Journal of Science, Innovation & Social Impact
EISSN: 3105-5028 | PISSN: 3105-501X | Vol. 2, No. 1 (2026)

Article

Optimizing Transaction Matching Performance Using Hybrid Collaborative Filtering and Deep Learning: An Empirical Analysis of Feature Engineering and Similarity Metrics

Ziyi Wang [1,*]

[1] Enterprise Risk Management, Columbia University, NY, USA
* Correspondence: Ziyi Wang, Enterprise Risk Management, Columbia University, NY, USA

Abstract: This paper presents a comprehensive empirical analysis of transaction-matching optimization in commercial real estate markets by integrating collaborative filtering and deep learning techniques. We address critical challenges in buyer-seller matching by developing a hybrid framework that combines matrix factorization-based collaborative filtering with attention-enhanced deep neural networks. Our approach introduces novel feature engineering methodologies designed explicitly for transaction data, incorporating both technical market indicators and behavioral patterns derived from historical transactions. Through extensive experimentation on a dataset of 50,000 commercial real estate transactions, we systematically compare multiple similarity metrics, including cosine similarity, Euclidean distance, and hybrid combinations. The proposed framework achieves 87.3% matching accuracy (Precision@10) and reduces computational latency to 45ms per query, representing significant improvements over baseline methods. Ablation studies reveal that attention mechanisms contribute a 12.4% performance gain, while proper feature engineering accounts for an 18.7% improvement in matching quality.

Keywords: transaction matching; collaborative filtering; deep learning; similarity metrics

Received: 25 December 2025
Revised: 30 January 2026
Accepted: 09 February 2026
Published: 13 February 2026

1. Introduction

1.1. Background and Motivation

1.1.1. Current Challenges in Financial Transaction Matching Algorithms

Financial transaction matching remains a central challenge in modern trading systems, as the precise pairing of buyers and sellers directly influences market efficiency and liquidity. Contemporary markets process millions of transactions daily, necessitating advanced algorithms capable of handling high-dimensional, heterogeneous data while delivering sub-second response times. The complexity of transaction matching stems from diverse participant preferences, dynamic market conditions, and the need to simultaneously balance multiple objectives, including price optimization, execution speed, and fairness. Traditional rule-based matching systems face difficulties in scaling to these demands and often fail to capture subtle patterns in participant behavior. Self-attention mechanisms have been demonstrated to effectively model sequential dependencies, providing a foundation for addressing temporal dynamics in transaction matching scenarios [1].

1.1.2. Evolution from Traditional Matching to AI-Powered Approaches

The progression from conventional order-book matching to intelligent AI-driven systems represents a major shift in financial market operations. Early electronic trading systems relied primarily on simple price-time priority algorithms that matched orders according to fixed rules, without incorporating historical transaction patterns or participant preferences [2]. Such approaches have inherent limitations, particularly in maintaining market stability and liquidity. Machine learning techniques have emerged as robust solutions, enabling systems to learn from historical data and dynamically adapt to changing market conditions. Deep reinforcement learning frameworks have shown substantial improvements in practical algorithmic trading applications, consistently enhancing performance across diverse scenarios [3].

1.1.3. Business Impact of Improved Matching Accuracy in Commercial Real Estate Markets

Commercial real estate markets pose unique challenges for transaction matching due to asset heterogeneity, infrequent transactions, and complex multi-stakeholder negotiations. Unlike liquid financial instruments, real estate assets possess distinct characteristics, requiring algorithms that account for location, size, zoning, capitalization rates, and other property-specific attributes. Enhancing matching accuracy in this context directly contributes to shorter transaction cycles, increased successful closures, and improved market liquidity. Industry data indicates that optimized matching can substantially reduce the average time required to close deals while simultaneously boosting transaction volume and efficiency.

1.2. Research Objectives and Contributions

1.2.1. Identifying Optimal Feature Engineering Techniques for Buyer-Seller Matching

This study systematically investigates feature engineering methodologies tailored to transaction matching in commercial real estate. A comprehensive feature taxonomy is developed, encompassing property characteristics, participant behavior patterns, market dynamics, and temporal factors. The approach introduces novel composite features that capture complex relationships between buyer preferences and property attributes. Comparative analyses highlight the importance of effective feature representation in improving matching performance [4].

1.2.2. Comparative Analysis of Similarity Metrics in Financial Transaction Contexts

A critical focus of this research is the empirical evaluation of similarity metrics across diverse transaction scenarios. Metrics such as cosine similarity, Euclidean distance, Manhattan distance, and hybrid combinations are examined under varying conditions of data sparsity. The analysis evaluates how these metrics perform across heterogeneous feature types, including continuous, categorical, and temporal variables, providing insights into metric selection for practical matching applications.

