machine learning & knowledge extraction

MDPI

Article

Temporal Relational Graph Convolutional Network Approach to Financial Performance Prediction

Brindha Priyadarshini Jeyaraman *, Bing Tian Dai * and Yuan Fang

School of Computing and Information Systems, 80 Stamford Rd., Singapore 178902, Singapore; yfang@smu.edu.sg
* Correspondence: jeyaramanp.2021@engd.smu.edu.sg (B.P.J.); btdai@smu.edu.sg (B.T.D.)

Abstract: Accurately predicting financial entity performance remains a challenge due to the dynamic nature of financial markets and vast unstructured textual data. Financial knowledge graphs (FKGs) offer a structured representation for tackling this problem by representing complex financial relationships and concepts. However, constructing a comprehensive and accurate financial knowledge graph that captures the temporal dynamics of financial entities is non-trivial. We introduce FintechKG, a comprehensive financial knowledge graph developed through a three-dimensional information extraction process that incorporates commercial entities and temporal dimensions and uses a financial concept taxonomy that ensures financial domain entity and relationship extraction. We propose a temporal and relational graph convolutional network (RGCN)-based representation for FintechKG data across multiple timesteps, which captures temporal dependencies. This representation is then combined with FinBERT embeddings through a projection layer, enabling a richer feature space. To demonstrate the efficacy of FintechKG, we evaluate its performance using the example task of financial performance prediction. A logistic regression model uses these combined features and social media embeddings for performance prediction. We classify whether the revenue will increase or decrease. This approach demonstrates the effectiveness of FintechKG combined with textual information for accurate financial forecasting. Our work contributes a systematic FKG construction method and a framework that utilizes both relational and textual embeddings for improved financial performance prediction.

Keywords: knowledge graph; finance; BERT; tweets; text; LSTM; RGCN; news; NLP; commercial entities; concept entities

Citation: Jeyaraman, B.P.; Dai, B.T.; Fang, Y. Temporal Relational Graph Convolutional Network Approach to Financial Performance Prediction. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 2303–2320. https://doi.org/10.3390/make6040113

Received: 5 September 2024
Revised: 27 September 2024
Accepted: 1 October 2024
Published: 10 October 2024

1. Introduction

In the finance industry, textual data play a vital role in various financial business use cases. However, the representation of financial knowledge within these textual data is often unstructured, limiting its value and potential. Financial knowledge graphs have gained significant attention by providing a structured way to represent financial data. This structured nature enhances their applicability in various domains, including risk management [1], fraud detection [2], investment analysis [3], and financial decision-making.

Recent advancements in temporal knowledge graph (TKG) representation learning have shown significant promise for tasks that involve reasoning and forecasting on evolving knowledge. Our work builds upon several key contributions in this area. Cai et al. [4] propose a taxonomy for TKG representation learning methods and highlight various approaches for capturing temporal dynamics. He et al. [5] introduce the concept of a "Hip network" that leverages historical information for reasoning on TKGs. For temporal knowledge graph completion, Wang et al. [6] present MADE, a method using multicurvature adaptive embeddings, while Dong et al. [7] explore temporal inductive path neural networks. Liao et al. [8] investigate generative forecasting on TKGs with large language models, while Luo et al. [9] explore the potential of large language models for learning and forecasting with TKG completion. Ding et al. [10] introduce zrLLM for zero-shot

relational learning, and Wang et al. [11] propose IME, a method integrating multicurvature embeddings for TKG completion. Finally, Gastinger et al. [12] establish a baseline for temporal knowledge graph forecasting. Our work introduces a framework that combines temporal relational graph convolutional networks with financial performance prediction, offering a way to integrate temporal dynamics from financial data.

The first challenge in real-world applications is that knowledge graphs often face issues related to incomplete and inaccurate data. Cheng et al. [13] focus on structured news events extracted from a raw text corpus, constructing a large-scale knowledge graph called FinKG, which covers over 3000 companies listed in the Shanghai and Shenzhen stock exchanges. Despite this, the paper highlights the difficulties in creating a complete and accurate knowledge graph. Our approach uses scenario-based extractors to accurately capture complex relationships between entities in financial textual data, resulting in a more complete and accurate representation of financial information.

The second challenge is the lack of a systematic and repeatable approach to knowledge graph construction. Zehra et al. [14] present a financial knowledge graph-based financial report query system utilizing Wikidata and DBpedia, but the construction process lacks repeatability. Loster et al. [15] highlight challenges in the entity resolution, maintenance, and exploration of such graphs. Cheng et al. [16] propose a multi-modality graph neural network (MAGNN) but do not provide sufficient details on the construction process. Our paper proposes a systematic way of constructing a financial knowledge graph using scenario-based knowledge extractors, which ensures an accurate capture of complex relationships. A scenario-based knowledge extractor includes modules such as Financial Growth Extractor, Market Development Extractor, Contract Extractor, and Acquisition Extractor. These modules focus on specific scenarios to extract relevant financial information, accurately identifying key entities and relationships to ensure comprehensive and precise financial knowledge representation.

The third challenge is incorporating consistent domain-specific knowledge within the knowledge graphs. Knowledge graphs constructed from internet sources like Wikipedia have been explored in various domains [17–19]. Domain-specific graphs have also been developed in healthcare [20,21]. However, these approaches lack structured methods for knowledge extraction. K-BERT [22] injects domain knowledge into BERT but does not leverage domain-specific taxonomies for consistent extraction. Our approach incorporates domain knowledge through taxonomies and scenario-specific extraction methods.

Wang et al. [23] introduced the Financial Research Report Knowledge Graph (FR2KG) dataset in Chinese, focusing on domain-specific triples. Our work presents a generalized systematic approach applicable to any domain with a well-defined taxonomy, emphasizing a formal, standardized, and semantically enriched representation using domain-specific taxonomies. Wu et al. [24] proposed an audit information knowledge graph specific to a particular market, which limits generalizability. Our work uses a domain-specific taxonomy as the backbone, providing a formal and consistent representation regardless of market.

In the domain of temporal financial data representation, previous work has explored various temporal graph-based techniques. Messner et al. [25] introduced BoxTE for temporal knowledge graph completion (TKGC) using box embeddings. Xu et al. [26] proposed TEA-GNN with time-aware attention for TKGs. However, these approaches do not comprehensively represent temporal financial data. Our approach combines Relational Graph Convolutional Networks (RGCNs) [27] with Long Short-Term Memory (LSTM) to capture temporal patterns, providing a more nuanced representation.

Graph learning methods have seen applications in healthcare, particularly brain analysis for disease detection [28]. Bi et al. [29] developed a structure-adaptive graph neural network with temporal representation (TR-SAGNN) that addresses limitations of existing methods by incorporating a temporal attention learning module, which is similar to our temporal RGCN's ability to capture temporal dynamics in financial data.

To evaluate our FinTechKG, we adopt the task of financial performance prediction as a benchmark task. In the domain of financial performance prediction, Tao et al. [30] utilized

ConvLSTM networks for extracting stock price features and incorporating mutation points with graph convolutional networks (GCNs). Lam et al. [31] explored neural networks to integrate fundamental and technical analysis for financial performance prediction. Although effective, these approaches did not consider real-time trends from social information such as news or tweets, which can significantly impact financial entity performance.

