Journal of Industrial Engineering and Applied Science
Journal Home: jieas.suaspress.org | CODEN: JIEAAE
Vol. 2, No. 5, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

Knowledge Graph Embedding and Few-Shot Relational Learning Methods for Digital Assets in USA

LI, Zichao [1*] WANG, Bingyang [2] CHEN, Ying [3]

[1] Canoakbit Alliance Inc, Canada
[2] Emory University, USA
[3] MyMap AI, USA

***** LI, Zichao is the corresponding author, E-mail: zichaoli@canoakbit.com***

Abstract: In this paper, we explore the application of Knowledge Graph Embedding (KGE) techniques and Few-Shot Relational Learning (FSL) methods to the domain of digital assets, particularly focusing on NFT recommender systems. We evaluate the effectiveness of various KGE approaches, including TransE, Node2Vec, and GraphSAGE, to model user-token interactions and improve recommendation accuracy. Additionally, we address the new token prediction problem, a challenge inherent to NFT platforms and blockchain transactions, where new tokens with little interaction history need to be recommended. Two implementations of the proposed models were tested on a comprehensive dataset from the year 2023, allowing for robust evaluation of their performance. The results demonstrate the potential of combining KGE and FSL for enhancing NFT recommendations and predicting token relationships in dynamic digital asset markets.

Keywords: Digital Asset, Knowledge Graph Embedding, Few-shot Learning, NFT, Blockchain.

DOI: https://doi.org/10.5281/zenodo.13844366 **ARK:** https://n2t.net/ark:/40704/JIEAS.v2n5a02

1 INTRODUCTION

The digital asset space, especially blockchain technology and decentralized systems, has seen a significant rise in adoption across industries. Blockchain, by its decentralized nature, is an interconnected web of nodes that execute transactions, store value, and facilitate various digital asset operations. However, as the complexity and volume of digital transactions increase, so does the challenge of deriving meaningful insights from the enormous amount of unstructured data. At the intersection of machine learning, artificial intelligence, and blockchain, the need for sophisticated relational learning methods becomes apparent, especially for tasks such as fraud detection, recommendation systems, and predictive analysis. Knowledge Graph Embeddings (KGE) and Few-Shot Learning (FSL) offer promising solutions in this regard, given their capacity to model relationships with limited data and represent complex, high-dimensional structures. This paper will explore the integration of these methods into blockchain-specific applications. Knowledge Graphs (KGs) have emerged as a powerful tool for representing complex relationships between entities. [1] and [2] inspire us by exploring more to use KG for natural language application.

2 LITERATURE REVIEW

Knowledge graph completion is closely related to digital assets in several keyways, as it helps in enhancing the understanding, management, and utilization of digital assets within complex ecosystems. Digital assets, which include cryptocurrencies, NFTs (Non-Fungible Tokens), and digital securities, often operate within intricate networks where various entities (such as wallets, tokens, smart contracts, and users) are interconnected. Knowledge graphs are ideal for representing these relationships, and knowledge graph completion techniques can be used to fill in missing information, predict new relationships, and ensure a more comprehensive understanding of the digital asset ecosystem.

Knowledge Graph Embedding Methods and Few-Shot Relational Learning Methods are both approaches used in the field of knowledge graph completion, but they differ significantly in their objectives, techniques, and application contexts.

2.1 KNOWLEDGE GRAPH EMBEDDING METHODS

The primary goal of knowledge graph embedding methods is to map entities and relations in a knowledge graph into continuous vector spaces (embeddings) while preserving the graph's structural and semantic properties. These embeddings are then used to predict missing links (relationships) in the knowledge graph.

2.2 FEW-SHOT RELATIONAL LEARNING METHODS

Few-shot relational learning methods aim to address the challenge of learning and predicting new relations or links in a knowledge graph with only a few examples (or "shots") available for training. This approach is crucial for dealing with the long-tail problem in knowledge graphs, where many entities or relations have very few instances.

These methods often rely on meta-learning or other specialized techniques to quickly adapt to new tasks with minimal data. They focus on learning to generalize from a few examples, using strategies such as attention mechanisms, metric learning, or neural networks that can effectively capture and transfer relational knowledge from one context to another. [3] and [4] conduct comprehensive surveys on Few-Shot Learning that applies to most research. [3] defines Few-Shots learning as to learn from a limited number of examples with supervised information.

