الجمهورية الجزائرية الديمقراطية الشعبية

Democratic and Popular Algerian Republic

Ministry of Higher Education and Scientific Research

FERHAT ABBAS UNIVERSITY - SETIF1

FACULTY OF SCIENCES

THESIS

Presented for the Department of Computer Science

For the graduation of

PhD

Domain : Mathematics and Computer Science

Field : Computer Science **Option : Artificial Intelligence and its Applications**

By : Djihad Arrar

THEME

Trend Analysis: Link Prediction in Complex Networks and Forecasting of International Trade Flows

Thesis defended in front of the jury members:

- Mansouri Houssem, Professor, University Ferhat Abbas Sétif 1 – President
- Kamel Nadjet, Professor, University Ferhat Abbas Sétif 1 – Supervisor
- Lakhfif Abdelaziz, Professor, University Ferhat Abbas Sétif 1 – Co-Supervisor
- Ben Yahia Sadok, Professor, University of Southern Denmark – Examiner
- Mediani Chahrazed, MCA, University Ferhat Abbas Sétif 1 – Examiner
- Cherroun Hadda, Professor, University of Laghouat – Examiner
- Refoufi Allaoua, Professor (retired), University Ferhat Abbas Sétif 1 – Invited

Abstract

Trend analysis plays a central role in understanding and predicting the evolution of complex systems by extracting meaningful patterns from historical data. This thesis addresses two fundamental predictive tasks that arise in this context: link prediction in complex networks and time series forecasting, with a particular focus on social networks and international trade. These domains are characterized by large-scale, high-dimensional, and dynamically evolving data, which pose significant challenges to traditional heuristic, statistical, and econometric approaches.

For link prediction in social networks, this thesis proposes a novel hybrid framework that integrates similarity measures with supervised machine learning and heuristic-based feature selection. The proposed approach learns predictive patterns directly from similarity-based features while employing a Binary Butterfly Optimization inspired strategy with a mutation-driven mechanism to select informative feature subsets, thereby reducing computational complexity while preserving an effective balance between exploration and exploitation. Model predictions are interpreted using SHAP-based explainability. Extensive experiments on a real-world social network datasets, including Facebook and Email-Eu-Core, demonstrate the effectiveness of the proposed framework, achieving prediction accuracies of up to 98% on Facebook and 85% on Email-Eu-Core.

For time series forecasting in international trade, the thesis develops an interpretable forecasting framework based on the Temporal Fusion Transformer (TFT), capable of modeling long-range temporal dependencies and integrating heterogeneous inputs such as economic and geographic indicators. Using enriched datasets from the UN Comtrade database, the CEPII Gravity dataset, and the World Bank, the TFT-based model achieves a 17% improvement in the coefficient of determination (R^2) compared to baseline models, including Random Forest and Graph Attention Networks. Building on this framework, a hybrid approach termed *TransFusion-LM* is introduced, which integrates large language models through prompt-based strategies without fine-tuning. By embedding TFT outputs and contextual features into structured prompts, TransFusion-LM further improves forecasting performance, with prompt-based GPT-4 variants achieving an R^2 score of up to 94% while also reducing forecasting errors. In addition to improved accuracy, this approach provides intuitive, human-readable explanations of contextual influences on trade forecasts.

Overall, this thesis advances trend-based predictive modeling by developing scalable, accurate, and interpretable frameworks for both network-based and time series-based prediction tasks. By combining optimization-inspired feature selection, transformer architectures and large language models, the proposed methods support more reliable and transparent decision-making in complex social and economic systems.

Resumé

L'analyse des tendances joue un rôle central dans la compréhension et la prédiction de l'évolution des systèmes complexes, en extraisant des motifs significatifs à partir de données historiques. Cette thèse aborde deux tâches prédictives fondamentales qui s'inscrivent dans ce contexte : la prédiction de liens dans les réseaux complexes et la prévision de séries temporelles, avec un accent particulier sur les réseaux sociaux et le commerce international. Ces domaines se caractérisent par des données de grande échelle, à forte dimensionnalité et en constante évolution, ce qui pose des défis majeurs aux approches heuristiques, statistiques et économétriques traditionnelles.

Pour la prédiction de liens dans les réseaux sociaux, cette thèse propose un nouveau cadre hybride qui intègre des mesures de similarité, des techniques d'apprentissage automatique supervisé et une sélection de caractéristiques basée sur des heuristiques. L'approche proposée apprend directement des motifs prédictifs à partir de caractéristiques fondées sur la similarité, tout en s'appuyant sur une stratégie inspirée de l'optimisation Binary Butterfly, enrichie par un mécanisme de mutation permettant de sélectionner des sous-ensembles de caractéristiques informatives. Cette conception permet de réduire la complexité computationnelle tout en maintenant un équilibre efficace entre l'exploration et l'exploitation. Les prédictions du modèle sont interprétées à l'aide de la méthode SHAP afin d'en assurer l'explicabilité. Des expériences approfondies menées sur des jeux de données réels de réseaux sociaux, notamment Facebook et Enron Email-Eu-Core, démontrent l'efficacité du cadre proposé, avec des précisions de prédiction atteignant jusqu'à 98 % sur Facebook et 85 % sur Email-Eu-Core.

En ce qui concerne la prévision des séries temporelles appliquée au commerce international, cette thèse développe un cadre de prévision interprétable basé sur le Temporal Fusion Transformer (TFT), capable de modéliser des dépendances temporelles à long terme et d'intégrer des entrées hétérogènes telles que des indicateurs économiques et géographiques. En utilisant des jeux de données enrichis provenant de la base UN Comtrade, du jeu de données CEPII Gravity et de la Banque mondiale, le modèle basé sur le TFT obtient une amélioration de 17 % du coefficient de détermination (R^2) par rapport aux modèles de référence, notamment Random Forest et Graph Attention Networks. Sur la base de ce cadre, une approche hybride nommée TransFusion-LM est introduite, intégrant des grands modèles de langage à l'aide de stratégies de prompting, sans recourir au fine-tuning. En intégrant les sorties du TFT et des caractéristiques contextuelles dans des prompts structurés, TransFusion-LM améliore davantage les performances de prévision, les variantes basées sur GPT-4 atteignant un score R^2 allant jusqu'à 94 %, tout en réduisant les erreurs de prévision. En outre, cette approche fournit des explications intuitives et compréhensibles pour l'humain sur l'influence des facteurs contextuels sur les prévisions commerciales.

Dans l'ensemble, cette thèse contribue à l'avancement de la modélisation prédictive fondée sur l'analyse des tendances en développant des cadres évolutifs, précis et interprétables pour des tâches de prédiction basées sur les réseaux et les séries temporelles. En combinant une sélection de caractéristiques inspirée de l'optimisation, des architectures de type transformer et des grands modèles de langage, les méthodes proposées favorisent une prise de décision plus fiable et plus transparente dans des systèmes sociaux et économiques complexes.

Dedication And Acknowledgements

I dedicate this work to my beloved family, especially my parents. To my husband, my sisters Khaoula and Malak, my brother Aymen, and to my husband's family, with love and gratitude.

First and foremost, my deepest gratitude goes to Almighty God, who has granted me the strength, health, and patience to persevere throughout this long academic journey. Without His blessings, this achievement would not have been possible.

I would like to express my sincere appreciation to my supervisor, **Pr.Nadjet Kamel**, for her invaluable guidance, continuous support, and patience. Her vast knowledge, encouragement, and positivity have been a source of inspiration that has made this journey both fruitful and motivating. I am equally indebted to my co-supervisor, **Prof. Abdelaziz Lakhfif**, whose insightful advice and unwavering support have been greatly appreciated.

I would also like to extend my sincere thanks to the Estonia team — **Pr. Sadok Ben Yahia, Chahinez Ounoughi, Marek Tiis, and Tarmo Kalvet** for their support, guidance, and collaboration. Their help, both academically and professionally, was invaluable in enabling me to benefit from the Erasmus internship and in enriching the progress of this research.

Special thanks are due to the jury members and to the president of the jury, **Pr. Houssem Mansouri** for accepting the task of evaluating this thesis. Their valuable feedback and thoughtful suggestions will undoubtedly enhance the quality of this work and guide future research directions.

On a personal level, I extend my heartfelt gratitude to my parents, whose sacrifices, love, and prayers have been the foundation of my success. I am also deeply thankful to my husband for their love, understanding, and unwavering support, which have been a true source of strength throughout this journey. To my sisters, Khaoula and Malak, my brother Aymen, and my brother-in-law, thank you for your constant encouragement and faith in me.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

NLP Natural Language Processing

CV Computer Vision

SPA Speech Analysis

LLM Large Language Model

GPT-3 Generative Pre-trained Transformer 3

LLaMA Large Language Model Meta AI

RAG Retrieval-Augmented Generation

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

TFT Temporal Fusion Transformer

TSF Time Series Forecasting

GNN Graph Neural Network

GAT Graph Attention Network

UN United Nations

GDP Gross Domestic Product

GDP per capita Gross Domestic Product per capita

SHAP Shapley Additive Explanations

LIME Local Interpretable Model-Agnostic Explanations

CAM Class Activation Mapping

IR Image Recognition

XAI Explainable Artificial Intelligence

SA Sentiment Analysis

Introduction

Trend analysis is the systematic study of consistent patterns or directions of change in data over time [247]. By transforming historical observations into predictive knowledge, it reduces uncertainty and enables strategic planning in dynamic environments. Its value lies not only in describing the evolution of systems but also in providing a foundation for anticipating future developments. For this reason, trend analysis has become central to disciplines as diverse as climate science [32, 78], where long-term patterns inform adaptation strategies; finance [209, 139], where market movements guide investment and risk management; and social or communication networks [29], where detecting interaction patterns supports tasks such as link prediction. In this thesis, we address two predictive tasks: link prediction in complex networks and time-series forecasting. Link prediction is essential for understanding the evolution of interconnected systems, such as online platforms [114], scientific collaborations, and infrastructure networks [222, 129, 208]. Anticipating future connections enables the development of recommendation systems [112, 224], the detection of emerging communities [143], and the optimization of large-scale network structures. Time series forecasting aims to predict future values of a variable based on its historical observations. It plays a fundamental role in numerous domains, including economics [75, 172], energy [225, 145], and climate science [130, 77]. Time series data often exhibit complex characteristics, including trends, seasonality, volatility, and structural breaks, making accurate forecasting challenging yet essential for planning, risk management, and decision-making in dynamic environments. Within this broad context, the thesis focuses on two representative real-world applications: link prediction in social networks and time-series forecasting in international trade. Modern social platforms generate interaction graphs at scales where accurate link prediction becomes both valuable and challenging. For instance, Facebook reports more than 3.07 billion monthly active users—nearly 40% of the global population — and over 2 billion daily active users, resulting in massive volumes of evolving relationships [57]. At this scale, the graph's sheer size, sparsity, and dynamic evolution make manual analysis or simple descriptive techniques impractical, necessitating algorithmic methods that can automatically extract structural patterns and predict future links in a scalable manner. In such settings, similarity measures (e.g., common-neighbor and Jaccard) remain attractive because they are simple, fast, and often effective at capturing local structural signals; however, when used, they provide only a partial view of link formation and tend to miss heterogeneous and nonlinear mechanisms that govern connections in real networks [4, 92, 17]. Machine learning (ML) offers a complementary advantage by learning decision boundaries from multiple signals and adapting to different network characteristics, but its performance depends critically on the quality and relevance of the input features; in practice, naïvely combining many similarity measures produces high-dimensional feature spaces with redundancy and noise, increasing computational cost and risking degraded generalization, while also reducing interpretability [212]. An open question, therefore, is how to integrate similarity-based structural information with data-driven learning models to balance predictive performance, scalability, and interpretability.

In parallel, time-series forecasting in international trade was selected for its strong economic relevance and inherent modeling challenges. International trade systems consist of complex, multivariate time series that capture trade flows between countries over time and are influenced by economic indicators, geographic factors, and geopolitical events. Global trade in goods and services reached a record level of US 32.2 trillion in 2024, highlighting both its scale and importance for international economic stability [241]. Accurate forecasting of trade flows is essential for governments, policymakers, and businesses to anticipate market fluctuations, assess the impact of external shocks, and manage supply chain risks [239, 80, 213]. Traditional time series forecasting methods, such as ARIMA [33] and Prophet [216], have been widely used due to their simplicity and effectiveness in modeling linear trends and seasonality. However, these approaches are limited in their ability to capture nonlinear dynamics and handle high-dimensional data, restricting their applicability in complex forecasting scenarios. Advances in Deap Learning (DL), particularly LSTM networks [2, 252], have improved forecasting performance by modeling temporal dependencies, yet they remain constrained by sequential processing and limited scalability for long sequences. These limitations raise the question of how forecasting models can effectively capture nonlinear dynamics and long-range temporal dependencies in high-dimensional international trade data while remaining scalable and suitable for real-world decision-making.

To address these challenges, this thesis develops a unified set of data-driven predictive frameworks for trend analysis. For link prediction in social networks, we propose a novel hybrid framework that integrates similarity measures with supervised ML and heuristic-based feature selection. Our approach learns predictive patterns directly from similarity-measure features, employing a Binary Butterfly Optimization-inspired strategy enhanced by a mutation-driven mechanism to select informative feature subsets. This design avoids the use of an explicit, computationally costly fitness function while preserving an effective balance between exploration and exploitation in the feature search space. To enhance transparency, SHAP explainability is incorporated to interpret model predictions, and extensive experiments on real-world social network datasets demonstrate improvements in accuracy, scalability, and interpretability.