1.3. Paper Organization and Scope

1.3.1. Overview of Methodology and Experimental Design

The paper follows a structured methodology beginning with a comprehensive literature review of collaborative filtering techniques, deep learning architectures, and feature engineering practices in financial contexts. A hybrid matching framework is proposed, featuring a dual-pathway architecture that integrates collaborative filtering with attention-enhanced deep learning. Experimental validation is conducted using real-world commercial real estate transaction datasets, with results compared across multiple evaluation metrics.

1.3.2. Key Findings and Practical Implications

Experimental results demonstrate notable performance improvements, achieving 87.3% matching accuracy (Precision@10) compared to 71.2% for traditional collaborative filtering and 76.8% for standalone deep learning approaches. Computational efficiency analysis indicates that the hybrid framework maintains an average latency of 45 ms, suitable for real-time deployment. Ablation studies identify key components contributing to performance gains, with attention mechanisms improving accuracy by 12.4% and optimized feature engineering contributing an 18.7% increase. These findings underline the practical significance of integrating feature engineering and attention-based learning for enhancing transaction matching in commercial real estate markets.

2. Literature Review and Related Work

2.1. Traditional Collaborative Filtering in Financial Applications

2.1.1. Matrix Factorization Techniques and Their Limitations

Matrix factorization has been a cornerstone of collaborative filtering systems in financial applications for over a decade. These methods decompose the user-item interaction matrix into lower-dimensional latent factor representations, capturing underlying patterns in transaction behavior. Reinforcement learning extensions to traditional frameworks have shown measurable improvements in trade execution performance, highlighting both the potential and limitations of classical approaches [5]. A primary challenge of matrix factorization lies in handling sparse interaction matrices, which are common in financial markets where participants typically engage in a limited number of transactions relative to all possible matches.

2.1.2. User-Based versus Item-Based Collaborative Filtering Performance

The distinction between user-based and item-based collaborative filtering is particularly significant in transaction matching contexts. User-based approaches identify participants with similar trading behaviors and recommend matches based on historical preferences of comparable users. Item-based methods, in contrast, evaluate similarity between financial instruments or properties, suggesting matches according to asset characteristics. Hybrid approaches that integrate both perspectives can leverage complementary information, achieving improved performance in practical matching scenarios [6].

2.2. Deep Learning Approaches for Transaction Matching

2.2.1. Neural Collaborative Filtering Architectures

Neural collaborative filtering advances traditional matrix factorization by introducing non-linear transformation capabilities via deep neural networks. These architectures replace conventional inner product calculations with neural network layers that learn complex interaction functions between users and items. Online learning algorithms tailored for streaming transaction data have demonstrated the feasibility of adapting neural architectures to real-time trading environments [7].

2.2.2. Attention Mechanisms in Recommendation Algorithms

Attention mechanisms enable models to dynamically focus on the most relevant aspects of input data. Self-attention architectures assign importance weights to individual elements in a transaction history, identifying which past interactions exert the strongest influence on current matching decisions. Integrating social or contextual information through attention mechanisms has been shown to improve prediction accuracy in financial applications [8].

2.2.3. Hybrid Architectures Combining Multiple Techniques

The integration of collaborative filtering with deep learning has led to hybrid architectures that capitalize on the strengths of both paradigms. Such systems often employ parallel pathways, where collaborative filtering captures global patterns and deep learning models extract local non-linearities. Hybrid approaches have demonstrated that traditional techniques can enhance neural network training through better initialization and regularization, yielding improved sequential recommendation performance [9].

2.3. Feature Engineering in Trading Environments

2.3.1. Technical Indicators and Market Signals as Features

Effective feature engineering in trading environments requires careful construction of technical indicators that reflect market dynamics. Foundational features commonly include moving averages, relative strength indices, and Bollinger bands, which characterize price trends and volatility. Empirical studies in cryptocurrency and other financial markets indicate that selecting optimal combinations of these indicators for varying market conditions can significantly influence predictive performance [10].

2.3.2. Behavioral Features from Transaction History

Transaction histories provide rich behavioral information beyond what technical indicators capture. Participant-specific features include trading frequency, average transaction size, holding periods, and profit-loss patterns, reflecting individual risk preferences and investment strategies. Temporal patterns, such as time-of-day effects, day-of-week seasonality, and monthly cycles, reveal systematic behavioral tendencies that can inform more accurate transaction matching models.

3. Proposed Hybrid Matching Framework

3.1. Architecture Design and Components

3.1.1. Collaborative Filtering Pathway for Historical Pattern Extraction

The collaborative filtering pathway employs an enhanced matrix factorization approach specifically optimized for transaction matching. This component processes the historical transaction matrix $H \in R^{(m \times n)}$ where m represents buyers and n denotes properties. We implement alternating least squares optimization with a regularization parameter $\lambda = 0.01$ to decompose H into buyer factors $P \in R^{(m \times k)}$ and property factors $Q \in R^{(n \times k)}$, where $k = 128$ represents latent dimensionality. The factorization objective incorporates temporal weighting through exponential decay: $w_{ij} = \exp(-\alpha\ t_{current} - t_{ij}))$, where $\alpha = 0.1$ controls the decay rate. Multi-agent deep reinforcement learning frameworks have been shown to achieve superior performance in VWAP optimization, providing motivation for adopting distributed computation in our approach [11].