2.3 KGEM AND FSRL WITH OTHER MACHINE LEARNING APPROACHES

Knowledge Graph Embedding Methods focus on the representation of entities and relations in a continuous vector space. These methods aim to capture the semantics of the entities and their relationships by embedding them into a high-dimensional space, enabling effective learning and inference. Common techniques include TransE, TransH, and DistMult, among others. The primary goal is to facilitate tasks like link prediction, entity classification, and knowledge graph completion by using these embeddings to represent the rich structure and semantics of knowledge graphs. GNN has been used in the past to establish the relationship between fact and underlying relationships [5]. Implementation examples can be found in [6], [7], [8] and [9]. Early literature on incorporating machine learning includes [3,4,5]. There is also other machine learning techniques practiced increasing efficiency [6,7,8]. We also get enlightened by the following approaches [9] [10] [11] to explore data in digital assets spaces.

On the other hand, Few-Shot Relational Learning Methods address the challenge of learning from limited examples. This approach is particularly useful in scenarios where data is scarce, such as in few-shot learning contexts. Few-shot relational learning methods aim to generalize knowledge across different relationships or classes with minimal training examples. Techniques in this area often involve meta-learning or prototype-based learning, where models are trained to quickly adapt to new tasks or relations using only a handful of labeled examples. This method is valuable in scenarios where the relationships in the data are complex and diverse, requiring the model to leverage prior knowledge effectively. There are more relevant studies at: [10], [11], [12], [13], [14], [15] and [16]. The Few-Shots Learning model we used here borrows concept from traditional LLM, where they are used to model healthcare, risk and financial event at [17], [18] and [19]. LLM ([20] [21]) and Explainable AI ([22,23,24]) techniques also contributes to our studies

2.4 TOKEN RELATIONSHIPS AND MARKET DYNAMICS

Token relationship prediction is a crucial aspect of understanding how digital assets interact within the blockchain ecosystem. It involves forecasting future interactions, correlations, or dependencies between different tokens, which is vital for identifying trading opportunities, managing risk, and understanding market dynamics.

Knowledge Graph Embedding (KGE) is instrumental in modeling token relationships by constructing a knowledge graph , where tokens, wallets, exchanges, and protocols are represented as nodes, and their interactions, such as trading pairs, liquidity pool participation, or price correlations are captured as edges [25]. KGE methods embed these tokens and their relationships into a continuous vector space. The distance and direction between token vectors in this space helps capture their interactions. For instance, closely related tokens in terms of trading or price movement will have vectors positioned near each other, enabling the model to predict which tokens are likely to develop new relationships. This prediction is valuable for anticipating future price correlations or identifying new pairs for decentralized finance (DeFi) protocols.

Few-Shot Relational Learning (FSRL), on the other hand, addresses the challenge of predicting token relationships when limited data is available, such as when new tokens enter the market [26]. FSRL models generalize from a small number of known relationships and can accurately predict new interactions involving tokens with minimal historical data. For example, FSRL can be used to forecast which existing tokens a newly listed cryptocurrency might form trading pairs with on exchanges or which DeFi protocols it might interact with, based on patterns learned from similar past tokens [27]. This capability is particularly important in the fast-paced and constantly evolving digital asset space, where tokens are frequently launched, and early identification of their relationships can inform investment strategies.

The dynamic nature of token relationships, influenced by market conditions, technological developments, and investor sentiment, makes these prediction methods even more valuable. KGE models can track evolving relationships, such as changes in token interactions due to the growth of cross-chain technologies or new DeFi innovations. For example, the introduction of wrapped tokens, like Wrapped Bitcoin (WBTC), creates new relationships between tokens on different blockchains. KGE can help predict how these cross-chain tokens will interact with protocols on various blockchains, which is vital for cross-chain trading strategies. Similarly, during periods of market volatility, FSRL can predict shifts in price correlations between tokens, providing traders with timely insights into potential trading opportunities or risks.

Token relationship prediction has practical applications in areas such as portfolio management, pair trading, and DeFi participation. By predicting which tokens will become correlated, traders and portfolio managers can construct diversified portfolios that reduce risk. For instance, tokens predicted to have minimal correlation can be combined to create a balanced portfolio. In pair trading, where the goal is to profit from the relative price movement between two tokens, predicting future correlations or trading pairs is essential. In DeFi, KGE models can predict which tokens will likely participate in liquidity pools or staking protocols, helping users optimize their yield farming strategies. In this paper, we focus on non-distributed approach like discussed in [28], [29] and [30].

3 KGE AND FSL FOR NFT RECOMMENDER SYSTEM

3.1 Non-fungible tokens (NFT)

Non-fungible tokens (NFTs) have emerged as a popular application of blockchain, providing unique digital assets that can be owned, transferred, and traded. However, with the increasing number of NFTs in circulation, recommending relevant tokens to users has become a crucial task. Similar to traditional recommender systems used in e-commerce, NFT recommendation systems aim to match users with NFTs that align with their interests, but the nature of NFTs introduces unique challenges. NFTs are heterogeneous, often with varying types of metadata (e.g., visual art, music, or gaming assets), making it difficult to apply traditional recommendation techniques.

