, tree-based models that excelled at leveraging explicit contextual features, or classic matrix factorization models with their strong inductive bias, proved more robust. This demonstrates that IoT data serves to enhance an existing collaborative signal rather than create one from a void.

This study underscores a crucial lesson for practitioners: the value of the IoT-CR approach lies in its ability to unlock predictive power from large-scale, context-rich data streams. The choice of model must be guided by a careful analysis of the data's scale and sparsity. For smaller or nascent systems, classic matrix factorization remains a robust choice. As interaction data grows, the integration of IoT context via a sophisticated framework like IoT-CR becomes a viable and powerful strategy.

Building upon these insights, future work should explore hybrid architectures that combine the robust collaborative signal modeling of matrix factorization with the contextual feature interaction capabilities of DNNs. Such models could potentially offer strong performance across different levels of data sparsity. Beyond architectural improvements, the practical deployment of systems like IoT-CR introduces significant ethical considerations. The very data that makes this approach powerful—such as user mood and location—is inherently sensitive and raises privacy concerns. Therefore, a critical direction for future research is the integration of privacy-preserving machine learning (PPML) techniques. This includes exploring federated learning, where models are trained on-device to prevent raw data from leaving a user's control, and applying differential privacy to provide mathematical guarantees that individual user data cannot be re-identified. Developing frameworks that balance powerful, context-aware personalization with robust user privacy is essential for the responsible application of this technology. Furthermore, investigating methods to dynamically quantify a dataset's "readiness" for a complex deep learning approach would provide invaluable guidance for automatically selecting the most appropriate model architecture in real-world systems.

REFERENCES

1. Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70.
2. Dey, A. K. (2001). Understanding and using context. Personal and ubiquitous computing, 5, 4-7.
3. Stitini, O., García-Magariño, I., Kaloun, S., & Bencharef, O. (2023). Towards Ideal and Efficient Recommendation Systems Based on the Five Evaluation Concepts Promoting Serendipity. Journal of Advances in Information Technology, 14(4), 701-717.
4. Idrissi, N., & Zellou, A. (2020). A systematic literature review of sparsity issues in recommender systems. Social Network Analysis and Mining, 10(1), 15.
5. Kolahkaj, M., Harounabadi, A., Nikravanshalmani, A., & Chinipardaz, R. (2021). Incorporating multidimensional information into dynamic recommendation process to cope with cold start and data sparsity problems. Journal of ambient intelligence and humanized computing, 1-20.
6. Stitini, O., Ouakasse, F., Rakrak, S., Kaloun, S., & Bencharef, O. (2024). Combining IoMT and XAI for Enhanced Triage Optimization: An MQTT Broker Approach with Contextual Recommendations for Improved Patient Priority Management in Healthcare. International Journal of Online and Biomedical Engineering (iJOE), 20(07), pp. 145–162. https://doi.org/10.3991/ijoe.v20i07.47483