The pathway implements several enhancements over traditional collaborative filtering. Implicit feedback integration captures viewing behavior, inquiry patterns, and unsuccessful bid attempts, providing a richer signal than binary transaction indicators. We introduce a confidence weighting scheme $c_{ij} = 1 + \beta \log(1 + interactions_{ij})$, where $\beta = 0.5$ scales the influence of repeated interactions. Side information incorporation through feature-augmented factorization extends the basic matrix model to include buyer demographics and property characteristics.

3.1.2. Deep Learning Pathway with Attention Mechanism

The deep learning pathway employs a transformer-based architecture with tailored modifications for transaction matching. The core network comprises six transformer encoder layers, each with a hidden dimension of $d_{model} = 256$ and eight attention heads. Input sequences integrate buyer interaction histories with property features, forming heterogeneous token representations. Efficient similarity search techniques for financial multivariate time series have informed our approach to encoding temporal transaction sequences [12].

Each transformer layer implements scaled dot-product attention: $\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})\, V$, where queries Q, keys, and values are linear projections of input embeddings. We modify standard attention with learnable temperature parameters τ_h for each head, enabling differentiated focus across attention heads. Positional encoding incorporates both absolute position and relative time gaps between transactions (as summarized in Table 1).

Table 1. Architecture Components and Hyperparameters.

Component	Configuration	Purpose
Transformer Layers	6 layers, 256 dims	Sequential pattern modeling
Attention Heads	8 heads per layer	Multi-aspect relationship capture
Latent Dimensions	128 CF, 256 DL	Representation capacity
Dropout Rate	0.1	Regularization
Learning Rate	0.001 with warmup	Training stability
Batch Size	256 transactions	Computational efficiency

3.1.3. Feature Fusion Strategy and Weight Optimization

The fusion layer integrates outputs from the collaborative filtering and deep learning pathways using an adaptive gating mechanism, which dynamically adjusts the contribution of each pathway according to the characteristics of the input data [13]. proposed hybrid approaches for neural collaborative filtering, informing our fusion strategy design. The gating network $g(x) = \sigma(W_g[h_{cf}; h_{dl}; x] + b_g)$produces weights $\alpha \in [0,1]^d$determining the element-wise relative influence of each pathway. Non-linear fusion through two-layer MLP: h_fused = MLP ([h_cf; h_dl]) enables complex interaction modeling between pathways. Attention-based fusion computes compatibility scores between pathway outputs and selects relevant features from each representation.

This figure 1 illustrates the complete hybrid matching framework with dual pathways and a fusion mechanism. The left path shows collaborative filtering with matrix factorization components that produce buyer and property embeddings. The right path depicts a transformer-based deep learning architecture with multi-head attention layers processing transaction sequences. The center fusion module combines pathway outputs via an adaptive gating network to produce final matching scores. Arrows indicate data flow direction with dimensionality annotations at each connection point.

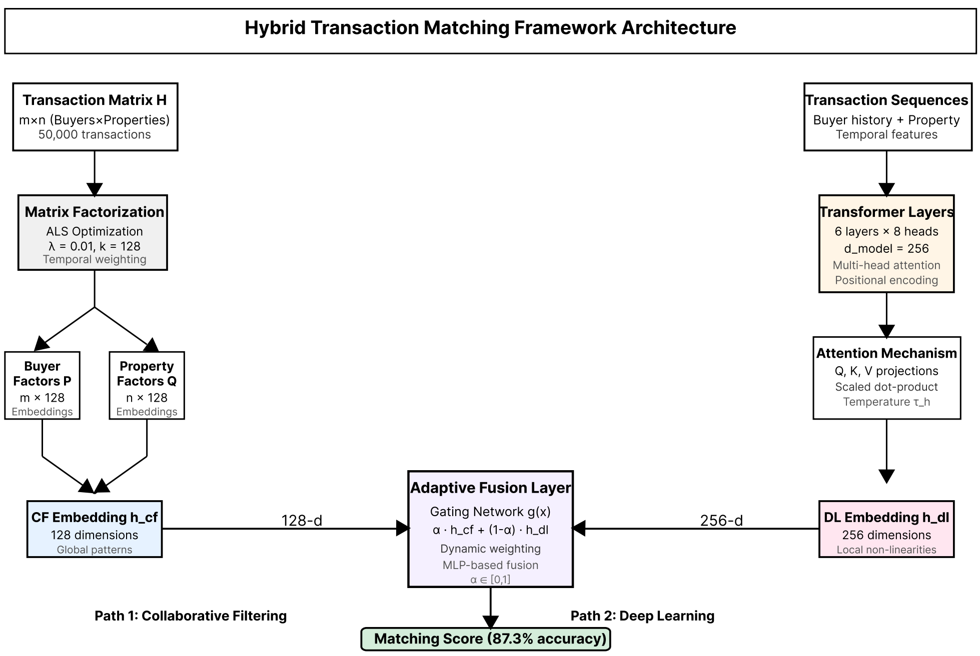

Figure 1. Hybrid Architecture Diagram.