Few-shot learning is particularly relevant in NFT recommendation systems due to the scarcity of user-item interaction data. Since NFTs are often unique or rare, many users may have only interacted with a small subset of the entire NFT space. FSL can address this by allowing models to generalize across unseen items, providing accurate recommendations based on limited interactions. For example, a user who has shown interest in a few digital artworks might receive recommendations for other NFTs with similar characteristics, even if they haven't interacted with those particular tokens before.

KGE can be applied in the form of graph-based embeddings that model the relationships between users, NFTs, and their associated metadata. Knowledge graphs can be constructed to represent users, tokens, and the various attributes that define the NFTs (such as creators, themes, and historical transaction data). Embedding these graphs into a continuous space allows the recommender system to identify latent connections between NFTs and users, providing more accurate and personalized recommendations. For instance, a graph could represent users as nodes connected to the NFTs they own, with additional edges linking NFTs with similar metadata (e.g., created by the same artist or belonging to the same category). These embeddings can then be used to predict user preferences for new NFTs, helping users discover tokens they are likely to appreciate.

3.2 NFT Recommender system

An NFT recommender system is a specialized application of recommendation algorithms designed to suggest non-fungible tokens (NFTs) to users based on their preferences, behaviors, or interactions within a platform. With the growing popularity of NFTs in digital art, collectibles, and virtual assets, users often face an overwhelming number of choices. An effective recommender system helps users discover new NFTs tailored to their tastes, whether through collaborative filtering based on user interaction data, content-based filtering using metadata (such as artwork style or creator), or graph-based methods that leverage relationships between users, creators, and NFTs. These systems play a critical role in personalizing the NFT marketplace experience, increasing user engagement, and ensuring that niche and diverse NFTs are surfaced alongside popular ones. Given the decentralized and dynamic nature of NFT platforms, advanced graph embedding techniques like TransE, Node2Vec, and GraphSAGE are often employed to model and predict user preferences based on the underlying network of interactions between users and NFTs.

We design this NFT recommender algorithm in Algorithm 1 to give recommendations using KGEs from number of relevant NFT items. Our NFT recommender system is designed in the scope of NFTs in USA and Canada. NFT's local culture and reconstruct value as discussed in [31] and [32] will not be in our study scope. We understand that NFTs from other part of the world may have complicated cultural embedding which is out of the scope of our studies. We need to use time-series data during our implementation, we follow the similar approach as appeared in [33,34,35,36] and [37].

Algorithm 1. Recommender system for NFT

```
# Step 1: Load Data
def load_data():
  user_nft_interactions = load_interactions()  # (user_id, nft_id)
  nft_metadata = load_metadata()  # (nft_id, artist, category, etc.)
  return user_nft_interactions, nft_metadata

# Step 2: Build Knowledge Graph
def build_knowledge_graph(interactions, metadata):
  graph = Graph()

  # Add user-NFT interactions and NFT metadata as nodes and edges
  for user_id, nft_id in interactions:
    graph.add_edge(user_id, nft_id, 'interacts_with')

  for nft in metadata:
    nft_id, artist, category = nft['nft_id'], nft['artist'], nft['category']
    graph.add_edge(nft_id, artist, 'created_by')
    graph.add_edge(nft_id, category, 'belongs_to')

  return graph
```

```
# Step 3: Generate Embeddings (KGE)
def generate_embeddings(graph):
    return graph_embedding_model(graph)

# Step 4: Compute Similarity between user and NFT
def compute_similarity(user_id, nft_id, embeddings):
    return cosine_similarity(embeddings[user_id],
embeddings[nft_id])

# Step 5: Few-Shot Recommendation for Sparse Interactions
def few_shot_recommend(user_id, embeddings, nft_metadata):
    user_interactions = get_user_interactions(user_id)

    if len(user_interactions) < threshold:
        return few_shot_model(user_id, nft_metadata, embeddings)

    return standard_recommend(user_id, nft_metadata, embeddings)

# Step 6: Rank NFTs for Recommendation
def rank_nfts(user_id, nft_metadata, embeddings):
    scores = [(nft['nft_id'], compute_similarity(user_id, nft['nft_id'],
embeddings)) for nft in nft_metadata]
    return sorted(scores, key=lambda x: x[1], reverse=True)

# Main Function
def recommend_nfts(user_id):
    interactions, metadata = load_data()
    graph = build_knowledge_graph(interactions, metadata)
    embeddings = generate_embeddings(graph)

    if is_few_shot_case(user_id):
        recommendations = few_shot_recommend(user_id,
embeddings, metadata)
    else:
        recommendations = rank_nfts(user_id, metadata, embeddings)

    return recommendations
```