Our approach uses tweets as social embeddings to capture real-time trends and demonstrates improved prediction accuracy by incorporating temporal features from social embeddings. The VGC-GAN model [32], an advanced adversarial framework utilizing multi-graph convolution for stock price forecasting, stands out by creating dynamic correlation graphs and integrating a GAN architecture with Multi-GCN and GRU. This methodology significantly improves prediction accuracy through a detailed analysis of inter-stock relationships and temporal dependencies, as demonstrated on various datasets.

While some prior work, like that from Elhammadi et al. [33], has focused on financial knowledge graph construction and semantic relationship discovery, it often neglects the temporal sequence of events. The lack of temporal representation may lead to outdated or irrelevant information in the knowledge graph, limiting its relevance and reliability in dynamic financial markets.

Zakhidov (2024) underscores the role of economic indicators like GDP, inflation, and employment in predicting market trends, advocating for a combination of quantitative and qualitative data for better analysis [34]. Olorunsola et al. (2024) link governance scores and financial metrics, such as GOPPAR, to corporate emission performance, offering insights into financial–environmental dynamics in the tourism industry [35]. Munir et al. (2024) highlight the influence of dividend-related metrics and financial indicators on stock price dynamics in Pakistan's non-financial sectors, emphasizing localized financial decisions [36].

Our paper presents a Relational Graph Convolutional Network (RGCN) [27] and LSTM [37] approach to enhance financial performance prediction. Further aggregating the temporal information into the LSTM [37] enables a more accurate prediction of financial entity performance, by integrating the comprehensive financial knowledge graph into the RGCNs [27] and training it at different timesteps.

The FintechKG knowledge graph extraction pipeline is shown in Figure 1. It comprises a reusable methodology that can be used to generate knowledge graphs on other textual data. (By identifying the taxonomy/concept entities for the domain).

The proposed approach, FintechKG, overcomes various challenges in financial knowledge graph construction and financial performance prediction by employing a systematic and comprehensive methodology as follows:

- Enriched Domain Knowledge: FintechKG captures the taxonomy of financial concepts from various financial statements such as income statements, balance sheets, and cash flow statements. This enriched domain knowledge provides a comprehensive understanding of financial concepts and relationships between entities.
- Effective Integration of Information: The proposed approach integrates textual embeddings (FinBERT) [38] with graph-based embeddings (RGCN) to exploit both textual knowledge and relational information encoded in the knowledge graph. This integration enhances the accuracy of financial entity performance predictions by mapping both embeddings into a shared latent space.
- Temporal Pattern Learning: The RGCN [27] embeddings from different time steps are used as input to an LSTM [37] (Long Short-Term Memory) model, capturing temporal dependencies and trends in financial data. This leads to more accurate predictions of financial performance. Our focus is to present an example of predicting whether a financial organization's revenue will increase or decrease. The same approach can be applied to predicting other financial indicators as well.

Overall, FintechKG offers a scalable and adaptable approach to constructing financial knowledge graphs, addressing challenges representing complex relationships through RGCN [27]. Its temporal awareness, enriched domain knowledge, and effective integration

of textual and graph-based information make it a valuable framework for financial decision-making and analysis.

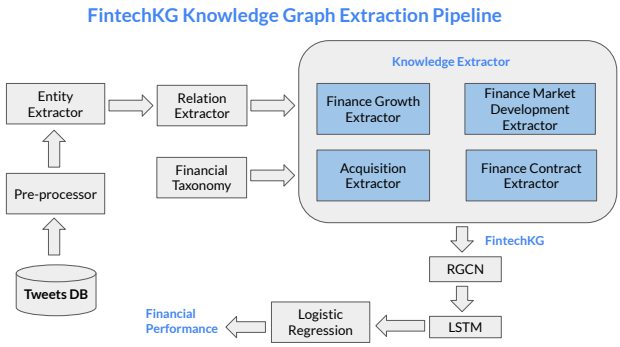

Figure 1. FintechKG knowledge graph extraction pipeline.

2. The Proposed Framework

In this section, we present the framework employed for constructing the FintechKG (Fintech knowledge graph) to perform Financial Entity prediction. Figure 1 illustrates the steps involved in the FintechKG Knowledge Graph Extraction and Prediction Pipeline. This framework combines the following:

- RGCN-based Representation: Representing FintechKG data at three timestamps using Relational Graph Convolutional Networks (RGCNs) [27] to capture relational information.
- FinBERT [38] Embeddings: Integrating textual information via pre-trained FinBERT [38], capturing semantic insights from financial news.
- Temporal Reasoning with LSTM: Employing Long Short-Term Memory (LSTM) [37] networks to model temporal dependencies within the RGCN representations.
- Projection Layer: Introducing a projection layer for both RGCN [27] and FinBERT [38] embeddings, enabling effective combination (e.g., linear or non-linear) into a unified latent space for prediction.

3. FintechKG Construction Phase

The pipeline follows a systematic process for extracting Concept Entities, Commercial Entities, and Relations from textual data. The input to the pipeline consists of sentences or paragraphs of text. The text undergoes pre-processing, including the removal of stop words and invalid characters. The pre-processed text is then fed into the BERT (Bidirectional Encoder Representations from Transformers) NER (Named Entity Recognition) Model [39] to extract entities. A rule-based algorithm is used to extract relations in the form of <subject, predicate, object> triples from the textual data.

Furthermore, we employ a Scenario-Based Knowledge Extractor that encompasses several extractors such as the Financial Growth Extractor, Finance Market Development Extractor, Finance Contract Extractor, and Acquisition Extractor. These extractors utilize specific scenarios to extract relevant financial information. The output of the Scenario-Based Knowledge Extractor contributes to the generation of the financial knowledge graph(FinTechKG).

The combination of these steps forms a comprehensive approach for constructing the FintechKG, enabling the extraction and organization of valuable financial information into a structured knowledge graph.

In this section, we define the key concepts in our methodology, where financial knowledge is extracted in three dimensions: Commercial Entity (ComE), Financial or Concept Entity (ConE), and temporal information (TI).

3.1. Commercial Entity (ComE)

The commercial entity refers to organizations' names. We use pre-defined ticker symbols to match with the tokens in the text and identify the occurrence of ticker symbols in the tweet. Ticker symbols are commonly used in financial text to refer to the performance of financial entities.

3.2. Financial or Concept Entity (ConE)

To create a taxonomy of financial concepts, we studied the structure of annual reports, balance sheets, and income statements, resulting in the identification of various financial concept entities. Examples of these entities include loss, operating profit, net sales, net profit, profit, earnings, operating loss, sales, net loss, pretax profit, cashflow, fair value, transaction, market share, diluted earnings, net interest, income, and total value. The taxonomy is illustrated in Figure 2.

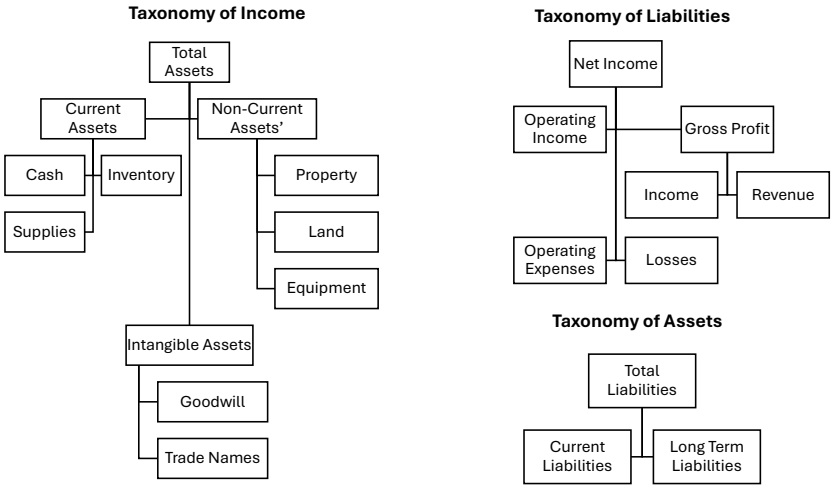

Figure 2. Taxonomy of income, liabilities, and assets.