In parallel, this thesis advances time-series forecasting in international trade through a progressive modeling strategy that builds on transformer-based forecasting and integrates large language models. The analysis relies on a multivariate trade dataset constructed by integrating multiple authoritative data sources, including the United Nations UN Comtrade database. (2023) [223] , which provides detailed international trade statistics, the CEPII Gravity dataset [40] , which captures geographic and economic relationships between trading partners, and the World Bank Open Data [240], which supplies macroeconomic indicators. Building on this enriched and heterogeneous data representation, an interpretable forecasting framework based on the Temporal Fusion Transformer (TFT) is developed to capture long-range temporal dependencies in multivariate trade flows and to integrate economic and geographic information beyond historical values. While this framework achieves strong predictive performance and offers valuable interpretability, empirical analysis identifies practical challenges related to model complexity, training time, and sensitivity to hyperparameter selection. To address these challenges, the TFT framework is extended by integrating large language models via prompt-based strategies, yielding a hybrid approach referred to as TransFusion-LM. In this unified framework, TFT outputs and structured contextual features are embedded into LLM prompts using advanced prompting techniques such as Few-Shot and Chain-of-Thought reasoning, without fine-tuning. This design enables the joint exploitation of quantitative temporal signals and contextual knowledge, reducing reliance on extensive feature engineering and large training datasets. Experimental results demonstrate that prompt-based LLM configurations—particularly Few-Shot CoT with GPT-4—consistently outperform standalone TFT models, fine-tuned LLMs, and classical ML approaches across multiple evaluation metrics, and provide interpretable, natural-language explanations of forecasting outcomes, thereby enhancing transparency and trust in sensitive domains such as international trade.

Thesis organization

This thesis comprises five chapters, each addressing a different aspect of our research in the two fields of link prediction and time series forecasting.

- **Chapter 1: State of the Art.** This chapter presents a comprehensive literature review on trend analysis, structured around two main domains. The first part focuses on link prediction in complex networks, where we introduce the problem, review commonly used datasets, and discuss the principal similarity measures, ML, and DL employed in this field. The chapter further analyzes the advantages, limitations, and emerging trends of existing approaches, highlighting open research gaps.

  The second part of the chapter is devoted to time-series forecasting. It introduces fundamental time-series models and reviews classical, machine-learning, and DL transformers, as well as advances in LLMs used in this domain. For each category, the chapter discusses the strengths and limitations of the approaches and provides a critical analysis of current trends and challenges in time-series modeling.

- **Chapter 2: Background.**
  This chapter provides the theoretical foundations and methodological background required for the developments presented in this thesis. It describes the core techniques and models employed throughout the work and serves as a conceptual reference for the proposed methods.

  The chapter begins with an overview of fundamental ML and DL models. Since these techniques are well established, they are introduced concisely, focusing on their key principles and roles in predictive modeling. The discussion then proceeds to more advanced architectures central to this thesis, including transformer-based models, with particular emphasis on the Temporal Fusion Transformer (TFT) and large language models (LLMs). These models are presented in greater technical and mathematical detail, highlighting their attention mechanisms and learning structures.

  Overall, this chapter establishes a common theoretical framework and notation, enabling subsequent chapters to focus on methodological contributions and experimental analysis without repeatedly reintroducing foundational concepts.

- **Chapter 3**: **Efficient Link Prediction in Social Networks Using a BBOA-Inspired Heuristic and Similarity-Based Machine Learning.**

This chapter presents the main contribution of the thesis: a novel approach to link prediction that integrates similarity measures with supervised ML techniques. The chapter begins by introducing the datasets and computing similarity-based features. It then describes the proposed feature selection strategy inspired by the Binary Butterfly Optimization algorithm, detailing how this algorithm is adapted and how the fitness function is modified for the link prediction task. Subsequently, the training and evaluation of the predictive models are presented. Finally, analyze and interpret the experimental results to assess the effectiveness and reliability of the proposed approach.

- **Chapter 4**: **High-Accuracy Prediction of International Raw Material Trade Flows Using Temporal Fusion Transformer.**
  This chapter presents the thesis contribution to time-series forecasting of international trade flows. It starts by describing the data acquisition pipeline and the construction of a multivariate time-series dataset that combines bilateral trade flows with economic and geographic covariates from authoritative sources. Next, the chapter formulates the forecasting task, defines the prediction targets and input variables, and explains the preprocessing and feature engineering steps applied to the data. It then introduces the proposed forecasting framework based on the TFT, detailing the model design for multivariate forecasting and the integration of exogenous drivers. Finally, the chapter reports the experimental protocol and evaluation measures, compares the TFT framework against baseline forecasting approaches, and analyzes the results in both accuracy and interpretability.

- **Chapter 5**: **TransFusion-LM: Revolutionizing Trade Flow Forecasting with LLMs and TFT Integration.**
  The motivation behind this chapter is to address the limitations identified in the previous chapter. contribution by enabling LLMs to reason over and enhance transformer-based forecasting outputs. This chapter introduces TransFusion-LM, a novel hybrid framework that integrates the outputs of the TFT with LLMs through prompt-based strategies. The chapter describes how multivariate, tabular trade data and TFT predictions are transformed into structured textual representations suitable for LLM input. It then details the prompting techniques employed, including zero-shot, few-shot, and chain-of-thought configurations and the LLMs considered. The experimental analysis ML baselines. Finally, the chapter interprets the forecasting results and discusses the reliability and trustworthiness of TransFusion-LM in the sensitive context of international trade forecasting.

Finally, we conclude the thesis by consolidating the overall contributions, reflecting on the limitations of the proposed approaches, and identifying potential avenues for future work.

Publication

The publications listed below represent the core research contributions developed throughout this thesis.

1. **A Comprehensive Survey of Link Prediction Methods**, Djihad Arrar et al., The Journal of Supercomputing, February 2024.

2. **Efficient Link Prediction in Social Networks Using a BBOA-Inspired Heuristic and Similarity-Based Machine Learning**, Djihad Arrar et al., 14th International Conference on Research in Computing at Feminine, December 2025.

3. **High-Accuracy Prediction of International Raw Material Trade Flows Using Temporal Fusion Transformer**, Djihad Arrar et al., International Journal of Data Science and Analytics, May 2025.

4. **TransFusion-LM: Revolutionizing Trade Flow Forecasting with LLMs and Temporal Fusion Transformer Integration**, Djihad Arrar et al., Under review in ACM Transactions on Intelligent Systems and Technology.

Role in Research: Challenges, Achievements, and Lessons

Throughout this research, a central role was played at every stage of the project, from conceptualization and methodology design to data analysis, model development, and publication preparation. The work was grounded in an interdisciplinary framework that bridges ML, complex networks, and economic forecasting.

One of the key challenges encountered early in the Ph.D. journey was the lack of suitable time series datasets for link prediction tasks. While the field has been extensively studied in static networks, extending it to temporal settings has been hindered by the scarcity of structured, time-dependent data and the computational demands of DL models, particularly given limited access to GPU servers or high-performance cloud infrastructure. These limitations also restricted experimentation with LLMs, which require substantial hardware resources for training and fine-tuning.

This research journey began with a theoretical study of link prediction, followed by a literature review to identify methodological gaps and propose new research directions for static networks. However, the advancement toward dynamic prediction was hindered by the scarcity of time-series network datasets. A pivotal transition occurred during an Erasmus+ internship at TalTech University in Estonia, during which the research scope was successfully redirected while maintaining the core topics of trend analysis and predictive modeling. With access to substantial computational resources (GPUs) and a large, real-world economic trade dataset, the work evolved to focus on time-series forecasting in international trade, thereby continuing the investigation of dynamic patterns in a different yet methodologically congruent context.

Beyond technical development, this Ph.D. journey taught me critical lessons about scientific research: how to structure and write scientific papers, collaborate effectively within interdisciplinary teams, interpret complex model outputs in consultation with domain experts, and communicate findings clearly. These experiences deepened my understanding of applied ML and prepared me to contribute meaningfully to both academic and practical challenges in data-driven decision-making.

Chapter 1

State of the art

1.1 Introduction

To support the analysis of trends in dynamic systems, this chapter presents a review of existing work in two main areas: link prediction and time series forecasting. These two fields form the methodological basis of this thesis and provide distinct approaches to understanding how systems evolve either through the progression of numerical values or the transformation of relationships within a network.

The goal of this chapter is to understand the current state of the art, highlight the strengths and limitations of the existing approaches, and identify the research gaps that motivate this thesis.

This chapter is organized into two main sections.

Section 1.2 reviews related work on link prediction, including classical and modern approaches. Section 1.3 presents related work on time series forecasting, particularly in the economic domain.

1.2 Related works on link prediction

In this section, we define the link prediction problem and provide an overview of the datasets, network types, and evaluation metrics commonly used in this area. In addition, we review the main categories of link prediction methods, including similarity-based approaches, machine learning models, etc. The section further discusses the strengths and limitations of these methods and analyzes current research trends and open gaps in the literature.

1.2.1 Definition

Link prediction is a task in network analysis that aims at predicting the likelihood of future connections between nodes in a network. It uses the existing network structure and, when available, node attributes to estimate the presence or absence of potential links. This task is often applied to graphs G(V, E), where V denotes the set of nodes and E denotes the set of edges. The network's existing relationships help predict potential connections between unconnected nodes.

Consider the example in Figure 1.1. If node A is linked to both nodes B and D, nodes B and D are not linked, and node A is a common link between B and D, then link prediction would suggest a potential connection between D and B, as well as C and D.

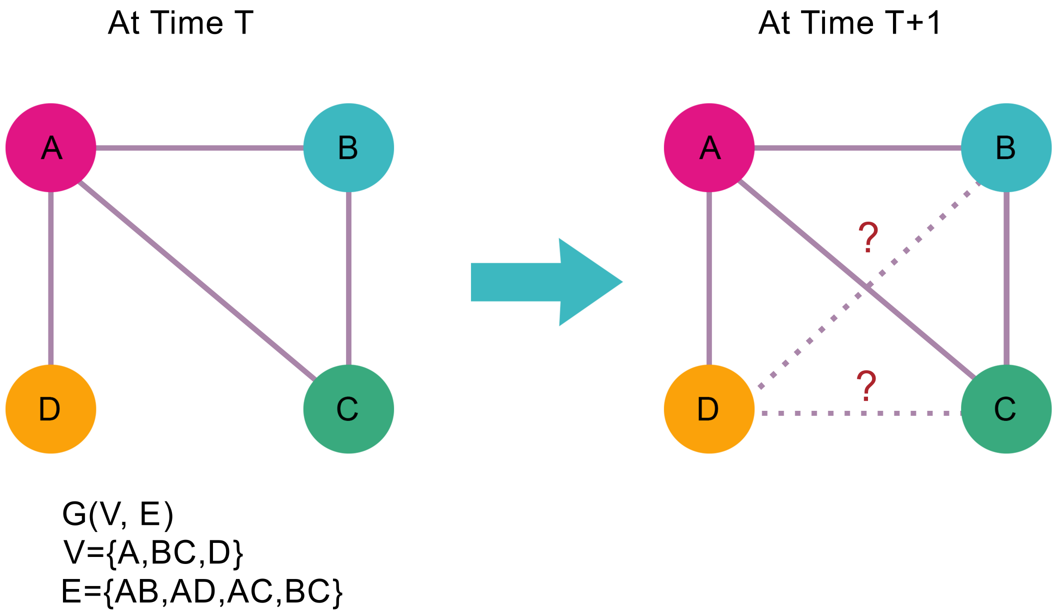

Figure 1.1: Example of Link prediction in Network Graph.

1.2.2 Networks Used in Link Prediction

Several types of networks are used in link prediction. The specific methods and techniques used for link prediction rely on the data type (network) and relationships being evaluated. The most prevalent networks employed in link prediction approaches are enumerated in this section and represented in Figure 1.2.
In Table 1.1, we represent a compilation of different types of networks and the corresponding articles that have used these networks for link prediction.

1.2.2.1 Static Network

A static network does not change over time, with its nodes and edges remaining constant. The paper [233] presents a method for link prediction in static networks based on the popularity of nodes.

1.2.2.2 Dynamic Network

A dynamic network is a network that changes over time, with new nodes and edges forming or disappearing as the network evolves. A dynamic network is used in link prediction to represent relationships between nodes that change over time, such as interactions between people in a social network. The authors employ a dynamic network to identify missing links in [273, 115, 205]. The method used in [273] is the time-varying topological feature learning, which addresses the temporal dependencies among nodes and connections in dynamic networks.

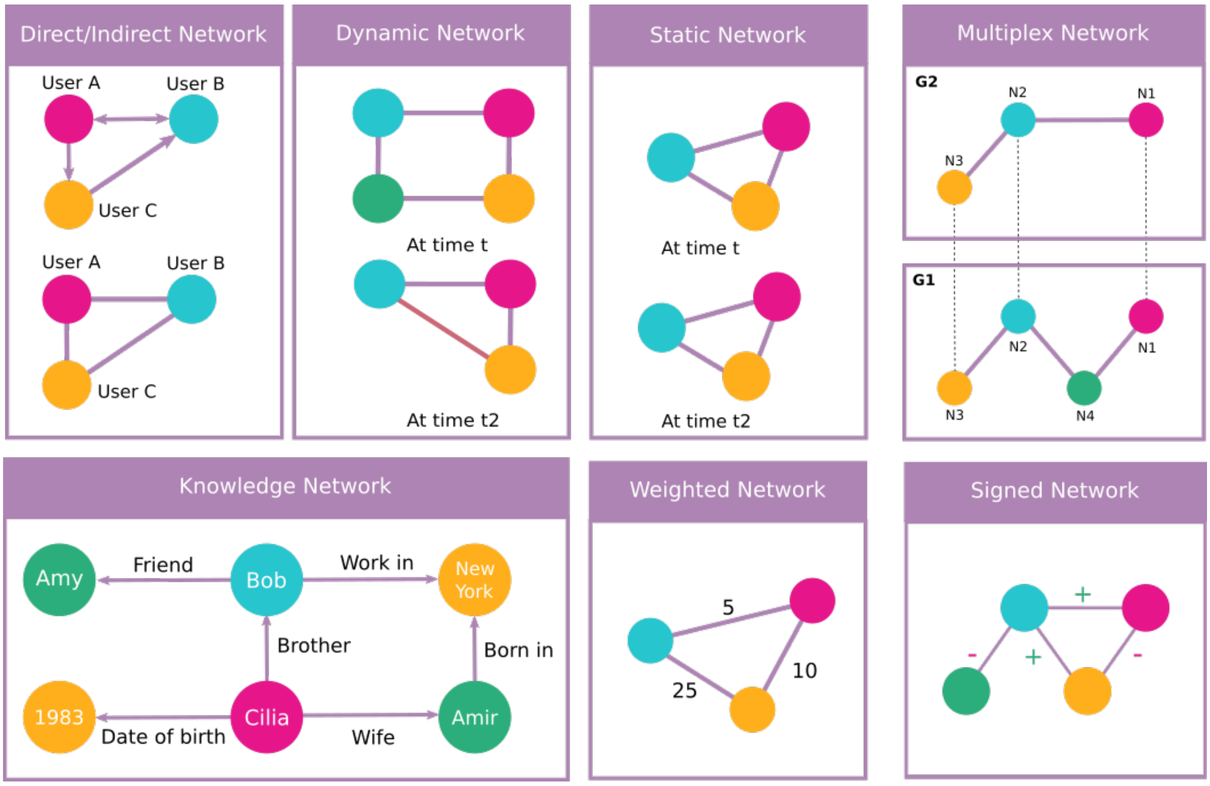

Figure 1.2: Representation of Different Types of Networks

1.2.2.3 Direct and Indirect Networks

A social network can be considered a direct or indirect network A direct network exhibits an asymmetric link structure, as in Twitter. User A can following user B, or B can follow A. The indirect network has symmetric links, as in Facebook. The authors in [36] present an effective method for predicting directional links in directed networks.