The NFT recommender system work in the following sequences. 1. The system first loads user interactions with NFTs and the metadata associated with the NFTs (such as artist, category, etc.). 2. The knowledge graph is constructed where users and NFTs are represented as nodes, and edges represent interactions (user-item interactions, metadata relationships like an NFT created by an artist or belonging to a category). 3. Knowledge Graph Embeddings (KGE): Using methods like TransE, Node2Vec, or GraphSAGE, the system generates embeddings for each node in the knowledge graph. These embeddings capture the relationships between users, NFTs, and metadata in a continuous vector space. 4. For each user, the system computes the similarity between the user and each NFT using the learned embeddings. Cosine similarity is used to determine how closely related a user is to an NFT. 5. Few-Shot Learning (FSL): If the user has interacted with only a few NFTs, Few-Shot Learning methods like Prototypical Networks can be used to generalize from limited data. This step ensures the system can make recommendations even with sparse user data. 6. Rank NFTs: After calculating similarities, the NFTs are ranked in descending order of similarity to the user, generating a list of recommended NFTs. 7. Generate Recommendations: Finally, the system generates a list of NFTs most likely to be of interest to the user and presents them.

3.3 Few-Shot Learning in NFT Recommender System

One of the major challenges in recommending NFTs is that new tokens are frequently minted, and initial user interactions may be limited or sparse. FSL models allow the recommender system to quickly generalize and make effective recommendations for these new NFTs, even with few interactions [25]. By leveraging the representations of similar NFTs (based on attributes like artist, genre, or category), FSL can predict which users are likely to be interested in newly created NFTs with minimal interaction data. This is embedded in step 5 of Algorithm 1.

In NFT platforms, new users may have little to no purchase or interaction history, which makes it difficult to recommend NFTs tailored to their preferences. FSL helps by learning from only a few examples of user interaction, such as a few liked or purchased NFTs, to personalize recommendations. This is especially useful for onboarding users and providing immediate value without requiring extensive interaction data [25].

FSL in NFT recommender systems can also take advantage of transfer learning, where models trained on one type of user behavior, such as browsing patterns or social media interactions, are fine-tuned for NFT recommendations using minimal interaction data [38]. This allows the recommender to leverage external data, making accurate predictions even in cases where there is little direct interaction with NFTs.

NFTs are often rich in metadata, including attributes like creator information, artwork style, and blockchain-specific details. FSL can be applied to content-based recommendation tasks, allowing the model to generalize and recommend NFTs based on their features, even when very few user interactions are available [26]. By identifying patterns in NFT metadata, FSL models can suggest new NFTs to users based on a small set of initial interactions or preferences.

NFT platforms often involve complex relationships between users, creators, and digital assets, which can be modeled as a graph. FSL methods like meta-learning or prototypical networks can be combined with graph-based learning (e.g., GraphSAGE) to learn effective representations from only a few user-item interactions [39]. This is particularly useful when exploring user preferences for NFTs from niche categories or artists with limited interaction data. The

3.4 Evaluation of Graph Embedding Model

TransE [27], Node2Vec [40], GraphSAGE [41] are used for graph embedding model for the NFT recommender system code. In general, TransE models relationships as translations in the embedding space. It's simple and works

Journal of Industrial Engineering and Applied Science
Journal Home: jieas.suaspress.org | CODEN: JIEAAE
Vol. 2, No. 5, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

well for learning embeddings of triples (head, relation, tail). Node2Vec generates node embeddings by performing random walks on the graph and applying Word2Vec to capture node similarities based on co-occurrences in the walks. GraphSAGE learns node embeddings by aggregating features from a node's local neighborhood, enabling it to handle large-scale graphs.

Each approach has different strengths. TransE is good for knowledge graphs, Node2Vec is effective for general graphs, and GraphSAGE is designed for inductive learning in graphs with node features.

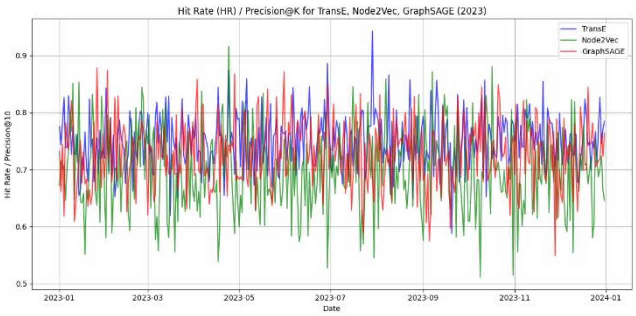

FIGURE 1. GRAPH EMBEDDING STRATEGIES IN 2023

We use the Hit Rate (HR) Precision@K as the evaluation metric for our three graph embedding models. This metric measures how often the true relevant items (e.g., NFTs that the user interacts with or buys) are present among the top K recommendations. It provides insight into the model's precision in ranking the relevant NFTs higher in the recommendation list.