3.3. Temporal Information (TI)

Temporal information is represented in various forms, such as date, first quarter, last quarter, year, recent quarter, second quarter, third quarter, fourth quarter, consecutive quarter, quarters, and last quarter.

There are different types of relations between the three dimensions. They are as follows:

Relation Type 1a: The source node can be a ComE, and the destination can be a ConE. The edge represents the relation, such as "increase" or "decrease".

Relation Type 1b: The source node can be a ComE, and the destination can be a TI (Temporal Node). The edge represents the ConE concatenated with its relation, for example, "profit increase in".

Relation Type 2: The source node can be a ComE, and the destination can also be another ComE. The edge represents the relation, such as "acquire" or "owns". This relation is referred to as a subset relation. The relation types are shown in Figure 3.

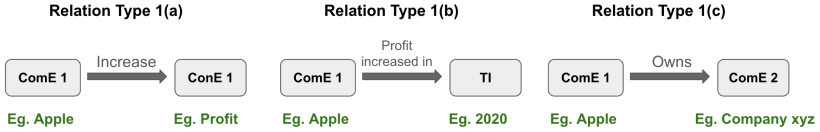

Figure 3. Relation types.

The pre-processor is an essential initial step in the extraction of the knowledge graph components [17,33]. It consists of a sequence of steps, including the removal of URLs,

hashtags, mentions, Twitter handles, emojis, smileys, stop-word removal, punctuation removal, lemmatization, and stemming.

3.4. Entity Extraction

The entity extraction process involves identifying various types of entities, including Commercial Entities (ComE), Concept Entities (ConE), time entities, ticker symbols, currencies, percentages, countries, and persons.

3.5. Extraction Using BERT Model

To extract entities from tweets, we utilize the BERT NER (Named Entity Recognition) model. The input to the model is the text from tweets, which is tokenized. The model classifies tokens into different entity types such as "DATE", "PERCENT", "ORG", "CARDINAL", "PERSON", "NORP", "GPE", and "MONEY". By analyzing the output of the model, we can identify Commercial Entities (ComE) from the entity type "ORG". Additionally, ticker symbols of financial organizations are identified from the tweets using this approach.

Figure 4 illustrates the architecture of the BERT model applied for Named Entity Recognition (NER). The model processes a tweet as input, tokenizing the text into individual tokens (Tok_1, Tok_2, …, Tok_n). Each token is fed into BERT, which generates embeddings (E(CLS), E_1, …, E_n) for each token. The output of BERT is then passed to classification layers that assign tags to each token, identifying named entities such as organizations (B-ORG), persons (B-PER), and non-entity tokens (O). This structure allows the model to effectively classify and recognize named entities within the unstructured textual data of the tweet. The following are the steps:

1. Input: A tokenized sequence (e.g., a tweet),

$$X = [[\text{CLS}], T_1, T_2, \ldots, T_n, [\text{SEP}]]$$

   where T_i represents each token.
2. Embeddings: For each token T_i, the embedding is as follows:

$$E(T_i) = \text{Token Embedding}(T_i) + \text{Position Embedding}(T_i)$$

3. BERT Encoding: BERT processes the embeddings through self-attention, producing contextual embeddings:

$$H = [h_1, h_2, \ldots, h_n]$$

4. Self-Attention: The self-attention mechanism calculates relationships between tokens:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

5. NER Classification: For each token T_i, BERT applies a linear layer:

$$z_j(T_i) = W_j h_i + b_j$$

6. Softmax: The logits $z_j(T_i)$ are transformed into class probabilities:

$$P(C_j|T_i) = \frac{\exp(z_j(T_i))}{\sum_{k=1}^{K} \exp(z_k(T_i))}$$

7. Output: The model predicts the entity label for each token T_i, selecting the class with the highest probability:

$$\hat{y}_i = \arg\max_j P(C_j|T_i)$$

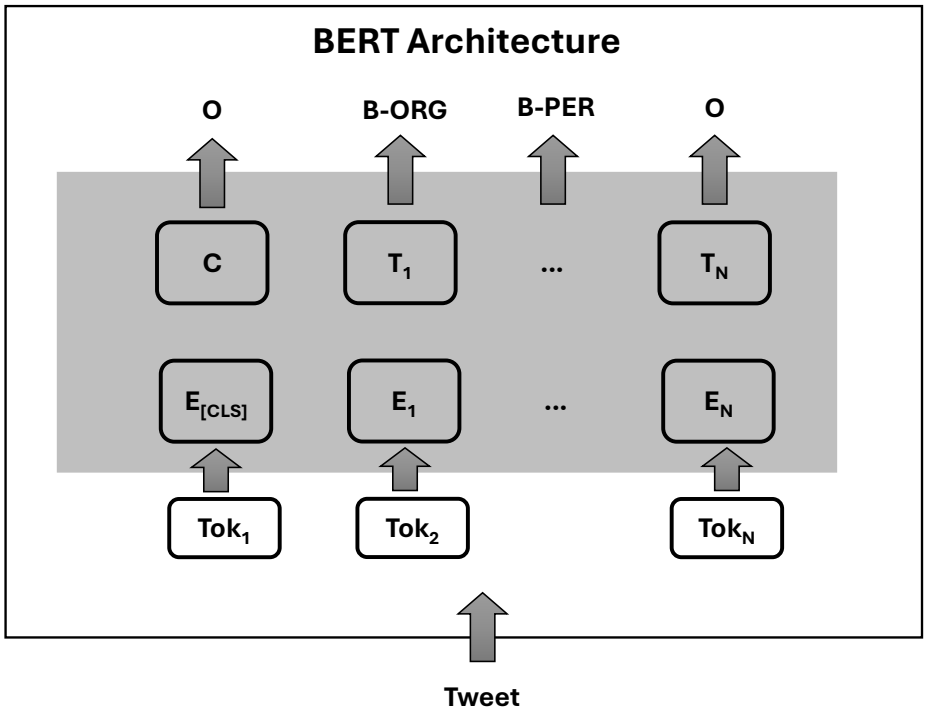

Figure 4. BERT NER model.

3.6. Extraction of Concept Entities

Concept Entities (ConE) are domain-specific in nature. In the case of tweets, we leverage financial reports such as income statements, balance sheets, and cash flow statements to identify a taxonomy of financial concepts, as illustrated in Figure 2. Pre-defined concepts from this taxonomy are then identified as Concept Entities (ConE).

This entity extraction pipeline can be applied to new domains by defining a relevant taxonomy and creating a list of pre-defined concept entities. This flexibility enables the FintechKG Extraction pipeline to accurately identify concepts and construct the knowledge graph for different domains.