1.2.2.4 Weighted Network

A weighted network is a network in which each link between nodes has a weight that represents various aspects of the relationship, such as its strength or importance. To predict links between nodes, link prediction algorithms often use weighted networks to capture their relationships. Recently, the paper [115] introduced a novel temporal link prediction model, GCN-GAN, designed explicitly to predict weighted links in dynamic networks. GCN-GAN is a non-linear model well-suited to handling the complexities of weighted dynamic networks.

1.2.2.5 Multiplex Network

A multiplex network is a network with many layers, where each layer represents a different type of relationship or interaction between nodes. Nodes are connected by multiple types of edges across different layers, capturing various dimensions of their associations. Multiplex networks provide a multidimensional representation of the network by accounting for diverse connections, such as friendships, business partnerships, familial ties, and more. They offer a richer and more comprehensive understanding of complex social interactions in social networks [164] presented an LP framework based on proximity-based characteristics and interlayer similarity (LPIS). Nasiri et al. [166] suggested a novel technique for predicting links in multiplex networks using topologically biased random walks (MLRW).

1.2.2.6 Knowledge Network

A knowledge network is a multirelational graph composed of entities and relations regarded as nodes and different types of edges, respectively [94]. In [185], the authors performed a comparative analysis of knowledge graph embedding methods for link prediction. In [215] tackled the problem of temporal link prediction in a dynamic knowledge graph (KG).

1.2.2.7 Signed Network

A signed network is a network in which edges have positive or negative weights. In a social network analysis, the positive weight of an edge indicates friendship between two nodes, whereas the negative weight of an edge indicates enmity between them. A link prediction method is proposed in [263]. It leverages the structural information of signed social networks. The method compares user similarities and leverages the structural information of the signed social network to predict links.

1.2.2.8 Ego Network

An ego network is a cluster of nodes directly connected to a central node, called the ego. The ego network encompasses the ego and its neighbouring nodes and their connections. The authors in [154] proposed an ELP algorithm that uses an ego network in a social network with relevant features based on the similarity approach.

1.2.2.9 Heterogeneous networks

Heterogeneous networks, also known as heterogeneous information networks or multimodal networks, are characterized by multiple node and edge types. These networks incorporate diverse entities, such as users, items, events, or concepts, with edges representing relationships or interactions, including friendship, co-occurrence, similarity, citation, and other relevant associations.

1.2.2.10 Popular Datasets Used in Link Prediction

Numerous datasets have been developed to support research on link prediction across domains, including social networks, biology, and citation networks. In the following, we overview several widely used datasets and summarize their key characteristics in Table 1.2.

- **The Zachary's Karate Club network** [265] is a social network of friendships among 34 karate club members at a US university.
- **Dolphin social network** [142] is a network of dolphins living in Doubtful Sound in New Zealan.

Table 1.1: Type of the Network with References.

Type of Networks	References
Ego network	[154], [51], [68].
Multiplex network	[166], [199], [100], [256], [155], [141], [258]
Directed network	[74], [6], [144], [67], [60], [45]
Sign network	[133], [1]
Dynamic network	[48], [186],[254], [192], [89], [244], [109], [144], [51], [284], [163], [126]
Knowledge network	[215],[157],[267]
Static network	[110], [163], [180], [6],[214], [281],[167], [148], [232], [264], [7], [23], [21], [96], [51], [231], [30], [119], [206], [269], [268], [110], [56], [103], [37], [31], [65], [47], [270], [127], [44],[54]
Weighted network	[115], [285]
redHeterogeneous network	[65],[276]
Undirect network	[7], [135], [264] [206], [269], [23], [30] [264], [232], [119]

- **The Football network** [107] is a network of American football games played between Division IA colleges during the regular season in the fall of 2000.
- **The Jazz network** [69] is a collaboration network of 115 jazz musicians, where a link between two musicians denotes that they have played in the same band.
- **The USAir network** [27] is an airline network of US airports and their connectivity.
- **The Twitter dataset** [12] is a non-bipartite graph. It was made available as part of the 2020 RecSys Challenge, as described by Belli et al. (2020).
- **Ego-Facebook** [117] is a large dataset. This dataset comprises ”circles” or ”friends lists” extracted from Facebook.
- **Facebook NIPS network** [14] is a social network dataset made publicly available in 2012 by Julian McAuley and Jure Leskovec. This undirected and unweighted network represents user-to-user friendships on Facebook. Each user is represented by a node and an associated link between two nodes representing each friendship.
- **Email-Eu-core network** [260] is an e-mail communication network generated by members of a large European institution. A node represents each user in the network, and an edge is drawn between two nodes if at least one email is exchanged between them.
- **Yeast** [230] is a well-known biological dataset comprising a protein-protein interaction network in the budding yeast Saccharomyces cerevisiae.
- **The Power** [236] is a dataset representing the power transmission network covering the western region of the United States.

- **CollegeMsg** [175] is a dataset containing records of messages exchanged on an online social network at the University of California, Irvine. It includes information about the source user (sender), destination user (receiver), and timestamp of each message. The dataset spans the period from April 2004 to October 2004. It provides insights into the communication patterns and interactions within the online social network during this duration.

- **Twitter-Foursquare** [87] is a dataset that combines data from both the Twitter and Foursquare platforms. It was collected between May and September 2012. It specifically focuses on three major cities in the United States. The dataset comprises tweets and check-ins from users who checked in during that period and shared their check-ins on Twitter.

- **Vickers** Vickers collected [229] Dataset, and it consists of data obtained from 29 seventh-grade students in a school located in Victoria, Australia. The students were asked to nominate their classmates based on several relations, focusing on three specific layers: The relationship layer, the Best friends layer, and the Working preference layer.

1.2.3 Evaluation Metrics

In this section, we present the measures commonly used to evaluate the results of link prediction algorithms. The choice of evaluation measures depends on several factors, such as the method used and the types of datasets. Some measures may be more suitable for certain methods or datasets than others, and the choice of measures can also depend on the research objective. For instance, if the goal is to identify as many true positive links as possible, recall may be more important than precision. Conversely, precision may be more crucial if the focus is on minimizing false positives.

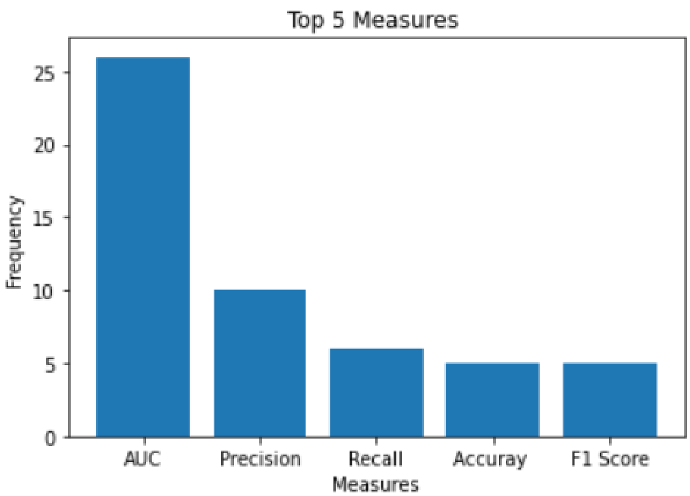

Figure 1.3: The most commonly used measures among the papers included in our survey
.

Table 1.2: Description of Popular Datasets Used in Link Prediction.

Dataset	Nodes	Edges	Dir/Undir	Type of Networks	Paper	Link
Karate	34	78	Undirected	Static Network	[7], [23], [30] [264], [232], [119]	`https://networkrepository.com/soc-karate.php`
Dolphin	62	159	Undirected	Static Network	[180], [7], [23], [21], [30], [232], [96]	`https://networkrepository.com/soc-dolphins.php`
Jazz	198	2742	Undirected	Static	[135], [119], [41]	`http://konect.cc/networks/arenas-jazz/`
Football	242	4100	Directed	Directed Network	[21], [51], [264], [232], [206], [96]	`https://datahub.io/collections/football`
USAir	332	2100	Undirected	Static Network	[7], [135], [264], [206], [269]	`https://networkrepository.com/USAir97.php`
Email-Eu-core	1005	25,571	Directed	Dynamic Network	[110], [109], [7], [23], [264], [119], [144], [45]	`https://snap.stanford.edu/data/email-Eu-core.html`
CollegeMsg	1899	T59835, S20296	Undirected	Dynamic Network	[109], [26]	`https://snap.stanford.edu/data/CollegeMsg.html`
Yeast	2375	11,693	Undirected	Static Network	[23], [30], [135], [264], [232]	`https://humgenomics.biomedcentral.com/articles/10.1186/1479-7364-3-3-291`
Facebook NIPS	2888	2981	Undirected	Ego Network	[110]	`http://konect.cc/networks/ego-facebook/`
The Ego Facebook	4039	88,234	Undirected	Ego Network	[68], [51]	`https://academictorrents.com/details/`
Power	4941	6594	Directed	Static Network	[51], [232], [206], [269], [37], [38], [272], [21], [96]	Power
Twitter	1043	4860	Directed	Static Network	[60], [100], [186]	`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5515441/`
Twitter-Foursquare	10,989	16,104, 23,348, 11,459	Directed	Multiplex network	[100]	Twitter
Vickers	29	240, 126, 152	Undirected	Multiplex	[155], [141]	Figshare Link

- **Area Under the Receiver (AUC)** is the area under the receiver operating characteristic curve (ROC) [79], which is a measure of the model's ability to distinguish between positive and negative links. The formula to calculate the AUC score is shown in Equation (1), where n' is the number of times the missing links have scores greater than the non-existing links, and n'' is the number of times their scores are equal. Assuming the scores are generated from an identical and independent distribution, the AUC is approximately 0.5. A higher AUC value indicates a more efficient algorithm compared to random choice [161].

$$\mathrm{AUC} = \frac{n' + 0.5n''}{n} \tag{1.1}$$

- **Accuracy** is a metric that indicates the percentage of correctly classified links by a link prediction model. It can be computed using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.2}$$

  where **TP (True Positive)** is the number of correctly predicted positive links.
  TN (True Negative) is the number of correctly predicted negative links.
  FP (False Positive) is the number of incorrectly predicted positive links .
  FN (False Negative) is the number of incorrectly predicted negative links.

- **Precision** is a metric used to evaluate the performance of a link prediction model. It measures the proportion of correctly classified positive links (i.e., links that actually exist) among all classified positive links.

$$Precision = \frac{TP}{TP + FP} \tag{1.3}$$

- **Recall** is the ratio of correctly classified positive links to the total number of positive links.

$$Recall = \frac{TP}{TP + FN} \tag{1.4}$$

- **F1 Score** is a measure of the balance between precision and recall.

$$\mathbf{F1} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{1.5}$$

1.2.4 Methods for Link Prediction

We classify link prediction methods based on recent literature (2020-2023). We reviewed approximately 80 papers on similarity heuristics, ML, Dimensionality reduction, and related approaches.

1.2.4.1 Similarity based methods

Similarity is a measure of the degree of connection between two nodes in a network. It is frequently used to predict the likelihood that a link will form between two nodes that are not currently connected in the network. The most common similarity measures

between nodes, used in link prediction studies, are local, quasi-local, and global indices. Local indices, such as Common Neighbors [171], Salton Index [191], Jaccard Index [92], Sorensen Index [210], Hub Promoted Index [123], Preferential Attachment Index [25], Adamic Adar Index [4], and Resource Allocation Index [282], calculate scores based on the neighbourhood information of nodes with a path distance less than two. Global indices, like Katz [102], Random Walk [220], SimRank [93], Leicht Holme Newman Index (LHNI) [116], and Matrix Forest Index (MFI) [43], use information from the entire network to calculate scores with a path distance greater than two. Quasi-local indices, such as Local Random Walk [134] and Local Path Index [283], combine the advantages of local and global indices and compute scores for nodes at most two hops away. These quasi-local approaches tend to have higher prediction accuracy compared to local methods.
In conclusion, these old metrics are adaptations of the common neighbour method that normalize or consider the significance of the neighbouring nodes to reduce biases resulting from the uneven distribution of node degrees. These old measures are still being used in recent studies, often in combination with other methods. They are computed using the equations given in Table 1.3, where Γx and Γy represent the sets of neighbours of nodes x and y, respectively.

Table 1.3: The Most Popular Similarity-Based Metrics.