7. S. Raza and C. Ding, “Progress in context-aware recommender systems — An overview,” Computer Science Review, vol. 31, pp. 84–97, Feb. 2019, doi: 10.1016/j.cosrev.2019.01.001.
8. Raza, S., & Ding, C. (2019). Progress in context-aware recommender systems—An overview. Computer Science Review, 31, 84-97.
9. Casillo, M., Gupta, B. B., Lombardi, M., Lorusso, A., Santaniello, D., & Valentino, C. (2022). Context aware recommender systems: A novel approach based on matrix factorization and contextual bias. Electronics, 11(7), 1003.
10. Patil, V. A., Chapaneri, S. V., & Jayaswal, D. J. (2022). Kernel-Based Matrix Factorization With Weighted Regularization for Context-Aware Recommender Systems. IEEE Access, 10, 75581-75595.
11. Xin, X., Chen, B., He, X., Wang, D., Ding, Y., & Jose, J. M. (2019, August). CFM: Convolutional Factorization Machines for Context-Aware Recommendation. In IJCAI (Vol. 19, pp. 3926-3932).
12. Aghdam, M. H. (2019). Context-aware recommender systems using hierarchical hidden Markov model. Physica A: Statistical Mechanics and Its Applications, 518, 89-98.
13. Zhao, J., Yang, S., Huo, H., Sun, Q., & Geng, X. (2021). TBTF: an effective time-varying bias tensor factorization algorithm for recommender system. Applied Intelligence, 51, 4933-4944.
14. Chen, H., & Li, J. (2019, September). Adversarial tensor factorization for context-aware recommendation. In Proceedings of the 13th ACM Conference on Recommender Systems (pp. 363-367).
15. Al Jawarneh, I. M., Bellavista, P., Corradi, A., Foschini, L., Montanari, R., Berrocal, J., & Murillo, J. M. (2020). A pre-filtering approach for incorporating contextual information into deep learning based recommender systems. IEEE Access, 8, 40485-40498.
16. Wang, J., Zhu, L., Dai, T., & Wang, Y. (2020). Deep memory network with bi-lstm for personalized context-aware citation recommendation. Neurocomputing, 410, 103-113.
17. Yuan, W., Wang, H., Yu, X., Liu, N., & Li, Z. (2020). Attention-based context-aware sequential recommendation model. Information Sciences, 510, 122-134.
18. Zhou, J. P., Cheng, Z., Pérez, F., & Volkovs, M. (2020, September). TAFA: Two-headed attention fused autoencoder for context-aware recommendations. In Proceedings of the 14th ACM conference on recommender systems (pp. 338-347).
19. Wu, C., Liu, S., Zeng, Z., Chen, M., Alhudhaif, A., Tang, X., ... & Peng, X. (2022). Knowledge graph-based multi-context-aware recommendation algorithm. Information Sciences, 595, 179-194.
20. Wu, J., He, X., Wang, X., Wang, Q., Chen, W., Lian, J., & Xie, X. (2022). Graph convolution machine for context-aware recommender system. Frontiers of Computer Science, 16(6), 166614.
21. Iqbal, M., Ghazanfar, M. A., Sattar, A., Maqsood, M., Khan, S., Mehmood, I., & Baik, S. W. (2019). Kernel context recommender system (KCR): A scalable context-aware recommender system algorithm. IEEE Access, 7, 24719-24737.
22. Feng, X., Chen, C., Li, D., Zhao, M., Hao, J., & Wang, J. (2021, October). Cmml: Contextual modulation meta learning for cold-start recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 484-493).
23. Chen, L., & Xia, M. (2021). A context-aware recommendation approach based on feature selection. Applied Intelligence, 51, 865-875.
24. Livne, A., Tov, E. S., Solomon, A., Elyasaf, A., Shapira, B., & Rokach, L. (2022). Evolving context-aware recommender systems with users in mind. Expert Systems with Applications, 189, 116042.
25. Linda, S., Minz, S., & Bharadwaj, K. K. (2020). Effective context-aware recommendations based on context weighting using genetic algorithm and alleviating data sparsity. Applied Artificial Intelligence, 34(10), 730-753.
26. Gabriel De Souza, P. M., Jannach, D., & Da Cunha, A. M. (2019). Contextual hybrid session-based news recommendation with recurrent neural networks. IEEE Access, 7, 169185-169203.
27. Wu, T., Sun, F., Dong, J., Wang, Z., & Li, Y. (2022). Context-aware session recommendation based on recurrent neural networks. Computers and Electrical Engineering, 100, 107916.
28. Musto, C., Lops, P., de Gemmis, M., & Semeraro, G. (2021). Context-aware graph-based recommendations exploiting personalized pagerank. Knowledge-Based Systems, 216, 106806.
29. Das, S. D., & Basak, A. (2021). Context-aware Retail Product Recommendation with Regularized Gradient Boosting. arXiv preprint arXiv:2109.08561.
30. Braunhofer, M., Elahi, M., Ricci, F., & Schievenin, T. (2013). Context-aware points of interest suggestion with dynamic weather data management. In Information and Communication Technologies in Tourism 2014: Proceedings of the International Conference in Dublin, Ireland, January 21-24, 2014 (pp. 87-100). Springer International Publishing.
31. Feng, C., Liang, J., Song, P., & Wang, Z. (2020). A fusion collaborative filtering method for sparse data in recommender systems. Information Sciences, 521, 365-379.
32. Wu, D., Luo, X., Shang, M., He, Y., Wang, G., & Zhou, M. (2019). A deep latent factor model for high-dimensional and sparse matrices in recommender systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(7), 4285-4296.
33. Jeong, S. Y., & Kim, Y. K. (2021). Deep learning-based context-aware recommender system considering contextual features. Applied Sciences, 12(1), 45.
34. Kosir, Andrej & Odi, Ante & Kunaver, Matevž & Tkalčič, Marko & Tasic, Jurij. (2011). Database for contextual personalization. ENGLISH EDITION. 78. 270-274.
35. Baltrunas, L., Church, K., Karatzoglou, A., & Oliver, N. (2015). Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. arXiv preprint arXiv:1505.03014.
36. Braunhofer, Matthias & Elahi, Mehdi & Ricci, Francesco. (2014). Techniques for cold-starting context-aware mobile recommender systems for tourism. Intelligenza Artificiale. 8. 129-143. 10.3233/IA-140069.
37. Baltrunas, L. et al. (2011). InCarMusic: Context-Aware Music Recommendations in a Car. In: Huemer, C., Setzer, T. (eds) E-Commerce and Web Technologies. EC-Web 2011. Lecture Notes in Business Information Processing, vol 85. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23014-1_8
38. Zheng, S., Ding, C., & Nie, F. (2018). Regularized singular value decomposition and application to recommender system. arXiv preprint arXiv:1804.05090.
39. Chafiki M. E., Banouar O., Benslimane M. Large-scale recommender systems using Hadoop and collaborative filtering: a comparative study. Mathematical Modeling and Computing. Vol. 11, No. 3, pp. 785–797 (2024)