3.7. Extraction of Temporal Entities

Temporal entities are extracted in the form of dates, months, weekdays, years, and quarters. We also perform extrapolation based on time by converting references to time, such as "last quarter", "previous quarter", "last year", "yesterday", and "last year", to the corresponding time entity derived from the current timestamp of the tweet.

Extraction of Currencies and Percentages

Tweets often contain informal representations of text, including currencies and percentages. To accurately extract currency values, we employ a custom Currency Extractor that matches informal representations of currencies and retrieves more precise currency values. For example, "3.9 mn" and "12.1 mln" both refer to "million". Similarly, a Percentage Extractor matches informal representations of percentages in tweets, such as "14 percent increase in operating profit".

3.8. Relation Extractor

The relation extraction process involves identifying relations between entities in the knowledge graph.

The first step is a rule-based approach using the Spacy NLP pipeline. The pipeline includes various components like tokenizer, tagger, parser, ner, etc., and enables POS tagging and dependency parsing. By defining rules to match the structure of the sentence, we can extract the "ROOT", "PREP", and "ADJ" components from the sentence structure.

These components are used to generate a list of relations. The frequency of these relations is calculated, and the top-N relations, representative of the domain, are selected.

For the financial tweet dataset, some of the pre-defined relations identified are increase, decrease, plan, develop, report, start, grow, sell, acquire, expand, own, announce, sign, agree, issue, and fell.

The second step involves using BERT to perform similarity matching of new relations with the pre-defined set of relations obtained from the rule-based approach. By obtaining the relation embeddings from the BERT model and applying mean pooling, we create a single vector encoding for the sentence embedding. We calculate the cosine similarity between the new relation and the pre-defined relations. If the similarity score exceeds a threshold, we replace the new relation with the corresponding relation from the pre-defined set. This ensures consistency and avoids redundancies in the representation of relations in the knowledge graph.

3.9. Knowledge Extractor

The Financial Growth Extractor module focuses on extracting relations related to financial growth, such as “Increase”, “Decrease”, “Rose to”, “Totalled”, “Fell”, “Jumped”, and “Dropped To”. This module aims to identify instances where there is a change or movement in financial indicators.

Table 1 presents a sample of the extraction process. In the first example, the Commercial Entity extracted is “Orion Pharma”, the Concept Entity is “Operating Profit”, and the relation is “increase”. The date “2004” and the percentage “42.5%” are also extracted and populated in the FintechKG.

Table 1. Samples of Financial Growth Extraction.

Tweet Example 1: Orion Pharma’s Operating Profit Increased by 42.5% from 2004	
Commercial Entity	Orion Pharma
Concept Entity	operating profit
Relation	increase
Date	2004
Percentage	42.5%

The Financial Market Development Extractor identifies relations related to market development activities, including “Plan”, “Develop”, “Reported”, “Start”, “Grow”, “Sell”, “Includes”, “Published”, “Sold”, “Won”, “Issued”, “Proposed”, “Declined By”, and “Forecasts”. This module can capture various market-related events and actions such as a company’s sales growth or plans to enter new markets.

The Acquisition Extractor focuses on extracting relations related to acquisitions, including “Acquire”, “Expand”, “Rose to”, “Own”, “Announced”, “Signed”, and “Bought”. This module aims to identify instances where a company acquires or expands its ownership of another entity, such as acquiring shares or announcing a new acquisition.

The Financial Contract Extractor extracts relations related to financial contracts, including “Contracted”, “Awarded”, “Sign”, and "Agree”. This module can identify instances where financial agreements, contracts, or awards are mentioned, such as signing a contract or being awarded a financial agreement.

These modules play a crucial role in extracting relevant information from financial tweets and populating the FintechKG with accurate and structured data, enabling a comprehensive understanding of financial events, trends, and relationships.

Figure 5 represents a knowledge graph generated from the information extracted from tweets. It shows relationships between various financial entities and events over time. For instance, Cargotec’s net sales exceeded EUR 2.3 billion in 2005, and on 5 September 2008, Nokia’s American Depositary shares fell by 8%. Orion Pharma’s operating profit increased by 42.5% in 2004, and Nordea Group’s operating profit increased by 18% in 2010. The knowledge graph links these financial entities (e.g., Cargotec, Nokia, Orion Pharma,

Nordea Group) with relevant actions or events (e.g., exceeded, shares fell, increased), showcasing a structured representation of financial data extracted from textual sources.

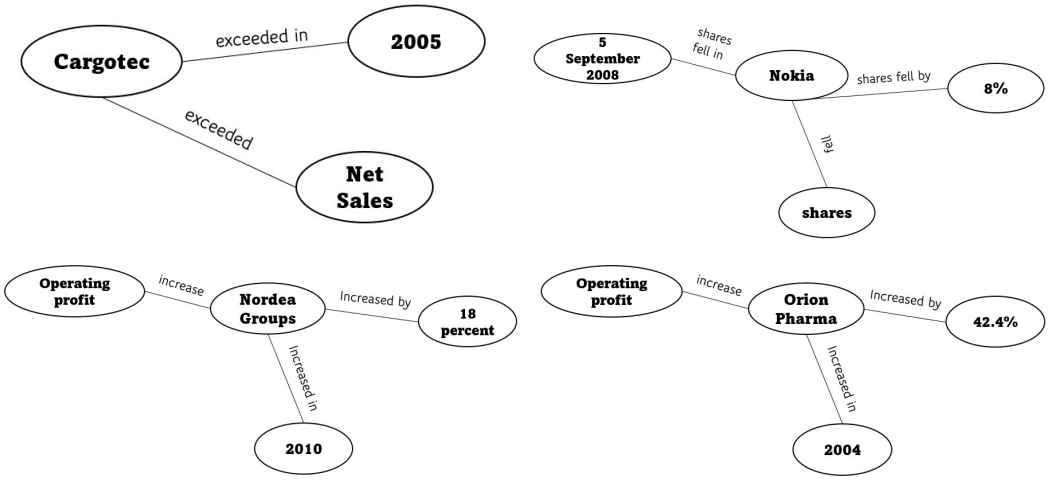

Figure 5. Knowledge graph for tweets: "In 2005 Cargotec's net sales exceeded EUR 2.3 billion.", "When this information was released on 5 September 2008, Nokia's American Depositary shares fell by 8%", "Orion Pharma 's operating profit increased by 42.5% from 2004". and "Nordea Group's operating profit increased in 2010 by 18 percent year-on-year to 3.64 billion euros and total revenue by 3 percent to 9.33 billion euros".

4. Temporal Relational Model

4.1. RGCN Model for Multiple Timesteps

In our methodology, a Relational Graph Convolutional Network (RGCN) model [27] is used. The constructed FintechKG is passed into the RGCN model at different timesteps. The RGCN model plays an important role in capturing the complex relationships between entities in the knowledge graph. It is illustrated in Figure 6.

The RGCN model is specifically designed to handle graph-structured data. Using FintechKG graph data, nodes represent commercial entities, and edges represent relations extracted from financial tweets, such as increase, decrease, sell, and acquire. Therefore, we can effectively capture the relational dependencies between commercial entities and exploit the contextual information encoded in the graph structure, which helps to improve the accuracy of financial performance predictions. The above RGCN model is designed to capture temporal information by considering graph data at three different time steps.