Measure	Equation	Reference
Common Neighbors	$sCN_{xy} = \|\Gamma(x) \cap \Gamma(y)\|$	[171]
Jaccard Index (sJI)	$sJI(x,y) = \frac{\|\Gamma X \cap \Gamma Y\|}{\|\Gamma X \cup \Gamma Y\|}$	[92]
Salton Index (sSalton)	$sSalton(x,y) = \frac{\|\Gamma X \cap \Gamma Y\|}{\sqrt{\|\Gamma X\| \cdot \|\Gamma Y\|}}$	[190]
Hub Promoted Index (sHPI)	$sHPI(x,y) = \frac{\|\Gamma X \cap \Gamma Y\|}{\min(\|\Gamma X\|, \|\Gamma Y\|)}$	[123]
Hub Depressed Index (sHDI)	$sHDI(x,y) = \frac{\|\Gamma x \cap \Gamma y\|}{\max(\|\Gamma x\|, \|\Gamma y\|)}$	[123]
Leicht-Holme-Newman Index (sLLHN)	$sLLHN(x,y) = \frac{\|\Gamma x \cap \Gamma y\|}{\|\Gamma x\| \cdot \|\Gamma y\|}$	[116]
Adamic-Adar Index (sAA)	$sAA(x,y) = \sum_{w \in \Gamma x \cap \Gamma y} \frac{1}{\log(\Gamma w)}$	[4]
Resource Allocation Index (sRA)	$sRA(x,y) = \sum_{w \in \Gamma x \cap \Gamma y} \frac{1}{\Gamma w}$	[282]
Preferential Attachment (sPA)	$sPA(x,y) = \Gamma x \cdot \Gamma y$	[25]
SimRank (sSimRank)	$sSimRank(x,y) = \frac{C}{\Gamma x \cdot \Gamma y}$	[93]

First, in [180], the authors introduce a new link prediction algorithm, CNDP (Common Neighbors Degree Penalization). The CNDP algorithm combines the network's average clustering coefficient and topological characteristics, such as the number of shared neighbours, to calculate the node similarity score. By including common neighbours in the calculation, this method improves upon the previously proposed adaptive degree penalization (ADP) method. The results demonstrate that CNDP delivers substantial improvements in accuracy and computational efficiency compared to other similar techniques. Additionally, CNDP considers both local and global features of the network, providing a more comprehensive and nuanced examination of the network structure. In the work of Ahamad et al.[7], a novel algorithm called the Common Neighbor and Centrality based Parameterized Algorithm (CCPA) was introduced for complex network link prediction. This algorithm is based on two key node characteristics: the number of common neighbours and the nodes' centrality, calculated using closeness and betweenness centrality. The "common neighbor" feature measures the number of shared neighbors between two nodes in the network. The effectiveness of CCPA was evaluated through experiments on eight real-world datasets. The authors in [23] presented a novel similarity measure based on the local path information of nodes. The similarity score between two nodes is computed from the local information of nodes within a specified distance, accounting for direct connections and all shorter paths between them. This approach combines the strengths of two well-known similarity indices, the Katz Index and the Adamic-Adar Index. The authors in [21] proposed a parameter-free new similarity metric based on degree and path depth. The metric extends their previous work ([96]), which focused on local features such as node degree and on global features, including path structure. The metric considers different-length paths to reduce the impact of high-degree nodes by using path lengths 2 and 3 and accounting for the degrees of the source and destination nodes. The authors applied ML techniques, including k-nearest neighbours, logistic regression, artificial neural networks, decision trees, random forests, and support vector machines, for binary classification. In [231], the authors present a novel multidimensional network model incorporating multiple public opinion factor dimensions. The model is designed to capture the complexity of link formation in ”We the Media” networks, where various factors, including structural information, occupational environments, interests, and social psychology, influence social user relationships. To evaluate the effectiveness of their model, the authors propose a link prediction algorithm tailored explicitly for multidimensional network links. The algorithm is compared against baseline methods, including the Common-Neighbourhood-Driven model, the Jaccard index, and the SimRank method. The empirical analysis was conducted using public opinion data from Weibo.com. In [141], the author explores link prediction in multiplex networks and proposes a multiple-attribute decision-making (MADM) approach to address the problem. By treating potential links in the target layer as alternatives and diverse layers in the network as attributes, the goal is to use information from all layers effectively. The proposed approach ranks alternatives by ideality scores and assigns weights to layers using a defined layer-similarity measure based on cosine similarity. Extensive experiments demonstrate that the proposed method outperforms competing methods in both accuracy and running time. In [284], the author proposed a model for growing networks and introduced novel time-sliced metrics to estimate the likelihood of missing links. These metrics, based on established link prediction indices, achieve superior performance compared to existing approaches, particularly when decay factors are small. Additionally, the paper proposes function expressions to determine the optimal number of slices and decay factor in real-world networks, thereby enhancing the efficiency of link prediction. By leveraging these formulas, the method accurately and efficiently predicts missing links by estimating the aging speed of growing networks. The implementation steps and advantages of the time-sliced metrics are exemplified using the Preferential attachment metric, further highlighting their effectiveness in networks with aging sites.

In the article [154], the authors proposed an ELP algorithm that used an ego network in a social network with relevant features based on the similarity approach. The features considered are Egocommon Neighbors, Ego Resource Allocation, Ego Page Rank, Ego Node2vec, and Ego Clustering Coefficient, focusing on Egocommon Neighbors (ELP). They found that this algorithm is more accurate than state-of-the-art methods for the same network. Similarly, the authors in [135] focus on the initial information contribution of nodes as a critical factor in prediction accuracy. By incorporating topological information and an adjustable parameter, the algorithm quantifies the significance of the node’s initial information. By analysing bidirectional information transmission between nodes, the algorithm measures structural similarity based on the total amount of information received by nodes. Experimental evaluations on real-world networks demonstrate the superiority of the proposed algorithm compared to existing benchmark indices. Next, in [109], the authors presented an enhanced feature set that addresses the link prediction problem in dynamic networks by incorporating individual snapshots and the overall network structure. Using a reward-and-penalty structure, the novel Cost-based feature for link prediction (CFLP) estimates edge behaviour across the entire network. Four categories of similarity indices are used to measure edge activity in individual snapshots. Feature selection is performed on fourteen different snapshot-based features to identify the optimal combination. The combined feature set is evaluated with ML models and outperforms state-of-the-art approaches. Experimental results on real-world datasets demonstrate the superior performance of the proposed method. A novel link prediction approach is introduced in [51]. It addresses spurious links in static networks and predicts the removal of existing links in dynamic networks. The strategy is based on the concept of attraction force between nodes and node-level assignment. For static networks, connection probabilities of existing links are calculated to detect spurious links. In dynamic networks, a virtual network is constructed from network changes, and connection probabilities are computed to predict link removals. The author of [155] presented a link prediction challenge on a multiplex network. The proposed method combines edge and node relevance to predict links between unconnected nodes accurately. It used an aggregation model to summarize information across layers into a weighted static network, accounting for layer density. Node relevance is determined by the node's importance within the overall graph structure, whereas edge relevance considers local information. The method outperforms existing approaches for link prediction in both weighted static networks and multiplex networks. Finally, the authors in [74] proposed a topological nearest-neighbours similarity method for directed networks. The method improves upon the Sorensen index and its variants by accounting for the directionality of network edges. The proposed method effectively combines local and global information of network nodes by deriving the topological nearest-neighbours similarity index based on the Global Local Hybrid Neighbors(GLHN) similarity index. Empirical validation on real directed network datasets demonstrates the superior performance of the proposed method, with lower error, higher accuracy, and greater robustness compared to benchmark indices. Although the proposed method shows promising results, future work can focus on addressing computational complexity challenges, exploring more efficient algorithms such as DL, and further cleaning global node information to improve link prediction accuracy. Yuliansyah et al.[264] address the cold-start problem in link prediction when new users with limited information join a network. They propose a Degree of Gravity for Link Prediction (DGLP) approach, inspired by Newton's law of gravity, that considers the gravity between node pairs and their common neighbours in a single-layer network. Evaluations using multiple datasets and benchmark methods demonstrate that DGLP significantly improves prediction performance, achieving higher AUC values and successfully predicting the cold-start problem in 99.94 % of node pairs. However, the performance of DGLP depends on the presence of cold-start issues in the network. The authors suggest future research directions, including incorporating non-topological information and exploring complementary combinations of node attributes and network-structural information to predict future relationships.

Similarity Approaches using Random Walk

The connections between nodes can be modelled through the random walk [134], where transition probabilities are used to determine the path a random walker would take from

one node to another. Various link prediction metrics use random walks to compute node similarities.
Nasri et al.[166] proposed a new approach for link prediction in multiplex networks using topologically biased random walks (MLRW). They explored these methods to assess the likelihood of a new connection in a target layer by considering inter- and intra-layer relationships within the network. Bias functions were used to compute cross-layer weights between layers, and these methods were evaluated on real-world datasets from various social networks. Next, the author of [30] introduced the Mutual Influence Random Walk (MIRW) algorithm for link prediction. In MIRW, the next node is selected based on its impact on the current node and on the path's effectiveness. The technique was evaluated on 11 real-world networks and compared with existing local, quasi-local, and global indices.

Similarity Approaches using Community Detection
Community detection involves identifying groups of nodes in a network that are more closely connected than to the rest of the network. For example, detecting groups of people with common characteristics such as university, age, country, or interests. This technique helps predict missing links in the network, as a link is likely to exist between two nodes if they belong to the same community. Using community detection as the initial step in link prediction methods can yield more effective link representations than those from similarity-based methods. The author in [232] implements a community information (CI) based local similarity index for link prediction in large-scale networks using Spark GraphX and a parallelization approach. They develop a family of nine CI-based local similarity indices, including Common Neighbors (CN), Salton (Sal), Jaccard (Jac), Sørensen (Sor), Hub Promoted Index (HPI), Hub Depressed Index (HDI), Leicht-Holme-Newman Index (LHN), Adamic-Adar (AA), and Resource Allocation (RA). The authors combine the similarity algorithm with a local index and use the parallel BGLL algorithm to discover community networks in larger-scale networks. The model is evaluated on both small and large datasets. On the other hand, In [119], the authors proposed to solve the link prediction problem with a framework based on community detection and a similarity-based approach.

They proposed community relationship strength (CRS) as a measure of community proximity and included CRS alongside basic similarity indices. They used two traditional algorithms to detect the community: FastQ and Louvain. These two algorithms are based on modularity optimization. They used three traditional local similarity indices: Common Neighbors (CN), Adamic-Adar (AA), and Resource Allocation Index (RA). They evaluated CRS-AA, CRS-CN, and CRS-RA on 12 networks and compared their results with those of other link prediction methods. Singh et al.[206] present a new algorithm, Community Link Prediction via Information Diffusion (CLP-ID), for predicting missing links in networks. The algorithm is based on community detection and employs CD methods to determine the community structure of the networks using information diffusion. They use the IC model to integrate the information diffusion model. Then, they compute the overall index for each existing link and the likelihood score for each non-existing link. Next, the authors proposed in [100] a new approach for link prediction in multiplex networks called Community-Guided Link Prediction based on External Similarity (CLPES). The authors employ a more advanced MOEA/D-TS algorithm for community detection, which determines the network-layer ordering and generates IP using the Clustering Coefficient metric. The proposed CLPES computes external similarity across network layers using a new external similarity metric (ExSim) and predicts internal links by considering various intralayer features, such as the Jaccard coefficient, preferential attachment, and Adamic-Adar while taking into account the community structure of the network. The likelihood of connection development between nodes is determined by integrating these processes.

In many real-world networks, community identification is a useful strategy for link prediction, and combining it with other approaches can enhance performance.

Several studies have demonstrated that similarity-based link prediction approaches can achieve high accuracy. These approaches can capture the global structure of the network, thereby enhancing accuracy. In addition, these similarity approaches are both efficient and low-cost, making them practical and cost-effective solutions for link prediction. However, these approaches have limitations. The high time cost of the algorithms can make them impractical for large or complex networks, while their sensitivity to noise and sparsity can lead to inaccurate predictions. In social network analysis, the similarity measure based on the number of common neighbours is often used as a first-order heuristic function to predict potential friendship relations. It has been shown to achieve satisfactory performance. However, this heuristic may be ineffective for protein-protein interaction networks. In these types of networks, two proteins sharing many common neighbours may still have a low probability of interacting, making this heuristic an unreliable predictor in this context. Also, many similarity indices, such as Common Neighbor (CN) and Resource Allocation (RA) indices, are commonly used in link prediction for static networks. However, these methods have a limited ability to handle high levels of nonlinearity. Despite these limitations, similarity-based approaches remain highly desirable for many network analysis applications due to their high accuracy and efficiency.

1.2.4.2 Machine Learning

ML has been widely used for various tasks, such as text and image classification [106, 197]. It has also been applied to the problem of link prediction in networks. A common approach in ML-based link prediction is to use feature engineering to extract relevant features from network data. These features are then used as input to a learning algorithm to predict the probability that a link forms between two nodes. We classify approaches to link prediction using ML into four categories: supervised, unsupervised, reinforcement learning, and DL. This paper [138] presents NCSM (Node Centrality and Similarity-based Parameterised Model), a novel link prediction method that addresses the limitations of existing models, such as Graph Convolutional Networks (GCNs), particularly in short-path and ego networks. NCSM enhances prediction by integrating node centrality and similarity as edge features within a custom Graph Neural Network (GNN) architecture, making it the first parameterised GNN model to exploit topological information for this task. Evaluated on five benchmark datasets, NCSM consistently outperforms state-of-the-art models, including GCN and Variational Graph Autoencoder, across multiple metrics. Its success is attributed to the innovative combination of centrality, similarity, and efficient topology-aware design.

1. **Supervised Learning**
   Supervised learning is an ML technique for binary classification problems, such as link prediction. In this approach, a graph (E) is transformed into data points (x, y), where each data point represents a pair of nodes. The data are labeled with a

positive class label (+1) if a relationship exists between the nodes and a negative class label (- 1) if no edge (link) exists between them. The labels of x and y are defined as follows:

$$s(x,y) = \begin{cases} (x,y) \in V & \\ +1 & \text{if } (x,y) \in E \\ -1 & \text{if } (x,y) \notin E \end{cases}$$

Figure 1.4 depicts the typical workflow in many studies that use ML for link prediction. First, we collect or select a correctly labelled database and partition it into training and testing datasets. Next, we extract the features from the network to describe each node pair. These features may include measures of similarity or other domain-specific features. To obtain a trained model, a suitable ML algorithm is selected and trained on the labelled training set to learn a function that maps the features of a pair of nodes to the predicted probability of a link between them. Finally, the test dataset is used to evaluate the trained model. Precision, recall, and AUC metrics evaluate the model's performance.