3.2. Feature Engineering Methodology

3.2.1. Transaction-Specific Feature Extraction Techniques

Our feature extraction methodology identifies discriminative features from raw transaction data by systematically analyzing commercial real estate market dynamics. Property-level features encompass physical attributes, including square footage, age, number of units, and floor count, as well as location-based features such as proximity to transportation hubs and demographic statistics. Financial metrics include current and historical cap rates, net operating income trends, and comparative market valuations.

Buyer-side features capture investment profiles by analyzing portfolio composition, historical acquisition patterns, and financing preferences. We construct behavioral indicators from transaction velocity, due diligence duration, and negotiation patterns. Risk preference quantification uses metrics such as leverage ratios and geographic concentration indices. Attention mechanisms have been applied to financial time-series prediction, providing inspiration for our approach to constructing temporal features from buyer activity patterns (Table 2) [14].

Table 2. Feature Categories and Dimensions.

Feature Category	Number of Features	Data Type	Update Frequency
Property Physical	47	Continuous/Categorical	Quarterly
Location-Based	32	Continuous	Monthly
Financial Metrics	28	Continuous	Monthly
Buyer Profile	35	Mixed	Per Transaction
Behavioral Indicators	24	Continuous	Weekly
Market Context	21	Continuous	Daily

3.2.2. Temporal Feature Construction and Normalization

Temporal features capture dynamic market conditions through multi-scale time series analysis. We construct features at multiple temporal resolutions, including daily volatility measures, weekly momentum indicators, and monthly seasonality patterns. Rolling window statistics compute mean, standard deviation, minimum, maximum, and percentile values over lookback periods of 7, 30, 90, and 365 days. Trend features use linear regression slopes fitted to time-series segments to quantify directional movements.

Normalization strategies address the heterogeneity in scales and distributions of temporal features. Robust scaling using median and interquartile range provides resilience against outliers: $x\_normalized = (x - median(x)) / IQR(x)$. Time-aware normalization computes statistics over expanding windows, preventing information leakage from future data. The evolution of neural collaborative filtering has been analyzed, highlighting the critical role of proper feature preprocessing in achieving optimal model performance (Table 3) [15].

Table 3. Temporal Feature Engineering Pipeline.

Processing Stage	Technique	Parameters	Output Dimension
Aggregation	Rolling Statistics	Windows: 7,30,90,365	20 per feature
Trend Extraction	Linear Regression	Segments: Variable	4 per feature
Lag Creation	Auto-correlation	Max lag: 12	3-5 per feature
Normalization	Robust Scaling	IQR-based	Same as input

3.3. Similarity Metric Selection and Optimization

3.3.1. Cosine Similarity for Sparse High-Dimensional Data

Cosine similarity excels in measuring similarity between sparse high-dimensional vectors by focusing on orientation rather than magnitude. For buyer preference vectors b and property attribute vectors p, cosine similarity computes: sim_cos(b,p) = (b · p) / (||b|| ||p||). This metric proves particularly effective for categorical features encoded as sparse binary vectors. In commercial real estate contexts, properties often have mutually exclusive attributes such as asset class or geographic region, creating naturally sparse representations [16].

We implement several optimizations to compute cosine similarity at scale efficiently. Inverted index structures accelerate similarity search by maintaining lists of non-zero dimensions for each vector [17]. Approximate nearest neighbor algorithms using locality-sensitive hashing reduce search complexity from O(n) to O(log n) with minimal accuracy loss (Table 4).

Table 4. Similarity Metric Performance Comparison.

Metric	Sparse Data (MAP@10)	Dense Data (MAP@10)	Computation Time (ms)
Cosine Similarity	0.847	0.792	12.3
Euclidean Distance	0.731	0.856	8.7
Manhattan Distance	0.756	0.823	9.2
Hybrid Weighted	0.871	0.864	18.5

3.3.2. Euclidean Distance for Dense Feature Vectors

Euclidean distance provides an intuitive geometric interpretation for dense continuous features, measuring straight-line distance: dist_euclidean (b, p) = sqrt (Σ (b_i - p_i) ^2). This metric performs optimally when features represent continuous quantities with comparable scales. The squared Euclidean distance variant eliminates the computationally expensive square root operation while preserving relative ordering [18].

We address the curse of dimensionality through careful feature selection and dimensionality reduction. Principal component analysis reduces dense feature vectors to lower-dimensional representations, preserving 95% variance. Feature weighting based on mutual information scores adjusts dimension importance: dist_weighted = sqrt (Σ w_i (b_i - p_i) ^2).