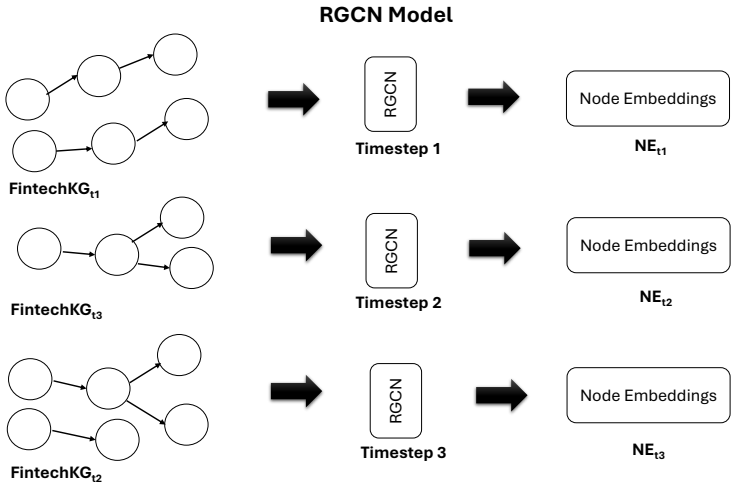

Figure 6. RGCN architecture capturing temporal information.

Architecture

The RGCN (Relational Graph Convolutional Network) model takes graph data with node features and edge features at three time steps and applies graph convolutions to learn node representations that incorporate temporal information. The model then uses these representations to make predictions on relations. Given a graph with node features and edge features at three time steps, the RGCN model can be mathematically formulated as follows:

Node Representation:

Source Nodes (Commercial Entities): each commercial entity node is associated with a node feature vector h_s. Destination Nodes (Concept Entities): each concept entity node is associated with a node feature vector h_d.

Edge Representation:

Edge Labels (Relations): each edge (between a source and destination) is associated with an edge feature vector e that represents the relation between the commercial entity and concept entity.

Relational Convolution:

The relational convolution operation updates the node representations based on the source nodes, edge labels, and destination nodes.

The update rule for the source node s connected to the destination node d through edge e is given by:

$$h_s = \sigma\left(\sum_{r \in R} \sum_{d \in N_r(s)} (h_d \cdot W_r^{(1)} \oplus e \cdot U_r^{(1)})\right) \tag{1}$$

In Equation (1),

$N_r(s)$ represents the set of destination nodes connected to source node s through relation r.
$W_r^{(1)}$ and $U_r^{(1)}$ are relation-specific weight matrices for the source node and edge, respectively.
$\oplus$ denotes concatenation.
σ is an activation function, such as ReLU or sigmoid, applied element-wise.

Stacked Convolutional Layers:

The RGCN typically consists of multiple stacked relational convolutional layers. Each layer refines the node representations by incorporating information from different levels of relational connections.

The update rule for the source node s in the $(l+1)$th layer is given by:

$$h_s^{(l+1)} = \sigma\left(\sum_{r \in R} \sum_{d \in N_r(s)} (h_d^{(l)} \cdot W_r^{(l)} \oplus e \cdot U_r^{(l)})\right) \tag{2}$$

where $h_d^{(l)}$ denotes the destination node representation in the l-th layer.

Readout and Prediction:

After the convolutional layers, a readout function is applied to aggregate the final node representations into a fixed-length graph representation. This graph representation can then be used for predicting the relations between commercial entities and concept entities.

The prediction for the relation between source node s and destination node d is given by the following:

$$\hat{e} = \text{softmax}\left(\sum_{l=1}^{L} (h_d^{(l)} \cdot V^{(l)})\right) \tag{3}$$

where L is the number of convolutional layers
$V^{(l)}$ is a weight matrix for the destination node representation in the l-th layer

$\hat{e}$ is the predicted relation probability vector.

Prediction:

The node features at various timesteps are passed through an LSTM to further predict the performance of the entities.

The FinBERT [38] embeddings extracted from the FinBERT model are concatenated with the knowledge graph embeddings in different ways. Using the embeddings generated by the RGCN model, three variations are explored: RGCN Type 1, RGCN Type 2, and RGCN Type 3, each employing different embedding configurations.

In RGCN Type 1, the concatenation of FinBERT embeddings, source embeddings, relation embeddings, and destination embeddings obtained from the RGCN results in a combined embedding size of 1168. This configuration integrates both textual and graph-based information, allowing for a more comprehensive representation.

Similarly, RGCN Type 2 follows a similar approach by concatenating the FinBERT embeddings with the RGCN embeddings. However, in this case, the total embedding size is reduced to 1068, providing a more compact representation.

In RGCN Type 3, the embedding representation is simplified through the dot product of the source embedding, relation embedding, and destination embedding from the RGCN. This simplification yields a total embedding size of 868.

By incorporating the FinBERT [38] embeddings with the RGCN embeddings in these different configurations, the aim is to leverage both textual knowledge and relational information encoded in the knowledge graph. This integration enhances the accuracy of financial entity performance predictions.

4.2. Architecture of LSTM Model

Our LSTM model introduces an approach to financial entity performance prediction by intricately combining RGCN embeddings with temporal pattern recognition. The LSTM architecture is specifically designed to process sequences of RGCN embeddings (NEt1, NEt2, and NEt3), corresponding to distinct quarterly financial periods, to capture the temporal dependencies critical for understanding market trends and entity behavior over time. This temporal granularity, aligned with the financial quarters, allows our model to account for seasonal fluctuations and other time-sensitive factors affecting financial performance.

By integrating RGCN embeddings, which encapsulate the relational dynamics within the financial knowledge graph, our LSTM model leverages both the spatial structure and temporal evolution of financial entities. This dual-focus architecture marks a significant advancement over traditional models that either neglect the temporal aspect or fail to incorporate complex inter-entity relationships.

The LSTM (Long Short-Term Memory) model consists of an LSTM layer followed by a fully connected layer for prediction. The LSTM model is designed to process sequences of RGCN embeddings and make predictions based on the learned temporal patterns. The LSTM layer takes input sequences of RGCN embeddings and captures the temporal dependencies within the sequence. The hidden state at the last time step is then passed through the fully connected layer to generate the final predictions.

The LSTM model architecture for predicting the performance of financial entities based on the embeddings from RGCN at timestep 1, timestep 2, and timestep 3 (NEt1, NEt2, and NEt3) are shown in Figure 7. The data are split into three time frames. NEt1 corresponds to tweets from Jan to Apr, NEt2 corresponds to May-Aug, and NEt3 corresponds to Sept-Dec. The update step of the LSTM performs the LSTM Cell Updates. At timestep t, the LSTM cell update formulas are as follows:

$$Inputgate : i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$
$$Forgetgate : f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$
$$Candidatevalue : g_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$
$$Cellstateupdate : c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \quad (7)$$
$$Outputgate : o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$
$$Hiddenstate : h_t = o_t \cdot \tanh(c_t) \quad (9)$$

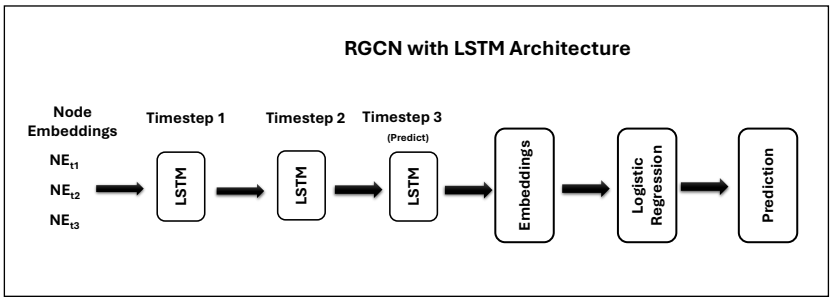

Figure 7. LSTM with RGCN embedding architecture.