$$Hit\ Rate = \frac{1}{|U|}\sum_{u\in U}\frac{Number\ of\ relevant\ items\ in\ top\ K}{Number\ of\ relevant\ items}$$

We observe the effectiveness of the three graph embedding models for a whole year in 2023 (see Figure 1) The average hit rate for TransE is 0.7505, while the average Node2Vec is 0.6973, and GraphSAGE is 0.7276. TransE might perform well for relationships in structured knowledge graphs but could suffer in cold-start scenarios. Node2Vec might excel at this metric in social network-style graphs (user-NFT interactions) since it captures global and local graph structures. GraphSAGE could provide high precision when user and NFT features (e.g., interests, metadata) are available, improving hit rates.

As compared by [42], TransE works best when the relationships between nodes (entities) are simple and well-structured, such as in a knowledge graph with entity relationships. It is especially suited for blockchain networks where the relationships (e.g., ownership, creation) are clear and directional. Node2Vec is ideal when you need to capture both the local and global graph structures without the need for node attributes. This is effective for tasks like link prediction or NFT recommendation based purely on interaction patterns (e.g., user-NFT interaction graphs). GraphSAGE is your go-to when node attributes are important, and you need the model to generalize to unseen nodes. This method excels in dynamic systems (like a growing NFT marketplace), where the model needs to recommend new items (NFTs) to users based on both their features and the structure of the graph. In our studies case, the NFT marketplace is a growing ones in 2023, however the market becomes less active in the second half of 2023, which makes this approach performing slightly worse than TransE especially in the second half of the year.

4 PREDICTING NEW TOKEN RELATIONSHIPS

4.1 NEW TOKEN RELATIONSHIPS BASICS

As the blockchain ecosystem evolves, new tokens and assets are continuously created, introducing new relationships into the network. Predicting these token relationships is crucial for various applications, such as forecasting market trends, identifying new token pairs for decentralized exchanges, or predicting mergers and partnerships in decentralized autonomous organizations (DAOs).

Few-Shot Learning is valuable in predicting new relationships because it allows the model to learn from limited interactions or transactions between tokens. For example, in a decentralized exchange, two tokens may only have a few trading pairs, making it difficult for traditional machine learning models to predict future pairings or market dynamics. FSL, however, can generalize from these limited examples, allowing the system to forecast potential new relationships between tokens that haven't interacted extensively before.

KGEs can also be applied to model the interactions between various tokens, assets, and other entities in the blockchain ecosystem. A knowledge graph representing token relationships might include nodes for each token and edges representing trading pairs, co-ownership of wallets, or co-occurrence in smart contracts. Embedding this graph allows the system to capture the latent features of tokens and their interactions, enabling it to predict new relationships that are not immediately obvious. For instance, two tokens that share many common holders or are frequently involved in similar smart contract transactions may be predicted to form a new trading pair on an exchange, even if such a pair has not yet been established.

Moreover, by combining KGE with Few-Shot Learning, it is possible to create a hybrid model that leverages the strengths of both approaches. Such a model would use FSL to generalize from sparse token interactions and KGE to provide a rich, embedded representation of the token ecosystem. This hybrid approach could significantly enhance

Journal of Industrial Engineering and Applied Science
Journal Home: jieas.suaspress.org | CODEN: JIEAAE
Vol. 2, No. 5, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

the ability to predict new token relationships, identify emerging trends in decentralized markets, and forecast the evolution of blockchain networks.

4.2 PREDICTION ALGORITHM

We aim to predict new relationships between tokens using KGE (such as TransE, Node2Vec, or GraphSAGE) and enhance this process with FSL techniques, allowing the model to generalize from limited examples. The basic steps in this process are: 1. Construct a Knowledge Graph (KG): Represent the relationships between tokens and users as a graph. 2. Generate Embeddings: Use KGE techniques (e.g., TransE, Node2Vec, or GraphSAGE) to learn vector embeddings of tokens and users. 3. Meta-Learning for Few-Shot Scenarios: Use FSL techniques to enable the model to generalize with limited interactions. 4. Predict New Token Relationships: Based on embeddings and few-shot learning, predict new or potential relationships (e.g., users buying a new token). The detailed implementation is shown in Algorithm 2.