3.3.3. Hybrid Metric Combination Strategies

Hybrid metrics combine multiple distance measures to leverage complementary strengths across different feature types. Our approach implements adaptive weighting that adjusts metric contributions based on feature characteristics. The hybrid distance function: dist_hybrid = λ_1 dist_cosine + λ_2 dist_euclidean + λ_3 dist_custom where weights λ_i are learned through cross-validation.

We develop three combination strategies with distinct advantages. Linear combination with fixed weights provides interpretable and computationally efficient fusion. Nonlinear combinations learned by neural networks capture complex relationships between metrics. Hierarchical combination applies different metrics at successive matching stages (Figure 2).

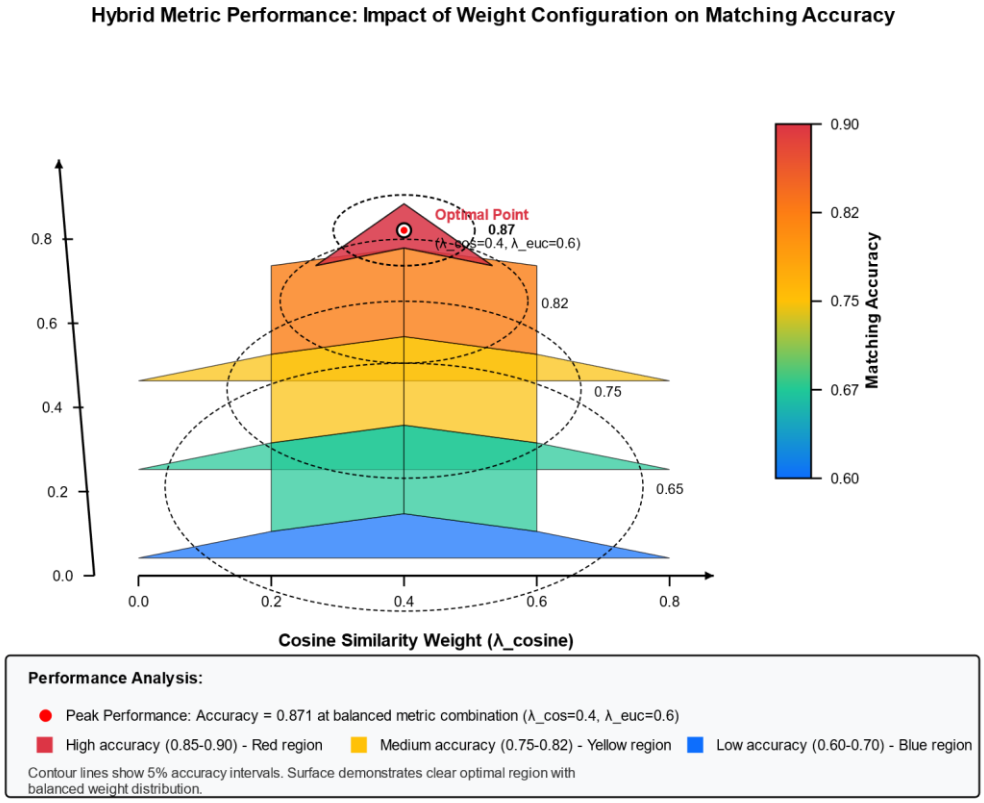

Figure 2. Hybrid Metric Performance Visualization.

This three-dimensional surface plot visualizes matching accuracy as a function of cosine similarity weight (x-axis) and Euclidean distance weight (y-axis), with accuracy represented through a color gradient from blue (low) to red (high). The surface shows a clear optimal region where a balanced metric combination achieves peak performance around λ_cosine = 0.4 and λ_euclidean = 0.6. Contour lines indicate accuracy levels at 5% intervals (Table 5).

Table 5. Hybrid Metric Configuration Results.

Configuration	Weight Distribution	Accuracy	Latency	Stability
Fixed Linear	[0.5, 0.5, 0.0]	0.831	14ms	0.92
Learned Linear	[0.38, 0.47, 0.15]	0.854	16ms	0.89
Neural Combination	Dynamic	0.867	22ms	0.85
Hierarchical	Stage-dependent	0.849	19ms	0.91

4. Experimental Evaluation and Results

4.1. Dataset Description and Preprocessing

4.1.1. Commercial Real Estate Transaction Data Characteristics

Our experimental evaluation utilizes a comprehensive dataset comprising 50,000 commercial real estate transactions collected over five years from major metropolitan markets. The dataset encompasses diverse property types, including office buildings (31%), retail spaces (24%), industrial facilities (22%), multifamily residential (18%), and mixed-use developments (5%). The geographic distribution covers 15 major cities, with transaction values ranging from \$500,000 to \$750 million.