Support Vector Machine Approaches

In [199] also applied SVM, Random Forest, and AdaBoost algorithms to a multiplex network. This research employed hybrid approaches that combined local and global similarity measures to construct feature vectors for node pairs. The feature vector included common neighbourhood, node degree, and clustering coefficient, as well as two new features, ”neighbor friendship (FoN)” and ”friendship in auxiliary layers (FAL),” which capture structural information across all layers of the multiplex network. The feature vector was then fed into a classifier to predict the relationships between pairs of nodes. The effectiveness of this method was evaluated by conducting experiments on six multiplex networks and comparing the results with those of NSILR [258]. This approach demonstrated the benefits of integrating structural information across all layers and avoiding parameter tuning. On the other hand, in [68], a binary classification approach was applied to the ego network in a social network. This approach extracts features such as age and location from user profiles using homophily theory. This approach uses similarity-based methods with various supervised learning algorithms, including Adaptive Boosting (AB), Extremely Randomized Trees, GB, K-KNN, Linear Discriminant Analysis, LR, NB, ANN, RF, SVM, and Extreme Gradient Boosting (XG). The proposed method achieves the highest accuracy with Extreme Gradient Boosting. The advantage of this method is that it uses essential similarity information extracted from user profiles, which cannot be deduced from the graph structure alone. However, this information may not always be available or accurate, potentially affecting prediction accuracy. Kumari [110] also employed ML algorithms such as SVM, KNN, DT, RF, and ANN to predict links in network data by extracting features from the network's structural information. The study found that relying solely on a similarity measure was insufficient to capture link information across all networks. Therefore, network structure and individual features were considered for link prediction. The authors evaluated the models on real-world networks. They found that SVM performed better on the Dolphin, Facebook, and Hamsterster friendship datasets, whereas random forest achieved higher accuracy on the email communication network. This approach's benefits include the use of ML methods to extract characteristics from network-structure information. However, this approach considers only the static, unweighted network for link prediction and does not incorporate content-based features.

Naive Bayes Approach

In [133], the authors proposed the Two Motif Naive Bayes (TMNB) model in the sign network. The TMNB model extends the Single Motif Naive Bayes (SMNB) model by combining information from two motif types and using the Maximal Information Coefficient (MIC) matrix to identify relationships between motifs. The proposed solutions were evaluated and shown to outperform existing methods. The (SMNB) model considers the contribution of each neighbouring node or edge in sign prediction.

XGBoost Classifier

This paper [56] presented the Self-Configured Framework (SCF) integrated with Spark for enhancing link prediction in large-scale social networks. The SCF autonomously configures the best settings based on the dataset size, workload, and cluster specifications. It uses the XGBoost classifier to predict the optimal number of executors per node. The framework demonstrates a 40 % reduction in prediction time and balanced resource consumption, efficiently using limited clusters. The SCF offers an advantage over manual configurations, improving performance and prediction quality without requiring extensive hardware setup.

One advantage of the Supervised learning approach to link prediction in network analysis is its ability to leverage a wide range of node and edge features and representations, leading to more robust and accurate models. Another advantage is the availability of several metrics, including precision, recall, F1-score, and ROC-AUC, for evaluating the performance of supervised learning models.

However, supervised learning for link prediction has some drawbacks. A significant issue is that building models via supervised learning is computationally expensive. Another problem is that social networks grow, and classification models can become outdated, necessitating frequent updates. Additionally, obtaining a large labeled dataset is challenging and time-consuming. Furthermore, the quality and accuracy of link prediction results depend critically on the quality of the features extracted from the network. However, extracting high-quality features from the network can be challenging and may require domain expertise and manual feature engineering.

In conclusion, despite its limitations, supervised learning remains a popular and widely used technology for link prediction, offering precise predictions and strong performance across diverse contexts.

2. **Unsupervised Learning**

   Unlike supervised learning techniques, Unsupervised learning techniques use unlabeled data. Muniz et al. [163] proposed the unsupervised link prediction Contextual-Temporal-Topological (CTT) criterion based on a weighted concept. This weighted concept focuses on three weighting criteria: Temporal-Topological, Contextual-Topological, and Contextual-Temporal-Topological. The CTT criteria merge the three weighting criteria, thereby differentiating them from other approaches that use these criteria without combining them. Ghorbanzadeh et al.[67] proposed a method to solve the problem of two nodes having no common neighbours, meaning

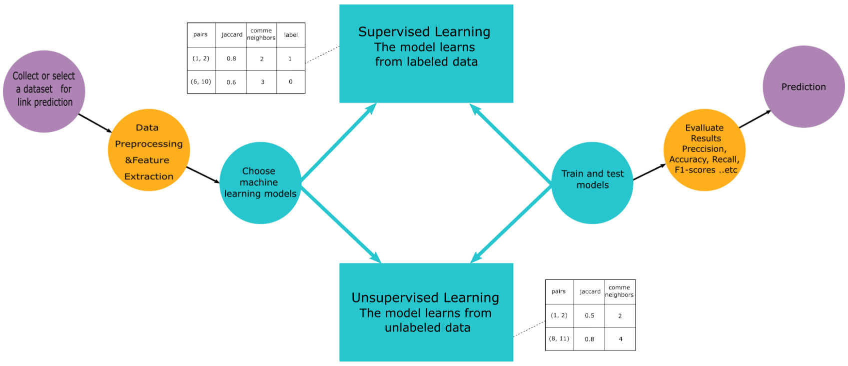

Figure 1.4: Supervised and Unsupervised Learning Workflow Process.

that they are not always predictable to link in the future. They consider neighbourhood direction and the hub and authority of neighbours. The performance of this method was evaluated using both supervised and unsupervised prediction models and compared with several widely used baselines, including Node2Vec, DeepWalk, LINE, and M-NMF.

K-means

Mavromatis et al.[148] applied the popular unsupervised learning algorithm K-means using a graph representation learning method called Graph InfoClust (GIC). The K-means algorithm was used to compute clusters by maximising simultaneous mutual information. The performance of GIC was compared with other unsupervised methods, including Deep Graph Infomax, Variational Graph Auto-Encoders, Adversarially Regularised Graph Autoencoder, DeepWalk, Deep Neural Network for Graph Representation, and Spectral Clustering.

3. **Deep Learning (DL)**

DL has garnered significant attention recently due to its effectiveness in solving a range of problems, including link prediction. The DL approach is an automated learning technique that uses neural networks to extract the most informative features from structural and content information for link prediction. Unlike other supervised learning methods, DL overcomes limitations in feature extraction by performing this task automatically.

Zhang et al.[268] introduced the Weisfeiler-Lehman Neural Machine (WLNM), a DL approach for link prediction. The WLNM method uses a fully connected neural network to identify local enclosing subgraphs around links that are highly correlated with link presence. By using these subgraphs as training data, this technique has achieved impressive results in link prediction, outperforming state-of-the-art methods. Subsequently, they proposed also [269] SEAL, a novel link prediction method. This later method addresses limitations of the previously proposed Weisfeiler-Lehman Neural Machine (WLNM) method [268]. SEAL improves upon WLNM by using a GNN instead of a fully connected neural network to learn graph-structural features from local enclosing subgraphs. This enables SEAL to learn from latent and explicit node attributes in addition to subgraph topologies. To demonstrate the ability to unify various high-order heuristics, SEAL presented a α-decaying theory. The method was tested on eight datasets and compared with different heuristics, latent feature techniques, and network embedding algorithms. Chen et al. [48] introduced an encoder-LSTM-decoder (E-LSTM-D) model for link prediction in dynamic networks. The key innovation of the method is the integration of structural and temporal features within a single framework. The authors employ an encoder-decoder to automatically describe the network and use an LSTM to capture its temporal evolution by stacking and representing a sequence of graphs. The model's effectiveness was evaluated using a newly developed error rate metric. The results demonstrate that the E-LSTM-D model effectively addresses the challenges posed by high dimensionality, nonlinearity, and sparsity in dynamic network link prediction, owing to its encoder-decoder architecture. On the other hand, Rossi [186] proposed using Temporal Graph Networks (TGNs) with DL techniques for dynamic link prediction. To balance parallel processing efficiency with the ability to learn from the sequentiality of the input, the authors propose a novel training method, TGN. TGNs can memorise long-range dependencies among nodes in the graph. The authors comprehensively studied various framework components and evaluated the trade-offs between speed and accuracy. The performance of the proposed models was compared with static and dynamic baseline models. In their study, Snakar et al. [192] presented a new neural network architecture called the Dynamic Self-Attention Network (DySAT). This approach captures a node's temporal dynamics and structural neighbourhoods in a combined self-attentional representation. The authors aimed to learn low-dimensional vectors that describe the structural properties of a node and its surroundings. To evaluate the performance of DySAT, the authors tested it on four real-world datasets. They found that it outperformed existing state-of-the-art static and dynamic graph embedding baselines. Keikha et al. [103] proposed the DeepLink framework for link prediction in social networks. The framework utilises DL to extract target features from both the content and the structure information of nodes. First, a Word2Vec framework is used to learn structural feature vectors for nodes, and then a Doc2Vec algorithm is applied to learn content feature vectors for each node. Finally, the weight vectors of structure and content information are aggregated into a single vector. The effectiveness of DeepLink was evaluated on the Telegram and irBlogs networks and compared with other link prediction methods.

In [38], the authors address the limitation of information loss in layers of graph pooling in graph neural networks. They propose a solution that learns the features of the target link directly, rather than extracting features from the entire enclosing subgraph. To facilitate this, the original graph is transformed into a line graph, enabling efficient feature learning. The proposed model is evaluated against the baseline techniques on 14 datasets. Zulaika et al. [285] proposed a Link Weight Prediction Weisfeiler-Lehman (LWP-WL) method. Inspired by the Weisfeiler-Lehman Neural Machine, LWP-WL extracts an enclosing subgraph around the target link and applies a graph labelling algorithm to create an ordered subgraph adjacency matrix. Then, this matrix is fed into a neural network, which includes a Convolutional Neural Network (CNN) layer with specialised filters designed for the input graph representation. Extensive evaluations demonstrate that LWP-WL outperforms state-of-the-art methods across a range of weighted graphs. Additionally, an ablation study is conducted to showcase the performance improvement achieved by incorporating different features into the approach.

DL in link prediction presents several key advantages. Firstly, DL models excel at capturing intricate patterns in network data by automatically extracting meaningful features, eliminating the need for manual feature engineering and reducing analysis effort. They demonstrate remarkable scalability, capable of handling large-scale networks with millions or billions of nodes and edges, enabling the analysis of massive datasets. Moreover, DL models effectively capture non-linear relationships between nodes, enabling accurate predictions within complex network dynamics. Additionally, techniques such as Long Short-Term Memory (LSTM) enable the modelling of the temporal evolution of networks by stacking and representing sequences of graphs. This enables accurate link prediction by accounting for the evolving relationships among nodes over time.
However, it is important to consider certain drawbacks. DL models typically require substantial labeled data for effective training, which can be challenging to obtain. Training deep neural networks can be computationally intensive, requiring significant computational resources. The interpretability of DL models is limited, posing challenges in understanding the rationale behind their predictions. Overfitting is a potential concern, particularly when working with limited or imbalanced training data. The effectiveness of DL models relies heavily on diverse and representative training data, as biased or limited data can lead to poor generalization.

In summary, DL in link prediction offers advantages in capturing complex patterns and scalability. However, it is crucial to address challenges such as data requirements, computational complexity, interpretability, overfitting, and the need for diverse training data to enable successful application to link prediction tasks.

Graph Neural Network (GNN)
A Graph Neural Network is a ML model that applies an optimized transformation to all graph attributes, including nodes, edges, and global-context information. This transformation is designed to preserve graph symmetries, such as permutation invariance, which are important for accurately representing and analyzing graph data.
In the field of DL, GNNs are a type of method specifically designed to operate on graph structured data. GNNs aim to effectively integrate feature information and graph structure to learn improved node representations through feature propagation and aggregation. GNNs aim to learn a low-dimensional vector representation for each node in the graph. **"The goal of GNNs is to iteratively update the node representations by aggregating the representations of node neighbours and their own representation in the previous iteration."** from the book Graph Neural Networks: Foundations, Frontiers, and Applications[245].

Graph Neural Networks (GNNs) can be viewed as methods for learning node embeddings by iteratively aggregating information from a node's local neighbourhood. In each iteration, the first information is learned about the direct neighbours, followed by the neighbours of the neighbours, and so on. There are various types of GNNs, each with its own unique update and aggregation functions. The three most common tasks GNNs perform are downstream graph analysis and prediction at the node, edge, and graph levels. A simple representation of the GNN process is depicted in Figure 1.5.

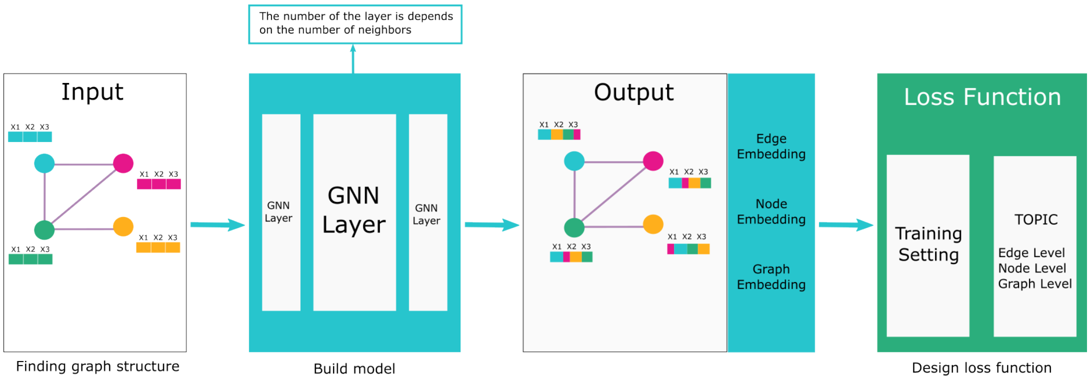

Figure 1.5: The Graph Neural Network Workflow Process

Several techniques for neighbour embeddings in GNNs have been proposed. Kipf and Welling present the first notable work in this area in [105], where they proposed aggregating neighbour information as a normalized sum of states, and incorporated the update operation into this aggregation by adding a self-loop for specific nodes. Another popular technique is using a Multi-layer Perceptron (MLP) for aggregation [266]. This involves using a feed-forward network to perform the aggregation operation, with weights optimized to achieve the best aggregation of neighbouring states. Gated Graph Neural Networks [122] use an attention mechanism in the aggregation process, considering the importance of the neighbouring node's features. This results in an updated embedding that contains more information about the neighbour's features. These networks use a recurrent unit to update the state iteratively over time.