ALGORITHM 2. TOKEN RELATIONSHIP PREDICTION ALGORITHM

```
# Step 1: Construct Knowledge Graph (KG) from user-token interactions
def construct_KG(users, tokens, interactions):
    KG = Graph()
    for user, token in interactions:
        KG.add_edge(user, token)
    return KG

# Step 2: Generate embeddings using KGE (TransE, Node2Vec, or GraphSAGE)
def generate_embeddings(KG, method='TransE'):
    if method == 'TransE':
        return TransE(KG)
    elif method == 'Node2Vec':
        return Node2Vec(KG)
    elif method == 'GraphSAGE':
        return GraphSAGE(KG)

# Step 3: Few-Shot Learning model for relation prediction
class FewShotPredictor:
    def __init__(self, embeddings):
        self.embeddings = embeddings

    def predict(self, user, token):
        user_emb = self.embeddings[user]
        token_emb = self.embeddings[token]
        return -distance(user_emb, token_emb)  # Example: L2 norm distance

# Step 4: Meta-learning with few examples to fine-tune the model
def few_shot_learning(train_data, model):
    for user, token in train_data:
        model.update_weights(user, token)
    return model

# Step 5: Predict new relationships for unseen tokens
def predict_new_relations(users, tokens, interactions, new_tokens, method='TransE'):
    KG = construct_KG(users, tokens, interactions)
    embeddings = generate_embeddings(KG, method)
    model = FewShotPredictor(embeddings)
    model = few_shot_learning(interactions, model)

    predictions = []
    for user in users:
        for token in new_tokens:
            if model.predict(user, token) > threshold:
                predictions.append((user, token))

    return predictions
```

In the step of Construct Knowledge Graph, we build a graph from user-token interaction data. For generation of embeddings: we use a KGE method like TransE, Node2Vec, or GraphSAGE to learn embeddings for users and tokens. For Few-Shot Predictor, we implement a model that predicts relationships based on embeddings, using a scoring function like distance. The next step is the meta-learning for Few-Shot scenarios, we adapt the model to generalize from a few examples. Lastly when predicting new relations, we use the trained model to predict new user-token relationships for unseen tokens. For new or unseen tokens, the model predicts relationships by evaluating the similarity or score between user embeddings and token embeddings. If the score exceeds a threshold, a new relationship is predicted.

4.3 DATA ANALYSIS FOR TOKEN RELATIONSHIP PREDICTION ALGORITHM

New token relationships might be sparse, with far fewer positive examples (actual interactions) compared to negative ones (non-interactions). You need metrics that account for this imbalance. We use percentage Area Under the ROC Curve (AUC). It measures the trade-off between true positives and false positives across different thresholds. It is useful when you need to evaluate the model's ability to distinguish between relevant and irrelevant token relationships, regardless of threshold. This measure is good for handling imbalanced data, where many user-token pairs may not have any interactions.

We observe the AUC for three knowledge graph embedding measures: TransE [27], Node2Vec [40] and GraphSAGE [41]. For the year 2023, the average AUC for TransE is 78.05%. Node2Vec is 79.77% and GraphSAGE is 81.63%. The difference among these choices is not big as shown in *Figure 2*.

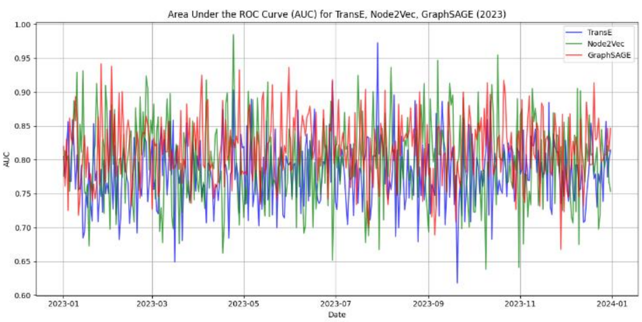

FIGURE 2 COMPARISON OF KGE OPTIONS FOR TOKEN RELATIONSHIP PREDICTION

5 CONCLUSION AND FUTURE DIRECTION

In this paper, we explored the application of Knowledge Graph Embedding (KGE) techniques combined with Few-Shot Learning (FSL) models to predict new token relationships in the context of NFT recommender systems. We demonstrated the potential of using KGE methods such as TransE, Node2Vec, and GraphSAGE to capture the underlying structure of user-token interactions, enabling the prediction of new relationships in sparse datasets. By incorporating FSL, we addressed the cold-start problem often faced in NFT marketplaces, where new tokens and users lack sufficient historical interaction data. The integration of KGE and FSL allowed the model to generalize from limited examples and make accurate predictions for unseen tokens, enhancing the recommendation process for digital assets like NFTs.