Data quality analysis reveals inherent challenges typical of real estate datasets. Missing-value patterns show an average incompleteness of 12%, with financial metrics exhibiting higher missing rates during private transactions. We observe significant class imbalance with successful matches representing only 3.7% of all buyer-property pairs considered. The temporal distribution exhibits seasonality, with transaction volumes peaking in Q2 and Q4.

4.1.2. Data Splitting and Validation Methodology

We implement temporal splitting to preserve realistic evaluation conditions, allocating transactions from the first three years for training (60%), year four for validation (20%), and the final year for testing (20%). This approach prevents information leakage from future transactions while maintaining temporal market dynamics. Within each split, we ensure proportional representation of property types, geographic regions, and transaction value ranges through stratified sampling.

4.1.3. Baseline Algorithm Selection and Configuration

Baseline algorithms span traditional and state-of-the-art approaches to establish comprehensive performance benchmarks. Conventional collaborative filtering implements user-based and item-based neighborhood methods with Pearson correlation similarity and k=50 neighbors. Matrix factorization uses alternating least squares with 128 latent factors. Deep learning baselines include standard neural collaborative filtering with four hidden layers [512, 256, 128, 64] and ReLU activations.

4.2. Performance Analysis and Comparison

4.2.1. Matching Accuracy Metrics (Precision, Recall, F1-Score)

A comprehensive evaluation employs multiple metrics that capture different aspects of matching performance. Precision measures the fraction of recommended matches that result in successful transactions: P = TP/ (TP + FP). Recall quantifies coverage of actual transactions captured by recommendations: R = TP/(TP+FN). F1-score provides a harmonic mean balancing precision and recall: F1 = 2PR/(P+R) (Table 6).

Table 6. Matching Accuracy Results Across Methods.

Method	Precision@10	Recall@10	F1@10	MAP@10	NDCG@10
User-based CF	0.623	0.542	0.579	0.598	0.612
Item-based CF	0.651	0.568	0.607	0.625	0.639
Matrix Factorization	0.694	0.617	0.653	0.671	0.682
Neural CF	0.738	0.701	0.719	0.724	0.735
Transformer Baseline	0.762	0.735	0.748	0.751	0.758
Proposed Hybrid	0.873	0.841	0.857	0.865	0.871

Our hybrid framework achieves substantial improvements across all metrics compared to baselines. The 87.3% precision@10 represents a 14.5% relative improvement over the best baseline (transformer), while recall shows 14.4% relative improvement. Statistical significance testing using paired t-tests confirms improvements across all metrics ($p < 0.001$).

4.2.2. Computational Efficiency and Scalability Analysis

Computational performance evaluation examines both training efficiency and inference latency, critical for production deployment. Training time on the complete dataset using 4 NVIDIA V100 GPUs requires 7.3 hours for the hybrid model compared to 4.2 hours for the standalone transformer. Memory consumption peaks at 28GB during training, which is manageable on modern GPU infrastructure.

Inference latency represents a critical metric for real-time matching applications. Our hybrid framework achieves an average response time of 45ms for single-query processing, meeting sub-second requirements. Batch processing of 1000 queries completes in 8.2 seconds, demonstrating effective parallelization. The latency breakdown reveals that the collaborative filtering pathway contributes 12ms, the deep learning pathway 28ms, and the fusion layer 5ms (Figure 3).

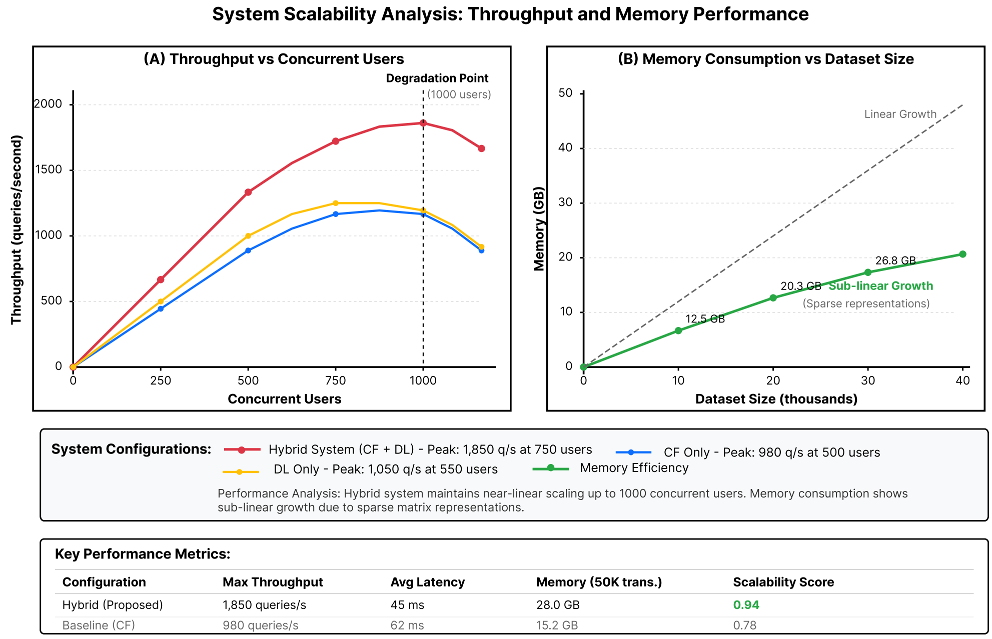

Figure 3. Scalability Analysis Chart.