In a new study presented in [37], a novel node aggregation method was proposed to transform the enclosing subgraph into different scales while preserving information. The multi-scale approach used subgraphs at different scales to extract graph-structural features and was evaluated on 14 datasets across various fields. In the field of heterogeneous graph analysis, a new model, the Metapath Aggregated Graph Neural Network (MAGNN), was proposed in [65]. This model addresses three limitations of previous models: omitting node content features, discarding intermediate nodes along the metapath, and considering only a single metapath. MAGNN combines structural and semantic information from neighbouring nodes and metapath context to generate final node embeddings using an attention mechanism for intra-metapath aggregation. The effectiveness of MAGNN was evaluated on three real-world heterogeneous graph datasets for tasks such as node clustering, link prediction, and node classification. In [47], the authors presented a new approach to graph analysis by representing the problem as a graph convolutional network. They proposed a novel multi-level graph convolutional network (MGCN) to uncover the embeddings of each network. The authors introduced a new paradigm that integrates multi-level graph convolutions into both the local network and hypergraph structures. They presented several solutions, including network splitting and space reconciliation, to manage the dispersed training process and to make their proposed framework scalable for large-scale social networks. The paper [81] introduced CPAGCN (Community Preserving Adaptive Graph Convolutional Networks), a novel method for link prediction in attributed networks. CPAGCN effectively integrates attribute and link information by using the AGCN (Adaptive Graph Convolutional Networks) algorithm for network embedding. It incorporates a community detection model to preserve the community structure within the node representations. The link prediction module employs a multi-layer perceptron (MLP) to learn prediction scores for potential links directly. Experimental results on real-world attributed networks demonstrate that CPAGCN outperforms state-of-the-art methods in link prediction. In [254], the authors address the challenge of link prediction in complex, dynamic networks by proposing a novel approach, STEM-GCN. STEM-GCN is a gated graph convolutional network incorporating spatiotemporal semi-variogram analysis to capture spatial and temporal correlations. It introduces a correlation-smoothing strategy to improve prediction accuracy and reduce noise. STEM-GCN effectively captures network dynamics across consecutive time steps using stacked memory-cell structures. Experimental results demonstrate that STEM-GCN outperforms existing methods, showcasing its potential to uncover evolving mechanisms of real-world dynamic networks. In [267], the authors proposed ComplexGCN, a novel extension of graph convolutional networks (GCNs) in complex space for knowledge graph embedding. ComplexGCN utilizes complex-valued embeddings and paratuck2 decomposition-based scoring function to enhance representation quality and predict missing links in knowledge graphs. The model outperforms existing methods on standard link prediction datasets. It introduces complex graph convolutional layers and residual connections to preserve initial embedding information. The paper referenced by [150] introduces DATGN, a novel model for link prediction in dynamic graphs. DATGN addresses the limitations of existing methods by accounting for nodes' local time-space environments. It utilises an activity-based sampling algorithm and a global attention network to aggregate information across the entire dataset. Additionally, a local attention network aggregates information from sampled sequences. Experimental results show that DATGN outperforms state-of-the-art models in both accuracy and efficiency, particularly on the inductive link prediction task. The local spatial-temporal network layer captures evolutionary patterns, improving link prediction accuracy. The study presented in [157] introduces LCILP, a novel strategy for inductive link prediction in Knowledge Graphs. LCILP used a Personalized PageRank (PPR)-based local clustering technique to sample tightly related subgraphs around target links, thereby improving the capture of meaningful local context. The approach employs a GNN-based model for reasoning over the extracted subgraphs. Experimental results demonstrate the superior performance of LCILP compared to state-of-the-art models on three benchmark datasets. The study also explores the relationship between graph properties, such as average clustering coefficient and average node degree, and the effectiveness of link prediction. Huang et al.[89] proposed a novel approach called Temporal Group-Aware Graph Diffusion Network (TGGDN) for dynamic link prediction. The TGGDN incorporates a group affinity matrix to capture mutual interactions between neighbours and integrates it into the graph diffusion process to simultaneously capture group and long-distance interactions. Additionally, a transformer block with self-attention is employed to model the temporal information and enhance interpretability. Experimental results on real-world datasets of varying sizes demonstrate that TGGDN outperforms state-of-the-art methods, achieving significant improvements in terms of accuracy (ACC) and area under the curve (AUC). The proposed method shows promise in dynamic link prediction tasks, and future work aims to address scalability for large-scale dynamic networks. In [270] proposed a novel approach, IEA-GNN (Anchor-aware Graph Neural Network fused with Information Entropy), to address the limitations of existing methods in capturing global location information and distinguishing located nodes in graphs symmetrically. IEA-GNN computes node information entropy and constructs candidate anchor-point sets to capture relative inter-node distances. It uses a nonlinear distance-weighted aggregation learning strategy based on these anchor points to enhance node feature information and improve discrimination between homogeneous neighbourhood nodes. The model avoids aggregating anchor points and highlights positional differences by selecting anchor points based on information entropy. Experimental results on multiple datasets demonstrate that IEA-GNN outperforms baseline models in node classification and link prediction tasks. However, the model's performance may be affected by over-aggregating nodes, an issue that could be addressed in future research to improve generalization. In [276] introduce an end-to-end link prediction method for heterogeneous networks, leveraging metapath projection and semantic graph aggregation. This approach learns node pair embeddings from different metapaths, projecting the network into multiple semantic graphs and employing a graph neural network. A semantic aggregation module combines node pair embeddings using an attention mechanism. Experimental results demonstrate the method's superior accuracy compared to competing approaches.

Graph Neural Networks (GNNs) offer significant advantages for link prediction by capturing and leveraging the underlying structural information of a graph. GNNs excel at extracting complex features from graph structures, thereby enabling accurate predictions. For example, Cai et al. [37] introduced a multi-scale approach that leverages subgraphs at different scales to extract graph-structural features, thereby improving link prediction performance.

Another advantage of GNNs in link prediction is their ability to handle sparse graphs effectively. Traditional methods often struggle with sparse graphs because they rely on explicit features for each node and edge. However, GNNs can utilize the graph structure itself as an implicit feature, making them particularly effective in sparse graph scenarios. This advantage has been demonstrated in various studies, including that of Huang et al.[89], in which the Temporal Group-Aware Graph Diffusion Network (TGGDN) successfully addressed dynamic link prediction tasks on real-world datasets.

Despite these advantages, GNNs also have limitations in link prediction. Scalability to massive graphs is one such limitation, as the computational and memory requirements of GNNs can become prohibitive. This challenge has been addressed in studies such as [47], which proposed a multi-level graph convolutional network (MGCN) to manage the dispersed training process and enhance scalability for large-scale social networks.

Another limitation is the risk of overfitting to small, noisy datasets, which can negatively affect link prediction performance. To mitigate this, techniques that incorporate temporal information have been introduced. For instance, STEM-GCN, proposed by the authors in [254], integrates spatiotemporal semi-variogram analysis to capture spatial and temporal correlations, thereby improving prediction accuracy and reducing noise in dynamic networks.

In summary, GNNs offer advantages in capturing structural information and handling sparse graphs in link prediction. However, scalability to large graphs and the risk of overfitting should be carefully considered. Incorporating temporal information can further enhance link prediction accuracy in dynamic networks.

4. **Reinforcement Learning (RL)**
RL is a method used in ML where programs, called agents, interact with both familiar and unfamiliar environments while continually adapt and learn depending on the grants received as points (positive or negative) for their job performance. These points can be positive or negative, and they are, respectively, labeled as rewards or penalties [156]. In link prediction, an agent can be trained to explore a network, collect information about nodes and edges, and then use this data to predict which links are likely to exist. The agent receives rewards when it correctly anticipates links, and penalties when it makes errors. The problem with the use of ML techniques such as support vector machines and Naive Bayes classifiers depend on the availability of large datasets for training. For this reason,Lim et al. [127] proposed a deep reinforcement learning (DRL)-based criminal network link prediction model and compared it with a gradient boosting machine (GBM) in ML, using a relatively small dataset. The DRL model transformed a graph dataset into a feature matrix for each pair of nodes and used Jaccard, Adamic, and Adar as the layer inputs, with the SNA metrics of the hub index and preferential attachment index serving as weights for the hidden layer. The results showed that the DRL model achieved higher AUC scores (0.85, 0.82, and 0.76) than the GBM models of JUANES, MAMBO, and JAKE. Later, the same authors [126] applied a reinforcement learning algorithm to a temporal dataset for a criminal network and found that the time-based link prediction model (TDRL) has higher prediction accuracy than the previous DRL model. In another study [125], two methods were compared for link prediction on a small dataset generated via self-simulation. One method used a model without a meta dataset (CNA-DRL), while the other used a meta dataset (MCNA-DRL) to improve performance. The results showed that the MCNA-DRL model achieved an AUC score of 79 %, while the CNA-DRL model achieved a score of 70%. Tao et al. [215] also tackled the problem of temporal link prediction in a dynamic knowledge graph (KG) using reinforcement learning models and presented a novel policy network that could learn predictable evolutionary patterns. An update system was also proposed to enable the agent to adapt to changes in the KG without retraining. Next, the authors of [49] proposed an approach to link prediction, RLPath, that combines representation learning and reinforcement-learning-based attentive relation paths. The approach learns relation routes via reinforcement learning using a Markov Decision Process (MDP) model with two agents: a relation agent that chooses relations and an entity agent that chooses entities, both represented by policy networks. The comparison results with the most advanced link prediction techniques demonstrate the competitive performance of RLPath.

1.2.4.3 Dimensionality Reduction

In the context of link prediction, graph dimensionality reduction is an important task. High-dimensional data can be challenging to process and analyze, especially large-scale datasets. By reducing the dimensions, the data can be simplified, making it easier to work with. This can lead to faster and more efficient processing and improved model accuracy. This section summarizes the three main approaches for reducing the dimensions of graphs. The first approach, known as Embedding Methods, involves mapping the graph's information to a lower dimensional space while preserving its structural information and components. The second approach is matrix factorization, where the graph is represented as a matrix and then factorized to obtain the node embedding vectors. The third approach, graph neural networks, also focuses on dimensionality reduction. It was already presented in the ML section 1.2.4.2, allowing for a comprehensive explanation of all ML techniques.

1. **Embedding based Methods**
   The network embedding approach maps network nodes to low-dimensional vectors, preserving their neighbourhood structures and enabling the learning of informative features from the graph. To be more precise, the primary objective of node embedding techniques is to transform the original high-dimensional node representations into lower dimensions, so that nodes that exhibit similar features in the original network are mapped to proximity in the embedded space. In [60], a framework for joint link prediction and network alignment is presented to improve the recall for both tasks. A cross-graph embedding technique based on structural and topological neighbours was developed to effectively embed nodes from separate graphs. This approach is based on random walks, and a new formula is proposed to support network alignment. The experimental results show that the proposed approach outperforms state-of-the-art methods in both link prediction accuracy and network alignment quality. One advantage of this work is its applicability to network alignment in scenarios where the degree scatter plot is narrow or attribute information is unavailable. The study presented in [246] proposes a novel link prediction method for social networks. It addresses the complexity of network features. The technique leverages network embedding to represent the network structure as low-dimensional vectors, capturing spatial relationships and user relevance. Additionally, word-embedding models are employed to convert user text into vectors, accounting for the diversity and complexity of text semantics. To account for the dynamic nature of user behaviour, a time-decay function is applied to quantify the impact of user text on link establishment. To simplify the complexity, the top-k relevant users are selected for each user. Moreover, an attention mechanism is incorporated into a convolutional neural network to better capture user interests in text data. By integrating and analyzing structural and text features, the proposed method achieves the objective of accurately predicting links in social networks. In two articles, the DeepWalk method is adapted for link prediction. The first article [167] focuses on protein-protein interactions, while the second [31] focuses on social networks. Nasiri presented a novel embedding-based approach to link prediction in protein-protein interactions. The author proposes using the DeepWalk algorithm, a graph embedding method that leverages random walks as a similarity measure, to address nonlinearity. The DeepWalk algorithm is modified using a feature selection-based approach to generate graph embeddings. The results are

compared with those obtained using other embedding methods, including node2vec, DeepWalk, Line, and GraphSAGE. The authors of [1] proposed a model for signed graphs in which positive and negative links provide insights into user associations. The authors address the challenges posed by imbalanced class distributions and hidden community structures, which are often overlooked in existing methods. They propose an ensemble framework, ϵNeLp, comprising a network-embedding phase and a classifier-prediction phase, to leverage hidden network communities for predicting negative links. The framework incorporates techniques such as community generation, optimization, probabilistic network embedding, and classifier prediction. Extensive experimental evaluation demonstrates the promising performance of ϵNeLp in terms of pertinency, robustness, and scalability. Barracchia et al.[26] proposed the LP-ROBIN method, which used incremental embedding based on a random walk to capture network dynamism and predict new links. LP-ROBIN can handle the addition of new nodes over time without prior knowledge. When new data arrive, LP-ROBIN updates the model by learning embeddings for new nodes and links, and by updating the latent representations of existing ones. Experimental results demonstrate that LP-ROBIN achieves superior performance in terms of AUC and F1-score, outperforming baselines, static node embedding methods, and state-of-the-art dynamic node embedding methods. The paper cited by [281] introduces NNWLP, a network-embedding-based method that leverages natural nearest neighbours to improve link prediction accuracy. Unlike existing methods that select neighbour nodes with equal probability, NNWLP leverages the network features to find the nearest neighbours. The clustering coefficients are used to assess the contributions of nearest-neighbour and direct-neighbour nodes, yielding node sequences and forming node vectors. These node vectors are then converted into edge vectors and used for link prediction. Experimental results demonstrate that NNWLP effectively utilizes neighbour information and significantly enhances link prediction accuracy. In the paper referenced by [41], the authors introduce LRNP (Low-Rank Network Projection), a novel link prediction algorithm. LRNP is designed by leveraging optimal interactive coefficients derived from solving the objective function, and it uses the adjacency matrix of a fully connected network as the base matrix to capture local structures in observed networks. Experimental findings demonstrate that LRNP surpasses existing state-of-the-art methods regarding link prediction accuracy. In [113], the authors introduce Rotate4D, a novel model for knowledge graph embedding that performs 4D rotations in quaternion space using a special orthogonal group. By embedding entities in quaternion space and applying rotations, the model improves link prediction performance on standard datasets compared to existing models. The Rotate4D model utilizes group theory and quaternion scaling to represent rotations and handle hierarchical relations efficiently. Experimental results demonstrate significant improvements across various evaluation metrics, including MRR and Hits@k. The paper suggests further exploring group-like structures, and their combination with neural networks can enhance link prediction in quaternion and octonion spaces. With a line graph, In [272], they proposed the Line Graph Contrastive Learning (LGCL) method for link prediction tasks. LGCL addresses information loss and limited generalization in similarity-based approaches by leveraging h-hop subgraph sampling and line graph transformation. The link prediction task is transformed into a node classification task using graph convolutional networks, thereby enhancing edge-embedding learning. A cross-scale contrastive learning framework is introduced to maximize the mutual information between line graphs and subgraphs, integrating structural and feature information. Experimental results demonstrate that LGCL outperforms existing methods, offering better generalization and robustness.