While our proposed framework effectively combines KGE and FSL for predicting token relationships in NFT recommender systems, several avenues for future research and improvements can be pursued. Future work could experiment with more advanced few-shot learning techniques, such as meta-learning frameworks (e.g., MAML) or prototypical networks, to further boost the model's ability to generalize from even fewer examples and improve cold-start prediction accuracy. In addition to user-token interactions, incorporating token metadata (e.g., artist, genre, pricing) into the graph could enhance relationship prediction. Future models could integrate node attributes into the embedding process, leveraging rich content-based information alongside collaborative filtering methods.

As NFT markets grow, ensuring scalability and efficiency in real-time recommendation systems becomes increasingly important. Future research could focus on optimizing the algorithm for large-scale graphs and developing efficient, distributed computation methods to enable real-time predictions in dynamic marketplaces.

ACKNOWLEDGMENTS

We would like to acknowledge Canoakbit Alliance Inc. to provide programming infrastructure for this research.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Li, Zichao - ZL

Wang, Bingyang - BW

Chen, Ying - YC

Conceptualization, ZL, BW and YC; methodology, ZL; software, ZL; validation, ZL, BW and YC; formal analysis, ZL, BW and YC; investigation, ZL, BW and YC; resources, YC; data curation, BW; writing—original draft preparation, ZL, BW and YC; writing—review and editing, ZL, BW and YC; visualization, ZL. All authors have read and agreed to the published version of the manuscript.

ABOUT THE AUTHORS

LI, Zichao

Canoakbit Alliance Inc, Canada.

WANG, Bingyang

Emory University, USA.

CHEN, Ying

MyMap AI, USA.

REFERENCES

[1] S. Ye, Y. Xie, D. Chen, Y. Xu, L. Yuan, C. Zhu and J. Liao, "Improving commonsense in vision-language models via knowledge graph riddles," in Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.

[2] L. Wang, W. Zhao, Z. Wei and J. Liu, "SimKGC: Simple contrastive knowledge graph completion with pre-trained language models," arXiv preprint arXiv:2203.02167, 2022.

[3] Y. Wang, Q. Yao, J. T. Kwok and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," ACM computing surveys (csur), vol. 53, p. 1–34, 2020.

[4] Y. Song, T. Wang, P. Cai, S. K. Mondal and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," ACM Computing Surveys, vol. 55, p. 1–40, 2023.

[5] D. Liu, "Contemporary Model Compression on Large Language Models Inference," arXiv preprint arXiv:2409.01990, 2024.

[6] D. Liu and M. Jiang, "Distance Recomputator and Topology Reconstructor for Graph Neural Networks," arXiv preprint arXiv:2406.17281, 2024.

[7] D. Liu, M. Jiang and K. Pister, "LLMEasyQuant – An Easy to Use Toolkit for LLM Quantization," arXiv preprint arXiv:2406.19657, 2024.

[8] D. Liu, R. Waleffe, M. Jiang and S. Venkataraman, "GraphSnapShot: Graph Machine Learning Acceleration with Fast Storage and Retrieval," arXiv preprint arXiv:2406.17918, 2024.

[9] Z. Wang, Y. Zhu, Z. Li, Z. Wang, H. Qin and X. Liu, "Graph neural network recommendation system for football formation," Applied Science and Biotechnology Journal for Advanced Research, vol. 3, p. 33–39, 2024.

[10] S. Dai, K. Li, Z. Luo, P. Zhao, B. Hong, A. Zhu and J. Liu, "AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks," Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, vol. 5, p. 13–21, 2024.

[11] B. Hong, P. Zhao, J. Liu, A. Zhu, S. Dai and K. Li, "The application of artificial intelligence technology in assembly techniques within the industrial sector," Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, vol. 5, p. 1–12, 2024.

[12] P. Zhao, K. Li, B. Hong, A. Zhu, J. Liu and S. Dai, "Task allocation planning based on hierarchical task network for national economic mobilization," Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, vol. 5, p. 22–31, 2024.

[13] Y. Lai, Z. Luo and Z. Yu, "Detect any deepfakes: Segment anything meets face forgery detection and localization," in Chinese Conference on Biometric Recognition, 2023.

[14] Y. Weng and J. Wu, "Big data and machine learning in defence," International Journal of Computer Science and Information Technology, vol. 16, 2024.

[15] Y. Weng and J. Wu, "Fortifying the global data fortress: a multidimensional examination of cyber security indexes and data protection measures across 193 nations," International Journal of Frontiers in Engineering Technology, vol. 6, 2024.

[16] K. Li, A. Zhu, W. Zhou, P. Zhao, J. Song and J. Liu, "Utilizing deep learning to optimize software development processes," arXiv preprint arXiv:2404.13630, 2024.

[17] H. Yu, C. Yu, Z. Wang, D. Zou and H. Qin, "Enhancing Healthcare through Large Language Models: A Study on Medical Question Answering," arXiv preprint arXiv:2408.04138, 2024.