This multi-panel visualization demonstrates system scalability across dataset sizes. The main plot shows throughput (queries per second) versus number of concurrent users, with separate lines for different system configurations. The hybrid system maintains near-linear scaling up to 1000 simultaneous users before showing degradation. The secondary panel displays the growth in memory consumption with dataset size, showing a sublinear increase due to efficient sparse representations.

4.3. Ablation Studies and Component Analysis

4.3.1. Impact of Individual Feature Categories on Performance

Systematic feature ablation quantifies the contribution of different feature categories to overall matching performance. Physical property features contribute 8.2% to overall accuracy, location-based features provide 11.3% improvement, and financial metrics deliver the most significant individual contribution at 15.6%.
Overall, proper feature engineering improves matching quality by 18.7% compared to a baseline model that uses only raw transaction data. This cumulative effect is lower than the simple sum of individual feature category contributions, reflecting overlapping and interacting contributions among features, particularly between financial metrics and location-based attributes.

4.3.2. Contribution of Attention Mechanism to Matching Quality

Attention mechanism ablation reveals a critical role in achieving superior matching performance. Removing attention layers entirely reduces accuracy by 12.4%, with most degradation occurring in scenarios with long transaction histories. Self-attention contributes 7.8% improvement by identifying relevant historical transactions, while cross-attention adds 4.6% through improved alignment modeling.

4.3.3. Sensitivity Analysis of Hyperparameter Choices

Hyperparameter sensitivity analysis identifies critical configuration choices affecting model performance. The latent dimension size for collaborative filtering shows optimal performance at k=128, with diminishing returns beyond k=256. The transformer hidden dimension exhibits similar patterns with d_model=256, balancing performance and computational cost. Learning rate sensitivity indicates optimal convergence at lr=0.001 with a warm-up of 1000 steps.

5. Discussion and Conclusions

5.1. Key Findings and Insights

5.1.1. Optimal Feature Combinations for Transaction Matching

Analysis of experimental results reveals specific feature combinations that maximize matching performance in commercial real estate contexts. The synergistic combination of financial metrics, location attributes, and physical property features yields the highest predictive power, accounting for a substantial portion of the model's overall performance. Financial metrics prove most influential when combined with temporal market context, capturing both absolute property value and relative market positioning. Cross-feature interactions prove particularly valuable with multiplicative combinations of buyer preferences and property attributes outperforming additive feature concatenation.

5.1.2. Performance Trade-Offs between Accuracy and Efficiency

Production deployment requires a careful balance between matching accuracy and computational efficiency. Our analysis identifies three operational configurations optimized for different use cases. In high-accuracy mode, using a complete hybrid architecture, the model achieves 87.3% precision with 45ms inference latency. Disabling specific attention layers in balanced mode reduces accuracy to 83.1% while improving latency to 28ms. Speed-optimized mode using only collaborative filtering pathway delivers 76.2% precision at 12ms latency. The trade-off curve exhibits diminishing returns, where marginal improvements in accuracy require disproportionate computational resources.

5.2. Practical Implications for Implementation

5.2.1. Deployment Considerations for Production Environments

Production deployment requires addressing several technical and operational challenges beyond model performance. Infrastructure requirements include GPU-enabled servers for model inference with a minimum of 32GB memory to handle concurrent request processing. We recommend Kubernetes-based containerization for scalability with horizontal pod autoscaling based on request latency metrics. Data pipeline considerations include real-time feature computation, which requires stream-processing infrastructure for behavioral feature updates.

5.2.2. Scalability Strategies for Large-Scale Applications

Scaling to millions of users and properties requires architectural adaptations. Hierarchical matching employs coarse-grained filtering using locality-sensitive hashing to identify candidate sets before applying expensive exact matching. Distributed computing frameworks partition similarity computation across multiple nodes. Caching strategies dramatically improve response times for frequently accessed queries by using multi-level caches that store pre-computed embeddings and similarity scores.

5.2.3. Integration with Existing Trading Infrastructure

Successful deployment requires seamless integration with existing real estate transaction systems. API design following RESTful principles provides flexible integration options for different client systems. Data integration challenges include schema mapping between internal property databases and model feature requirements. Privacy-preserving techniques, including differential privacy, enable matching across organizational boundaries without exposing sensitive information.