In the above formulas, h_{t-1} represents the previous hidden state (output) of the LSTM cell, x_t represents the input at timestep t, which includes the embeddings NEt1, NEt2, and NEt3 for a single financial entity, W_i, W_f, W_c, W_o are the weight matrices for the input, forget, candidate, and output gates, respectively, b_i, b_f, b_c, b_o are the bias vectors for the input, forget, candidate, and output gates, respectively, σ represents the sigmoid activation function and tanh represents the hyperbolic tangent activation function.

The LSTM model architecture includes the LSTM layers and an output layer. The input sequence (embeddings at timestep1) is passed through the LSTM layers, and the final hidden state is fed into the output layer for performance prediction.

5. Experiment Setup

The tweets dataset (https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020?select=Tweet.csv, accessed on 1 September 2024) contains over 3 million unique tweets and includes the following information for each tweet: Tweet ID, Author, Post Date, Text Body, Number of Comments, Likes, Retweets for the year 2015 to 2020. The tweets are related to five major companies: Amazon, Apple, Google, Microsoft, and Tesla, and each tweet is matched with the appropriate share ticker of the respective company. We have used tweets for the year 2016 and 2018 for our experiments. The sample tweets are shown in Table 2.

Table 2. Sample data from the tweets of top companies.

Sample Data (Tweet, Post Date)
This is so exciting. Can't wait for the model III! $tsla @elonmusk just bought over 500k shares too, 1,454,215,066
$TSLA gonna start falling now if people think $AAPL won't buy them anymore due to weak iPhone sales/production cut?, 1,452,015,021

For the Named Entity Recognition (NER) task, the BERT model "dslim/bert-large-NER" is used to identify entities. For the Relation Graph Convolutional Network (RGCN) model, the FintechKG extraction pipeline is employed to generate triplets for the data from tweets in Dataset 3, specifically for the year 2016. These triplets, in the form of (source, relation, destination), are used as input to train the RGCN model for relation prediction.

The RGCN model is trained with hyperparameters, such as a graph-split-size of 0.5, epoch of 30, dropout rate of 0.2, regularization of 1×10^{-2}, and a gradient norm of 1. The learning rate is set to 1×10^{-2}. The RGCN model is trained using data representing three timesteps.

The LSTM model used in the experiment consists of three layers. The first two layers correspond to the RGCN embeddings for timestep 1 and 2, respectively, while the third layer corresponds to the RGCN embedding for timestep 3. The data are split into three time frames. Timestep 1 corresponds to tweets from Jan to Apr, Timestep 2 corresponds to May-Aug, and Timestep 3 corresponds to Sept-Dec.

The LSTM embeddings obtained from the trained model for each entity are then passed to a logistic regression model to make a prediction of the financial performance of the entity. Two types of embeddings: LSTM Type 1 and LSTM Type 2 are used by the regression model as shown in Table 3.

Table 3. Input embeddings.

Input Embedding Type	Description
FinBERT	In this approach, only the FinBERT embeddings were used to train the model.
RGCN Type 1	In this approach, the FinBERT embeddings were concatenated with the (source embedding, relation embedding) and (destination embedding, relation embedding) pairs obtained from the RGCN. The total size of the embedding is 1168.
RGCN Type 2	In this approach, the FinBERT embeddings were concatenated with the source embedding, relation embedding, and destination embeddings from the RGCN. The total size of the embedding is 1068.
RGCN Type 3	In this approach, the dot product of the source embedding, relation embedding, and destination embeddings were used. The total size of the embedding is 868.
LSTM Type 1	In this approach, we use the LSTM embedding representing the source entity of size 204.
LSTM Type 2	In this approach, we use the LSTM embedding representing the source entity of size 204 and destination entity of size 204, with a total size of 408.

Additionally, logistic regression models are trained using both FinBERT embeddings [38] and RGCN embeddings. Four types of inputs are used for the logistic regression model, as indicated in Table 3. The FinBERT embeddings are obtained by passing the tweet as input to the FinBERT model, resulting in embeddings of size 768. The RGCN embeddings are obtained by passing the triplets (s, r, o) as input to train the RGCN model. From the trained model, the entity embedding is retrieved with a size of 100, and the relation embedding is also retrieved with a size of 100.

The hybrid embeddings used by the logistic regression model to perform financial prediction are shown in Table 4. To effectively combine RGCN and FinBERT embeddings for prediction, FintechKG employs a projection layer that maps both embeddings into a shared latent space. This allows the model to leverage the strengths of both information sources for improved performance. The two projection techniques used are the following:

- Linear Projection: A simple linear transformation projects both embeddings into a common space. This is computationally efficient but may not capture complex relationships between the features.
- Non-linear Projection: Techniques like autoencoders or multi-layer perceptrons (MLPs) can learn a more complex non-linear mapping for better representation alignment, potentially improving prediction accuracy.

Table 4. Hybrid embeddings.

Hybrid Embeddings Name	Combination
FinBERT + Graph 1	FinBERT + RGCN Type 1
FinBERT + Graph 2	FinBERT + RGCN Type 2
FinBERT + Graph 3	FinBERT + RGCN Type 3
FinBERT + Temporal 1	FinBERT + LSTM Type 1
FinBERT + Graph 1 + Temporal 1	FinBERT + RGCN Type 1 + LSTM Type 1
FinBERT + Graph 1 + Temporal 2	FinBERT + RGCN Type 1 + LSTM Type 2

The prediction problem involves performing financial performance prediction for companies using data from our dataset (https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020?select=Tweet.csv, accessed on 1 September 2024) from the year 2016 and predicting their financial performance for the subsequent year, 2017 and using data from the year 2018 and predicting their financial performance for the subsequent year, 2019. The companies of interest for the prediction are AAPL (Apple Inc.), GOOG (Google Inc.), GOOGL (Google Inc.), AMZN (Amazon.com), TSLA (Tesla Inc.), and MSFT (Microsoft Corporation).

We are using evaluation metrics such as overall accuracy, refers to the proportion of correct predictions (both true positives and true negatives) among all predictions, F1 Score, the harmonic mean of precision and recall and Macro Avg (Precision, Recall, F1 Score), the average precision, recall, or F1 score across all classes, treating each class equally.

6. Results

The Fintech knowledge graph (FintechKG) statistics for the year 2016, shown in Table 5, present a comprehensive overview of the dataset used for financial analysis. The dataset comprises 892,840 total rows, which indicates the volume of data collected. Within this dataset, there are 22,514 concept entities and 17,682 triplets, highlighting the rich relational information captured. Additionally, the dataset includes 2462 unique entities and 2177 unique relations, demonstrating its complexity and the diversity of financial concepts and relationships it encompasses.

Table 6 details the data used for training the Relational Graph Convolutional Network (RGCN) model. The training dataset consists of 10,682 data points, which provides a robust foundation for model training. To ensure the model's performance is accurately assessed, the dataset also includes 3500 data points each for testing and validation. This division allows for thorough evaluation and fine-tuning of the model, ensuring its accuracy and reliability in predicting financial performance.