The random walk-based embedding approach for missing link prediction offers several advantages, including simplicity and computational efficiency, making it a suitable method for large-scale networks. The approach has been shown to perform well on homophilic networks, where nodes are linked to similar ones, and to capture non-linear relationships. However, a challenge for this approach is its accuracy in predicting links for nodes with high centrality. These nodes tend to have complex connectivity patterns and numerous connections to other nodes.

2. **Factorization-Based Methods**
Factorization-based methods are a commonly used set of techniques for link prediction. These methods use matrix factorization, a technique that partitions a matrix into smaller matrices, to capture the latent representations of nodes in a graph. With these representations, predictions of missing links can be made efficiently. Chen et al.,[45] proposed two approaches for link prediction, NMF-AP [45] and MS-RNMF [46]. NMF-AP combines the information from both local and global network structures by using PageRank to determine the impact score of nodes, which reflects the whole network structure, and the asymmetric link clustering coefficient approach to obtain local information. The performance of NMF-AP was evaluated on ten networks and compared with other methods. NMF models with a single layer map the original network and its corresponding low-dimensional latent space. However, the technique is limited in its ability to uncover hidden multi-layer information in complex networks, such as biological, social, and transportation networks, which contain hierarchical structures with implicit lower-level features. On the other hand, MS-RNMF uses the heat kernel method to measure local similarity and the k-medoids algorithm to capture global information about the network structure. The non-negative matrix factorization model in MS-RNMF is regularized using manifold regularization and sparse learning techniques, which reduces random noise and spurious links. This model was evaluated on seven networks using various metrics and parameters, and the results showed superior performance compared with other methods, particularly for weak, sparse networks and in the presence of random noise. Additionally, In[44] , they introduced the Fusing Structure and Sparsity-constrained via Deep Non-negative Matrix Factorization (FSSDNMF) approach to address this issue. To extract topological details of each hidden layer, the common-neighbour approach was employed to compute similarity scores and to translate them into a multi-layer low-dimensional latent space. The authors eliminated random noise by jointly using the l2,1-norm restricted factor matrix at each hidden layer. In this article [214], the authors address the challenge of cold-start link prediction, where the network structure contains isolated nodes. They propose a multi-nonnegative matrix factorization model that integrates three types of information: community membership, attribute similarity, and first-order structure characteristics. The proposed model successfully predicts missing edges in the disconnected network structure by leveraging these multiple perspectives. This article [144] also presents a novel approach called GNMFCA (Graph Regu-

larized Nonnegative Matrix Factorization Algorithm) for temporal link prediction in directed temporal networks. The proposed algorithm incorporates both local and global information from temporal networks by using PageRank centrality and the asymmetric link clustering coefficient. Graph regularization is employed to capture local information within each network slice, whereas PageRank centrality measures node importance, capturing global information. By jointly optimizing these factors in a nonnegative matrix factorization model, the GNMFCA model simultaneously preserves local and global information. The paper also introduces effective multiplicative updating rules for solving the model and provides a convergence analysis of the iterative algorithm. Experimental results on artificial and real-world temporal networks demonstrate that the GNMFCA outperforms existing temporal link prediction algorithms. Yang et al.[256] introduced a novel anchor link prediction method, Multiple Consistency (MC), that leverages interlayer and intralayer structures to improve performance. The MC method iteratively leverages interlayer structural information and employs matrix-factorization-based network representation learning to capture global structural features of nodes. It further trains a radial-basis neural network as a mapping function to align embedding vectors across different spaces. The method predicts anchor links between node pairs by considering interlayer and intralayer structures. Experimental results demonstrate the superiority of the proposed approach over existing methods. Agibetov et al. [6] introduced an enhanced approach to learning neural graph embeddings by incorporating information from unlikely node pairs, addressing the limitation of traditional methods that truncate such information. Experiments on various networks demonstrate that the proposed approach yields significant improvements in link prediction performance over baseline methods. The research sheds light on the relationship between skip-gram powered neural graph embeddings and matrix factorization, revealing that the accuracy of graph embeddings in link prediction depends on the transformations applied to the original graph co-occurrence matrix. Notably, smoothing low-frequency pair entries rather than truncating them yields better performance. The findings contribute to a deeper understanding of graph embedding algorithms and provide insights for the design of future matrix-factorization-based approaches.

Factorization matrix methods can enhance network structure analysis by removing random noise and identifying multiple link types, improving prediction accuracy. Integrating global and local structure information also helps reduce the impact of random noise. However, a limitation of using matrix factorization for link prediction is that it relies on accurately representing the observed network as a low-rank matrix, which may be infeasible for networks with complex structures. Furthermore, the technique requires substantial computational resources and may lead to overfitting if not properly controlled, particularly in large-scale networks. These challenges make matrix factorization difficult for link prediction in real-world dynamic networks.

1.2.4.4 Other Methods

In this section, we summarize the articles using different methods and approaches. In [160], the authors proposed the QLP quantum algorithm for path-based link prediction across diverse networks. QLP encodes prediction scores for both even and odd-length paths using a controlled continuous-time quantum walk. Through classical simulations, it demonstrates comparable performance to established path-based predictors. The proposal highlights the potential of QLP to achieve a quantum speedup in link prediction, distinguishing it from conventional methods by utilizing quantum computing techniques for calculations and predictions. Kumar et al.[108] proposed a novel strategy for link prediction using Quantum Kernel-enhanced ML models that incorporate local and global information for feature generation. The aim is to develop a quantum-assisted feature-based approach that combines Projected Quantum Kernel (PQK) with ML models to improve prediction performance. By leveraging high-dimensional Hilbert spaces and a mathematical structure similar to quantum mechanics, the proposed approach enhances data for more accurate link prediction. Experimental results show that the quantum-enhanced models, such as PQK-neural networks and PQK-Random Forest classifier, outperform the corresponding classical ML. Comparative analysis of dynamic datasets demonstrates the superiority of the quantum-assisted methodology over individual link prediction approaches and state-of-the-art methods. The article [244] focused on temporal link prediction (TLP) and the need for high-accuracy white-box methods to explain the mechanisms of network evolution. Existing black-box models, such as network embeddings and graph neural networks, provide high prediction accuracy but lack interpretability. To address this limitation, the authors propose the Develop-Maintain Activity Backbone (DMAB) model, which models node dynamics at the microscopic level to predict future links. The DMAB model extracts and quantifies two dynamic properties of nodes: activity and loyalty. Comparative experiments with state-of-the-art black-box methods demonstrate excellent predictive performance and DMAB's ability to capture mechanisms of network evolution. The study highlights the effectiveness of accounting for node dynamics in understanding the temporal network's dynamic link generation process. It emphasizes the need for further exploration of mechanisms of temporal network evolution. In [207],they introduced a fuzzy-based link prediction algorithm, FLP-ID, designed to address the challenges of accuracy and efficiency in growing and multiplex social networks. FLP-ID considers critical factors such as different interaction channels, information diffusion, and group norms to form new connections. The algorithm generates a multiplex network by combining various relationship types and identifies the community structure. It computes node and relative relevance based on fuzzy criteria and group norms. By calculating the likelihood score of each non-existing link, FLP-ID predicts missing links with improved accuracy compared to crisp algorithms. Zheng et al. [277] proposed an explainable friend link recommendation method that leverages fusion embedding of heterogeneous context information. It integrates user content interests and external knowledge semantics to develop a fusion user embedding method. Using collaborative neighbourhood attention mechanisms, the method calculates direct and indirect similarity relationships between user pairs. It also incorporates a hybrid personalized and neighbour attention model for friend link prediction. The proposed method predicts users' friends and explains the link prediction results. In their study, [248] proposed a novel approach to improve link prediction accuracy by combining different link prediction methods. They comprehensively analyzed the hybrid method and introduced the Precision-Noise Ratio (PNR) metric to evaluate the accuracy of uncertain link predictions. They developed a scalable and parameter-free algorithm based on posterior Bayesian estimation to combine different methods. The results showed that the PNR-based combination outperformed traditional combination methods regarding prediction accuracy. Additionally, the proposed approach offered a general and efficient framework for integrating various existing link prediction methods without increasing computational complexity. The effectiveness and efficiency of the strategy were validated through extensive experiments on real datasets. This research provides valuable insights and opens opportunities to enhance link prediction systems by combining existing methods. The article [54] introduced a novel causal model called causal lifting for link prediction tasks with path dependencies, where the outcome of link interventions depends on existing links. Existing causal models are unsuitable for such scenarios because they struggle to identify causal effects or require many control variables. Causal lifting addresses this by allowing the identification of causal link prediction queries using limited intervention data. The article also investigates the use of structural pairwise embeddings, which offer lower bias and better capture the causal structure than traditional node embedding methods such as GNNs and matrix factorization.

1.2.5 Discussion

This section provides a comprehensive comparison of link prediction methods, including similarity-based approaches, ML techniques, DL models (Graph Neural Networks), and dimensionality reduction methods.
Similarity-based methods, which solely rely on the network's structure, offer simplicity and interpretability. They excel in scenarios where nodes with similar network neighbourhoods are likely to form links. For instance, users with common friends may be more likely to become friends on social networks. The input to similarity-based methods comprises the network's adjacency matrix or edge list, which represents the connections between nodes. The output of these methods is a similarity score or ranking that indicates the likelihood of a link between pairs of nodes. However, these methods may not fully capture complex global network dynamics, limiting their applicability in some cases.

ML techniques provide greater flexibility by incorporating network structure, node attributes, and contextual information. The inputs to ML methods include the network's adjacency matrix or edge list, along with any available node attributes or features. These attributes can be node characteristics, such as age or location, used to enrich information for link prediction. By leveraging this additional information, ML methods can capture more intricate relationships, making them well-suited for diverse networks. The output of these methods is a predictive model that can be used to infer potential links between nodes in the network. However, these methods often require a considerable amount of labeled data for training the predictive model, which may not always be available. Additionally, the increased complexity introduced by node attributes and DL models can make the resulting models less interpretable than simpler methods.

Dimensionality reduction methods aim to reduce the dimensionality of the input data while preserving essential network properties. The input of these methods is typically the network structure represented as an adjacency matrix or a graph. Some dimensionality reduction techniques, such as matrix factorization, embedding-based methods, and graph neural networks, can incorporate node attributes as additional input. The output of dimensionality reduction methods is a lower-dimensional representation of the network, often referred to as embeddings. These embeddings capture the essential information of the network in a reduced space. However, the interpretability of the embeddings may be limited. As the original node features are abstracted into lower-dimensional representations, it becomes challenging to directly interpret the meaning of individual dimensions in the embedding space. Therefore, while dimensionality reduction methods offer computational efficiency and robustness for large networks, the interpretability of the resulting embeddings should be considered when analyzing link prediction results.

After selecting suitable input and output configurations for link prediction methods, it becomes important to consider the availability of computational resources, particularly when using GNNs and DL models. These advanced techniques often require substantial computational power and may even require specialised hardware for efficient training and inference.

Each method has advantages and limitations, and the choice of method should be carefully considered in light of the network's characteristics and the application domain. Similarity-based methods may be appropriate for smaller networks with well-defined community structures, owing to their simplicity and interpretability. On the other hand, for larger and more complex networks with available node attributes, ML or GNN-based approaches could be preferred to capture intricate relationships and patterns.

1.2.6 Trends and Gaps

In addition to providing a comprehensive overview of existing link prediction methods, it is essential to discuss field trends and the gaps requiring further investigation. This section examines trends observed in link prediction research with respect to Attributes, Network Type, and Algorithms.

1.2.6.1 In terms of Attributes

One significant trend in link prediction is the increasing emphasis on integrating node and edge attributes. While traditional link prediction methods primarily rely on network topology, researchers have recognized the importance of incorporating additional information to improve prediction accuracy. Researchers aim to capture the diverse factors that influence link formation in real-world networks by considering attributes such as node features and textual content. For example, [246] proposed a method incorporating structural and text features for link prediction. They leverage the textual content of nodes in a social network to capture the semantic similarity between nodes and consider the network's structural properties. By combining these features, their approach yields more accurate link predictions than methods that rely solely on network topology.

Furthermore, Giubilei et al.[68] employed supervised algorithms and extracted attributes such as age and location from user profiles. Integrating attribute information into their approach significantly improved link prediction accuracy by effectively capturing node similarities based on their attributes.

The availability of datasets with comprehensive attribute information is often limited in link prediction. Many existing datasets lack specific attributes or have incomplete information, making it challenging to apply attribute-based methods effectively. This scarcity of datasets hampers the development and evaluation of accurate link prediction models. Moreover, even when datasets are available, they often require preprocessing to address issues such as missing values, outliers, and inconsistencies. Data preprocessing is crucial to ensure the quality and reliability of attribute data. It involves techniques such as data cleaning and normalization to enhance the accuracy and effectiveness of attribute-based link prediction algorithms.

1.2.6.2 In terms of Networks

Another significant trend is the recognition of the dynamic nature of networks. Real-world networks are constantly evolving, with new nodes, edges, and changes in network structure over time. Therefore, link prediction approaches must account for temporal dynamics and capture the changing nature of connections. Researchers [26, 89] are actively investigating dynamic link prediction techniques that can model and predict link formations over time. However, challenges remain in developing accurate and scalable methods for handling large-scale, dynamic networks. Future research should focus on designing efficient algorithms and techniques to capture temporal patterns and predict links in evolving networks. Another critical trend is the use of a knowledge network. This involves incorporating additional information or domain knowledge about the nodes, edges, or the network structure. External knowledge can provide valuable insights and enhance prediction accuracy by capturing contextual information, domain-specific relationships, or expert knowledge. Using external knowledge underscores the importance of considering information beyond the network structure and leveraging relevant information to improve link prediction performance.

1.2.6.3 In terms of Algorithms

In terms of algorithms, one notable trend in link prediction is the increasing use of dimensionality reduction techniques, particularly in conjunction with graph neural networks (GNNs). These techniques address the challenge posed by high-dimensional feature spaces in network datasets by reducing the input dimensionality while preserving relevant information. By doing so, these algorithms can improve the efficiency and effectiveness of link prediction models.