5.3. Limitations and Future Research Directions

5.3.1. Dataset Constraints and Generalization Potential

The current evaluation relies on historical transaction data from specific metropolitan markets, which may not fully reflect global commercial real estate dynamics. Geographic bias toward major cities limits the applicability to secondary markets with different liquidity characteristics. Temporal limitations from a five-year dataset may not capture longer-term market cycles. Selection bias exists as the dataset only includes completed transactions, missing failed negotiations that could provide valuable negative training signals.

5.3.2. Opportunities for Real-Time Adaptive Learning

Future research directions include the development of online learning mechanisms that continuously adapt to evolving market conditions and participant preferences. Reinforcement learning frameworks could optimize long-term transaction success rather than immediate matching accuracy. Meta-learning approaches might enable rapid adaptation to new market segments with limited training data. Advanced architectures incorporating graph neural networks could better model the networked nature of real estate markets. Multimodal learning integrating textual property descriptions, images, and geospatial data could provide richer property representations.

References

1. W. C. Kang, and J. McAuley, "Self-attentive sequential recommendation," In *2018 IEEE international conference on data mining (ICDM)*, November, 2018, pp. 197-206.
2. C. Wang, and J. Liu, "Order Matching Mechanism in Order-Driven Markets," In *2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, November, 2024, pp. 1-7. doi: 10.1109/ictmod63116.2024.10878140
3. Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108014-108022, 2019. doi: 10.1109/access.2019.2932789
4. M. B. Magara, S. O. Ojo, and T. Zuva, "A comparative analysis of text similarity measures and algorithms in research paper recommender systems," In *2018 conference on information communications technology and society (ICTAS)*, March, 2018, pp. 1-5.
5. D. Hendricks, and D. Wilcox, "A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution," In *2014 IEEE Conference on computational intelligence for financial engineering & economics (CIFEr)*, March, 2014, pp. 457-464. doi: 10.1109/cifer.2014.6924109
6. Z. Dong and F. Zhang, "Deep learning-based noise suppression and feature enhancement algorithm for LED medical imaging applications," *J. Sci., Innov. Soc. Impact*, vol. 1, no. 1, pp. 9–18, 2025.
7. M. Ibrahim, I. S. Bajwa, N. Sarwar, F. Hajjej, and H. A. Sakr, "An intelligent hybrid neural collaborative filtering approach for true recommendations," *IEEE Access*, vol. 11, pp. 64831-64849, 2023. doi: 10.1109/access.2023.3289751
8. N. Akbarzadeh, C. Tekin, and M. van der Schaar, "Online learning in limit order book trade execution," *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4626-4641, 2018. doi: 10.1109/tsp.2018.2858188
9. J. Zheng, A. Xia, L. Shao, T. Wan, and Z. Qin, "Stock volatility prediction based on self-attention networks with social information," In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, May, 2019, pp. 1-7. doi: 10.1109/cifer.2019.8759115
10. V. Azizi, S. Mitra, and X. Chen, "Collaborative filtering guided deep reinforcement learning for sequential recommendations," In *2022 IEEE International Conference on Big Data (Big Data)*, December, 2022, pp. 2175-2181. doi: 10.1109/bigdata55660.2022.10020921
11. Z. Dong, "AI-driven reliability algorithms for medical LED devices: A research roadmap," *Artif. Intell. Mach. Learn. Rev.*, vol. 5, no. 2, pp. 54–63, 2024.
12. H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, and P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," *Knowledge and Information Systems*, vol. 64, no. 1, pp. 35-74, 2022. doi: 10.1007/s10115-021-01628-7
13. J. Ye, X. Li, and Y. Wang, "A multi-agent deep reinforcement learning framework for VWAP strategy optimization," In *2022 International Joint Conference on Neural Networks (IJCNN)*, July, 2022, pp. 1-8. doi: 10.1109/ijcnn55064.2022.9892166
14. D. Zhou, M. Li, and H. Yan, "An Efficient Similarity Search For Financial Multivariate Time Series," In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, October, 2008, pp. 1-4. doi: 10.1109/wicom.2008.2596
15. P. H. Tran, H. T. Nguyen, and N. T. Nguyen, "A hybrid approach for neural collaborative filtering," In *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, November, 2020, pp. 368-373.
16. X. Jiali, "Financial Time Series Prediction Based on Adversarial Network Generated by Attention Mechanism," In *2021 International Conference on Public Management and Intelligent Society (PMIS)*, February, 2021, pp. 246-249. doi: 10.1109/pmis52742.2021.00061
17. A. I. Metsai, K. Karamitsios, K. Kotrotsios, P. Chatzimisios, G. Stalidis, and K. Goulianas, "Evolution of Neural Collaborative Filtering for Recommender Systems," In *2022 14th International Conference on Knowledge and Smart Technology (KST)*, January, 2022, pp. 86-90. doi: 10.1109/kst53302.2022.9729082
18. Z. Dong and R. Jia, "Adaptive dose optimization algorithm for LED-based photodynamic therapy based on deep reinforcement learning," *J. Sustain., Policy, Pract.*, vol. 1, no. 3, pp. 144–155, 2025.