The robustness of the proposed approach was evaluated through a series of experiments. These included hyperparameter tuning, where key parameters such as learning rate, batch size, and number of epochs were varied to assess sensitivity. Additionally, graph embedding variations were tested, utilizing different Relational Graph Convolutional Network (RGCN) variants in combination with FinBERT embeddings. The results indicated stable model performance, with a standard deviation of 0.11–0.16%, demonstrating the robustness of the approach across different configurations.

The financial performance prediction results for 2016 and 2018, as shown in Tables 7 and 8, highlight the effectiveness of different hybrid embeddings. In 2016, the FinBERT + Graph 1 achieved the highest accuracy, reaching 94.09% with 40 epochs and 94.62% with 60 epochs. This method also had the best F1 scores for predicting both decreases and increases in financial performance, showing its strong capability in handling financial data. Other methods like the FinBERT [38] and FinBERT + Graph 3 combinations also performed well, with accuracies above 92%, but slightly lower than FinBERT + Graph 1.

Table 5. FintechKG statistics for the year 2016.

Metric	Count
Total Rows	892,840
Total Concept Entities	22,514
Total Triplets	17,682
Unique Entities	2462
Unique Relations	2177

Table 6. Data used for financial performance prediction task.

Data Type	Count
Test Data	3500
Validation Data	3500
Train Data	10,682

Table 7. Financial Performance prediction results for 2016. "Decrease" and "Increase" refer to the scenario where the prediction of the performance of a financial entity decreases or increases, respectively, in the next time period/financial year.

Combination	Epoch	Accuracy	F1 (Dec)	F1 (Inc)	Precision	Recall	F1
FinBERT	40	92.41%	93.64%	90.60%	92.16%	92.08%	92.12%
FinBERT + RGCN Type 1	40	94.09%	95.01%	92.74%	93.81%	93.09%	93.88%
FinBERT	40	94.11%	95.03%	92.77%	93.82%	93.99%	93.90%
FinBERT + RGCN Type 3	40	93.75%	94.76%	92.25%	93.57%	93.44%	93.50%
FinBERT	60	93.10%	94.29%	91.28%	92.72%	92.86%	92.79%
FinBERT + RGCN Type 1	60	94.62%	95.54%	93.21%	94.28%	94.47%	94.37%
FinBERT + RGCN Type 2	60	94.59%	95.53%	93.16%	94.29%	94.40%	94.35%
FinBERT + RGCN Type 3	60	93.15%	94.33%	91.35%	92.76%	92.92%	92.84%

Table 8. Financial performance prediction results for 2018.

	FinBERT	FinBERT + L1	FinBERT + R1 + L1	FinBERT + R1 + L2
Epoch	30	30	30	30
Accuracy	93.21%	94.66%	95.01%	94.05%
F1 (Decrease)	36.21%	46.11%	48.36%	40.68%
F1 (Increase)	96.01%	97.23%	97.38%	97.56%
Precision	95.79%	96.24%	96.20%	95.79%
Recall	93.30%	94.66%	95.70%	94.82%
F1	94.29%	95.32%	95.93%	95.25%

For the 2018 results, the FinBERT + Graph 1 + Temporal 1 showed the highest overall accuracy at 95.01% and had the best F1 scores for predicting decreases (48.36%) and increases (97.38%). This indicates that FinBERT + Graph 1 + Temporal 1 is very good at analyzing financial data and predicting performance changes. The FinBERT + Temporal 1 configuration also demonstrated strong performance, with a 94.66% accuracy and competitive F1 scores.

Overall, these results demonstrate that using graph-aware embeddings like FinBERT + Graph 1 and FinBERT + Graph 1 + Temporal 1 can significantly improve the accuracy and reliability of financial performance predictions. These methods are effective in capturing complex financial trends and relationships, making them valuable tools for financial forecasting. The findings suggest that incorporating relational and contextual information from financial data leads to better prediction outcomes, which is beneficial for making informed financial decisions.

Table 9 presents a comparison of linear and non-linear projection methods for financial performance prediction. The baseline model, without any projection, achieved recall scores

of 0.9151 for the "Increase" class and 0.9459 for the "Decrease" class with an embedding size of 868. Applying non-linear projection at an embedding size of 1024 improved recall to 0.9279 and 0.9475, respectively, and further increasing the embedding size to 2048 resulted in recall scores of 0.9250 and 0.9421. Linear projection methods outperformed non-linear ones, with the best performance observed at an embedding size of 2048, yielding recall scores of 0.9350 for the "Increase" class and 0.9521 for the "Decrease" class. Even with a larger embedding size of 4082, linear projection maintained high recall scores of 0.9260 and 0.9521. These results indicate that linear projection techniques, particularly with an optimal embedding size, significantly enhance the accuracy of financial performance prediction models.

Table 9. Comparison of linear and non-linear projection methods for financial performance prediction.

Method	Recall (Increase)	Recall (Decrease)	Size of Embedding
Baseline [1]	0.9151	0.9459	868
Non-linear Projection	0.9279	0.9475	1024
Non-linear Projection	0.9250	0.9421	2048
Linear Projection	0.9293	0.9509	1024
Linear Projection	0.9350	0.9521	2048
Linear Projection	0.9260	0.9521	4082

[1] Baseline method refers to the original model without any projection.

7. Conclusions

The proposed FintechKG framework systematically extracts and organizes financial knowledge from textual data, incorporating temporal information, taxonomies, and customized extractors to enhance analysis. The extraction pipeline, which includes entity and relation extraction, generates a detailed FintechKG that captures complex financial information. By integrating FintechKG into the Relational Graph Convolutional Network (RGCN) model and Long Short-Term Memory (LSTM) networks, the framework significantly improves financial performance predictions, achieving an accuracy of up to 95% when combining embeddings.

The results of this study demonstrate the effectiveness of the FintechKG framework in understanding financial events, trends, and relationships. The integration of temporal information and contextual knowledge within the framework allows for more accurate and reliable predictions of financial performance. This comprehensive approach addresses key challenges in financial knowledge graph construction and performance prediction, offering a scalable and adaptable solution for various financial applications.

Financial institutions, investors, and analysts can utilize the FintechKG framework to enhance decision-making across several real-world applications:

Improved Forecasting: By integrating diverse data sources like social media and industry knowledge, institutions can make more accurate decisions in asset management, investment strategies, and risk management.
Real-Time Monitoring: Investors can track financial performance in real-time, enabling better portfolio management and quicker responses to market shifts.
Fraud Detection: The scalable FintechKG framework supports fraud detection and compliance by identifying anomalies through additional data sources, ensuring regulatory adherence.
Further research could apply FintechKG to sectors like insurance or real estate and expand its predictive capability by integrating unstructured data such as news and market reports.