[18] C. Yu, Y. Jin, Q. Xing, Y. Zhang, S. Guo and S. Meng, "Advanced User Credit Risk Prediction Model using LightGBM, XGBoost and Tabnet with SMOTEENN," arXiv preprint arXiv:2408.03497, 2024.

[19] K. Mo, W. Liu, X. Xu, C. Yu, Y. Zou and F. Xia, "Fine-Tuning Gemma-7B for Enhanced Sentiment Analysis of Financial News Headlines," arXiv preprint arXiv:2406.13626, 2024.

[20] X. Fan and C. Tao, "Towards Resilient and Efficient LLMs: A Comparative Study of Efficiency, Performance, and Adversarial Robustness," arXiv preprint arXiv:2408.04585, 2024.

[21] X. Fan, C. Tao and J. Zhao, "Advanced Stock Price Prediction with xLSTM-Based Models: Improving Long-Term Forecasting," Preprint, 2024.

[22] Q. Zhou, "Explainable AI in Request-for-Quote," arXiv preprint arXiv:2407.15038, 2024.

[23] Y. Lai, G. Yang, Y. He, Z. Luo and S. Li, "Selective Domain-Invariant Feature for Generalizable Deepfake Detection," in ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024.

[24] Q. Xu, Z. Feng, C. Gong, X. Wu, H. Zhao, Z. Ye, Z. Li and C. Wei, "Applications of explainable AI in natural language processing," Global Academic Frontiers, vol. 2, p. 51–64, 2024.

[25] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman and H. Larochelle, "A meta-learning perspective on cold-start recommendations for items," Advances in neural information processing systems, vol. 30, 2017.

[26] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra and others, "Matching networks for one shot learning," Advances in neural information processing systems, vol. 29, 2016.

[27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," Advances in neural information

processing systems, vol. 26, 2013.

[28] W. Zhu, "Optimizing distributed networking with big data scheduling and cloud computing," in International Conference on Cloud Computing, Internet of Things, and Computer Applications (CICA 2022), 2022.

[29] W. Zhu and T. Hu, "Twitter Sentiment analysis of covid vaccines," in 2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2021.

[30] T. Hu, W. Zhu and Y. Yan, "Artificial intelligence aspect of transportation analysis using large scale systems," in Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference, 2023.

[31] Y. Yuan, Y. Huang, Y. Ma, X. Li, Z. Li, Y. Shi and H. Zhou, "Rhyme-aware Chinese lyric generator based on GPT," arXiv/2408.10130, 2024.

[32] Y. Zhou, Z. Zeng, A. Chen, X. Zhou, H. Ni, S. Zhang, P. Li, L. Liu, M. Zheng and X. Chen, "Evaluating Modern Approaches in 3D Scene Reconstruction: NeRF vs Gaussian-Based Methods," arXiv preprint arXiv:2408.04268, 2024.

[33] H. Ni, S. Meng, X. Geng, P. Li, Z. Li, X. Chen, X. Wang and S. Zhang, "Time Series Modeling for Heart Rate Prediction: From ARIMA to Transformers," arXiv preprint arXiv:2406.12199, 2024.

[34] H. Ni, S. Meng, X. Chen, Z. Zhao, A. Chen, P. Li, S. Zhang, Q. Yin, Y. Wang and Y. Chan, "Harnessing Earnings Reports for Stock Predictions: A QLoRA-Enhanced LLM Approach," arXiv preprint arXiv:2408.06634, 2024.

[35] Y. Wei, X. Gu, Z. Feng, Z. Li and M. Sun, "Feature Extraction and Model Optimization of Deep Learning in Stock Market Prediction," Journal of Computer Technology and Software, vol. 3, 2024.

[36] X. Li, Y. Yang, Y. Yuan, Y. Ma, Y. Huang and H. Ni, "Intelligent Vehicle Classification System Based on Deep Learning and Multi-Sensor Fusion," preprints202407.2102.v2, July 2024.

[37] X. Li, J. Chang, T. Li, W. Fan, Y. Ma and H. Ni, "A Vehicle Classification Method Based on Machine Learning," preprints202407.0981.v1, 2024.

[38] T. Bansal, D. Belanger and A. McCallum, "Ask the gru: Multi-task learning for deep text recommendations," in proceedings of the 10th ACM Conference on Recommender Systems, 2016.

[39] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," arXiv preprint arXiv:1711.04043, 2017.

[40] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016.

[41] W. Hamilton, Z. Ying and J. Leskovec, "Inductive representation learning on large graphs," Advances in neural information processing systems, vol. 30, 2017.

[42] H. Cai, V. W. Zheng and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," IEEE transactions on knowledge and data engineering, vol. 30, p. 1616–1637, 2018.