Author Contributions: Conceptualization, B.P.J.; Methodology, B.P.J.; Validation, B.P.J.; Formal analysis, B.P.J.; Writing—original draft, B.P.J.; Writing—review & editing, B.T.D. and Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Zhang, Z.; Zeng, X. Risk identification and management through knowledge Association: A financial event evolution knowledge graph approach. *Expert Syst. Appl.* **2024**, *252*, 123999. [CrossRef]
2. Zhu, W.; Chen, Z. An Intelligent Financial Fraud Detection Model Using Knowledge Graph-Integrated Deep Neural Network. *J. Circuits Syst. Comput.* **2024**. [CrossRef]
3. Zuo, R.; Molnár, B. Knowledge Graph Powered LSTM in Stock Investment Decision Making. In Proceedings of the 14th Conference of PhD Students in Computer Science, Szeged, Hungary, 3–5 July 2024; p. 80.
4. Cai, L.; Mao, X.; Zhou, Y.; Long, Z.; Wu, C.; Lan, M. A Survey on Temporal Knowledge Graph: Representation Learning and Applications. *arXiv* **2024**, arXiv:2401.12345.
5. He, Y.; Zhang, P.; Liu, L.; Liang, Q.; Zhang, W. Hip network: Historical information passing network for extrapolation reasoning on temporal knowledge graph. *arXiv* **2024**, arXiv:2401.12346.
6. Wang, J.; Wang, B.; Gao, J.; Pan, S.; Liu, T.; Lehmann, J. MADE: Multicurvature Adaptive Embedding for Temporal Knowledge Graph Completion. *IEEE Trans. Knowl. Data Eng.* 2024, *early access*. [CrossRef] [PubMed]
7. Dong, H.; Wang, P.; Xiao, M.; Ning, Z.; Wang, P.; Zhou, Y. Temporal inductive path neural network for temporal knowledge graph reasoning. *Artif. Intell.* **2024**, *329*, 104085. [CrossRef]
8. Liao, R.; Jia, X.; Li, Y.; Ma, Y.; Tresp, V. GenTKG: Generative Forecasting on Temporal Knowledge Graph with Large Language Models. *Find. Assoc. Comput. Linguist.* **2024**, *2024*, 4303–4317.
9. Luo, R.; Gu, T.; Li, H.; Li, J.; Lin, Z.; Li, J.; Yang, Y. Chain of history: Learning and forecasting with llms for temporal knowledge graph completion. *arXiv* **2024**, arXiv:2401.12347.
10. Ding, Z.; Cai, H.; Wu, J.; Ma, Y.; Liao, R. zrLLM: Zero-Shot Relational Learning on Temporal Knowledge Graphs with Large Language Models. In Proceedings of the Association for Computational Linguistics, Mexico City, Mexico, 16–21 June 2024.
11. Wang, J.; Cui, Z.; Wang, B.; Pan, S.; Gao, J.; Yin, B. IME: Integrating Multi-curvature Shared and Specific Embedding for Temporal Knowledge Graph Completion. *Proc. ACM* **2024**, *2024*, 1954–1962.
12. Gastinger, J.; Meilicke, C.; Errica, F.; Sztyler, T. History repeats itself: A Baseline for Temporal Knowledge Graph Forecasting. *arXiv* **2024**, arXiv:2401.12348.
13. Cheng, D.; Yang, F.; Wang, X.; Zhang, Y.; Zhang, L. Knowledge graph-based event embedding framework for financial quantitative investments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 2221–2230.
14. Zehra, S.; Mohsin, S.F.M.; Wasi, S.; Jami, S.I.; Siddiqui, M.S.; Raazi, M.K.-U.-R. Financial Knowledge Graph Based Financial Report Query System. *IEEE Access* **2021**, *9*, 69766–69782. [CrossRef]
15. Loster, M.; Repke, T.; Krestel, R.; Naumann, F.; Ehmueller, J.; Feldmann, B.; Maspfuhl, O. The challenges of creating, maintaining and exploring graphs of financial entities. In Proceedings of the Fourth International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets, Houston, TX, USA, 15 June 2018; pp. 1–2.
16. Cheng, D.; Yang, F.; Xiang, S.; Liu, J. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognit.* **2022**, *121*, 108218. [CrossRef]
17. Elhammadi, S. Financial Knowledge Graph Construction. Ph.D. Thesis, University of British Columbia, Kelowna, BC, Canada, 2020. Available online: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0392614 (accessed on 23 April 2023).
18. Gao, Y.; Liang, J.; Han, B.; Yakout, M.; Mohamed, A. Building a large-scale, accurate and fresh knowledge graph. *KDD-2018 Tutor.* **2018**, *39*, 1–159.
19. Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; Choi, Y. COMET: Commonsense transformers for automatic knowledge graph construction. *arXiv* **2019**, arXiv:1906.05317.
20. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
21. Rotmensch, M.; Halpern, Y.; Tlimat, A.; Horng, S.; Sontag, D. Learning a health knowledge graph from electronic medical records. *Sci. Rep.* **2017**, *7*, 5994. [CrossRef] [PubMed]
22. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 2901–2908.
23. Wang, W.; Xu, Y.; Du, C.; Chen, Y.; Wang, Y.; Wen, H. Data set and evaluation of automated construction of financial knowledge graph. *Data Intell.* **2021**, *3*, 418–443. [CrossRef]
24. Wu, H.; Chang, Y.; Li, J.; Zhu, X. Financial fraud risk analysis based on audit information knowledge graph. *Procedia Comput. Sci.* **2022**, *199*, 780–787. [CrossRef]
25. Messner, J.; Abboud, R.; Ceylan, I.I. Temporal knowledge graph completion using box embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 1 March 2022; Volume 36, pp. 7779–7787.
26. Xu, C.; Su, F.; Lehmann, J. Time-aware graph neural networks for entity alignment between temporal knowledge graphs. *arXiv* **2022**, arXiv:2203.02150.

27. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the 15th International Conference on the Semantic Web, Heraklion, Crete, Greece, 3–7 June 2018; pp. 593–607.
28. Wang, R.; Li, B.; Hu, S.; Du, W.; Zhang, M. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access* **2019**, *8*, 5212–5224. [CrossRef]
29. Bi, X.; Jiang, Q.; Liu, Z.; Yao, X.; Nie, H.; Yuan, G.Y.; Zhao, X.; Sun, Y. Structure-adaptive graph neural network with temporal representation and residual connections. *World Wide Web* **2023**, *26*, 3389–3408. [CrossRef]
30. Tao, M.; Gao, S.; Mao, D.; Huang, H. Knowledge graph and deep learning combined with a stock price prediction network focusing on related stocks and mutation points. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 4322–4334. [CrossRef]
31. Lam, M. Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decis. Support Syst.* **2004**, *37*, 567–581. [CrossRef]
32. Ma, D.; Yuan, D.; Huang, M.; Dong, L. VGC-GAN: A multi-graph convolution adversarial network for stock price prediction. *Expert Syst. Appl.* **2024**, *236*, 121204. [CrossRef]
33. Elhammadi, S.; Lakshmanan, L.V.S.; Ng, R.; Simpson, M.; Huai, B.; Wang, Z.; Wang, L. A high precision pipeline for financial knowledge graph construction. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 967–977.
34. Zakhidov, G. Economic Indicators: Tools for Analyzing Market Trends and Predicting Future Performance. *Int. Multidiscip. J. Univers. Res.* **2024**, *2*, 23–29.
35. Olorunsola, V.O.; Saydam, M.B.; Arici, H.E. The Predictive Roles of Financial Indicators and Governance Scores on Firms' Emission Performance in the Tourism and Hospitality Industry. *Tour. Manag.* **2024**, *30*, 1382–1403. [CrossRef]
36. Munir, Q.; Akram, B.; Abbas, S.A. Understanding Stock Price Dynamics with Dividend-Related Metrics and Financial Indicators in Pakistan's Non-Financial Sectors. *J. Bus. Econ.* **2024**, *7*, 1–9.
37. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv* **2019**, arXiv:1909.09586.
38. Araci, D. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv* **2019**, arXiv:1908.10063.
39. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.