However, gaps and challenges remain in this area. One such gap is the need for algorithms with lower complexity and reduced computational resource requirements. As network datasets continue to grow in size and complexity, it becomes increasingly important to develop algorithms that efficiently handle the computational demands. This can involve exploring more efficient optimization strategies, model architectures, and algorithms that scale well to large networks.

1.3 Related Works on Time Series Forecasting

In this section, we define the time series forecasting and overview the approaches proposed in the literature.

1.3.1 Definition of Time Series Forecasting

A time series (TS) is a sequence of data points collected over time (as shown in Figure 1.6), either at regular or irregular intervals, that provides a temporal record of observed phenomena such as weather, sales trends, stock prices, vital signs, conditions, and more. These data points may include numerical values, categorical labels, or other formats. They are widely used for analyzing and forecasting patterns and trends that change over time [159] TSF denotes the process of using historical time-series data to predict future values or trends. By leveraging large volumes of sequential observations, TSF enables the identification of temporal patterns and dynamics across diverse domains [211]. Its applications are extensive, ranging from financial market prediction [76, 66, 5] and weather forecasting [262, 132], to energy consumption modeling [11, 104], healthcare analytics [42, 20], and beyond. We focus our review on time series forecasting in the economic field, as it is closely aligned with our research in international trade flow.

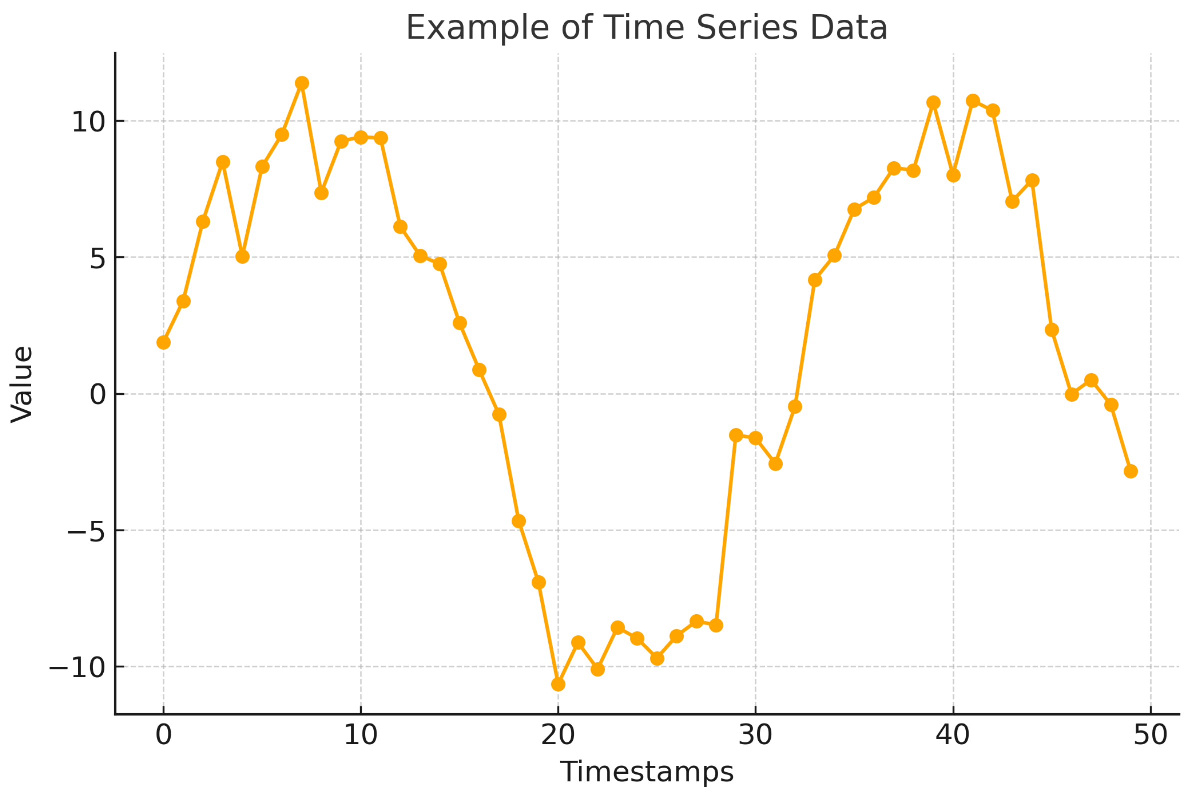

Figure 1.6: Example of a time series dataset.

1.3.2 Methods used for TSF

This section reviews traditional statistical and modern ML methods for time-series forecasting, emphasizing their respective strengths and limitations. A foundational approach, the gravity model of international trade, introduced by Tinbergen [219], posits that trade flows are determined by economic size and geographic distance and has proven effective for analyzing export potential and trade agreements. However, it cannot capture the temporal dynamics essential for understanding evolving trade patterns. Classical time series models like ARIMA and Prophet address this by leveraging historical data to forecast future trends. More recently, ML and DL approaches have advanced the field by offering greater flexibility in modeling complex, nonlinear relationships and capturing intricate temporal patterns. Table 1.5 summarizes these models and their trade-offs, offering a comparative perspective on their capabilities.

1.3.2.1 Conventional Model for TSF

In the early stages of time-series forecasting, relatively straightforward models such as ARIMA [33] and Prophet [216] were commonly employed. These models are well-regarded for their ease of implementation and their ability to capture linear patterns in time-series data [259].

ARIMA, introduced in the 1970s by Box and Jenkins [33], became one of the most widely used parametric models for time-series prediction. Its strength lies in modeling data with trends and seasonality, making it particularly useful in applications such as international trade forecasting. For instance, Elsayir [62] demonstrated the effectiveness of ARIMA in predicting Saudi-US trade flows under dynamic trade conditions. Similarly, Sharma in [200] applied ARIMA to model trade deviations between India and China during the COVID-19 pandemic, highlighting its adaptability to global disruptions.

However, despite its versatility, ARIMA has notable drawbacks. It struggles with unexpected external shocks, such as those observed during the pandemic, in which sudden trade disruptions render its linear assumptions less effective. Furthermore, ARIMA requires substantial pre-processing, particularly when handling non-stationary data, which can risk information loss if not handled carefully. Identifying the correct parameters (p, d, q) is a complex and iterative process that can lead to overfitting or underfitting if not appropriately done [62, 91].

Facebook developed Prophet as a more flexible and user-friendly forecasting tool, particularly well-suited for business time-series data, in response to some of ARIMA's limitations. Unlike ARIMA, Prophet handles non-linear trends more effectively by incorporating features such as yearly, weekly, and daily seasonality and accounting for holiday effects, making it a practical alternative when additional complexities are present in the data [216].

The authors of [73] sought to improve the accuracy of oil price forecasting, which is crucial given the volatility and economic impact of crude oil prices. They applied two models, LSTM and Facebook's Prophet (FBPr), using a 32-year dataset of weekly Brent oil prices. The data was split into a training set (25 years) and a test set (7 years). Their findings showed that LSTM outperformed FBPr, achieving a higher coefficient of determination (R^2) of 0.92 in training and 0.89 in testing, compared to FBPr's 0.89 in training and 0.62 in testing. In [22], the authors applied time series forecasting techniques to predict the future export quantities of Turkey's hazelnuts. They used Facebook's Prophet algorithm, combined with the Box-Cox power transformation, to improve forecast accuracy. Additionally, they employed the Augmented Dickey-Fuller test and the autocorrelation function to assess the stationarity and periodicity of the time series data. The analysis projected an upward trend in hazelnut exports over the 36 months from June 2020, with clear seasonal patterns in export quantities.

Unlike non-parametric methods, these conventional approaches offer a notable advantage in international trade flow forecasting, as they do not require large datasets for training. However, this can also be a drawback, as they cannot leverage the wealth of information in larger datasets. Moreover, these methods often restrict themselves to specific linear trade patterns and typically represent only one trade relationship per model. This leads to extensive and time-consuming model development when dealing with multiple trade flows. Moreover, they are susceptible to even minor data drift, which can significantly reduce their prediction accuracy.

1.3.2.2 Machine Learning Model for TSF

ML models have increasingly become essential for time series forecasting due to their ability to capture complex relationships and adapt to new data. Unlike conventional statistical methods that require manual feature selection and iterative model tuning, ML models can automate these processes, efficiently managing large and diverse datasets to improve predictive accuracy. For example, Tiits et al. [218] successfully used an RF model to build a gravity model for international trade, handling monthly data and predicting exports across more than $5,000$ product groups.

However, the RF model's extrapolation limitations may limit its utility for predicting future time flows. Therefore, various alternatives have been sought. Graph-based ML models are an especially appealing class of models for analysing international trade.

Sellami et al. [196] introduced a GNN-based method for predicting trade value between countries. The study used Graph Convolutional Networks (GCN) and Graph Attention Networks (GATs) to model complex trade relationships using detailed UN Comtrade data, including over $5,000$ product groups. This approach represents countries as nodes and trade flows as edges in a graph. GCN aggregates information from a node's neighbors to learn embeddings that capture each node's structural and feature-based context. At the same time, GAT enhances this process by applying attention mechanisms to weigh the importance of different neighbors, allowing for more nuanced representation learning. This GNN-based method demonstrated superior predictive accuracy compared with traditional ML techniques such as random forests.

Minakawa et al. [152], and Monken et al. [158] respectively proposed innovative GNN-based models, such as the Gravity-informed Graph Auto-encoder (GGAE) and AINET, that effectively tackle edge-weight prediction and outlier events in trade. Ahmed et al. [8] developed a GNN method for modeling product relationships, and Panford-Quainoo et al. [174] explored the synergy between trade gravity models and GNN, highlighting the growing relevance of GNN in international trade analysis. Verstyuk et al. [228] further supported the use of GNN by demonstrating their theoretical and practical benefits in examining trade accessibility.

Most traditional ML regression models, such as random forests, are not ideal for time series forecasting because they disregard the sequential nature of time-dependent data. These methods assume that each feature at a given time step is independent of its value at previous time steps, which contradicts the inherent serial correlation in time-series data. Due to their ability to capture dependencies across time steps, LSTMs have gained widespread adoption for time-series forecasting. For example, [253] and [201] show that combining DL techniques and LSTM models on large historical datasets can make time-series predictions of trade more accurate. However, these deep neural networks often struggle with capturing long-range dependencies. Similarly, GNNs face their own set of challenges. They encounter scalability issues as graph size grows, leading to high computational costs. Additionally, GNNs often struggle with interpretability, making it difficult to understand how they derive their predictions and insights. Transformer architectures have overcome this limitation.

1.3.2.3 Transformers Model for Time Series Forecasting

In recent years, transformer-based models have become increasingly popular for time series forecasting due to their ability to capture complex temporal dependencies and efficiently handle long sequences. Vaswani et al. [227] introduced the Transformer model, a significant advance in DL architecture and a compelling alternative to traditional recurrent and convolutional networks.

Various transformer-based models have been employed for TSF. For instance, Liu et al. [131] applied the Vanilla Transformer for traffic forecasting, leveraging its ability to capture temporal patterns. Li et al. [118] employed the Informer to improve the accuracy of multi-step predictions of building energy consumption, demonstrating its efficiency in handling long sequences. [235] used the Autoformer for electricity demand forecasting. Additionally, the architecture has led to the development of advanced models such as the Temporal Fusion Transformer (TFT) [124]. This diverse range of models underscores the Transformer's versatility and growing importance in time-series forecasting. TFT [124] tailors the transformer's architecture to time series forecasting. Incorporating attention mechanisms, gating systems, and the ability to handle both static and dynamic features makes it well-suited for complex forecasting tasks across diverse domains. A recent study [19] proposed a TFT-based model to predict international raw material trade flows. The study demonstrated that the model outperformed conventional models such as Random Forest and Graph Attention Network, achieving superior forecasting accuracy. The TFT has also been successfully applied to electricity load forecasting [90] and energy consumption prediction at daily, weekly, and monthly levels [170].

However, these transformer models pose significant challenges despite their effectiveness, particularly in optimizing numerous hyperparameters such as learning rates, batch sizes, and network architectures. Additionally, these models often struggle to transfer knowledge across domains. To overcome these limitations, recent research has shifted toward language models, which have shown impressive results on natural language processing (NLP) tasks and are now being applied to time series forecasting, offering potential solutions to these challenges.

Language models have demonstrated remarkable effectiveness across various fields, such as education and research [86], healthcare [195, 165, 242], and financial modeling [121, 275]. Recently, their capabilities have expanded into time series forecasting. Advances in language model technology are now being applied to predict future values from historical data with greater accuracy and insight.

In the context of time series forecasting, two primary approaches have emerged for applying LLMs:

1. Fine-Tuning: is the process of adapting a pre-trained model to a specific task or domain by continuing its training on a domain-specific dataset that differs from the original dataset used to train the base model [13]. Xue et al. [249] fine-tuned PLMs such as T5 and RoBERTa for energy consumption forecasting, while Lai et

al. [111] developed BERT4ST for spatio-temporal wind power forecasting. These approaches often require extensive computational resources and task-specific data.

2. Prompting Techniques: Prompt-based methods leverage LLMs without retraining, enabling them to adapt to new tasks without fine-tuning. Cao et al. [39] introduced TEMPO, a framework that uses GPT-3.5 with prompts designed to decompose time-series data into trend, seasonal, and residual components. Gruver [72] employed zero-shot prompting with LLaMA 2 and GPT-3 for customer flow and temperature forecasting, demonstrating the efficacy of LLMs even without fine-tuning. Yu et al. [261] explored the use of GPT-4 and Open LLaMA for stock price prediction, employing zero-shot and few-shot prompting, along with Chain of Thought (CoT) reasoning, to improve predictive accuracy. Xue et al. [249] used various PLMs, including GPT-3.5-turbo, to forecast temperature, energy consumption, and customer flow via zero-shot prompting. These studies underline the potential of advanced prompting techniques, such as Zero-Shot-CoT and Few-Shot-CoT, to improve forecasting without extensive model retraining.

Table 1.4 summarizes recent studies on using LLMs for time series forecasting, highlighting models, techniques, and application areas.

Table 1.4: Overview of recent research on language models for time series forecasting

Year	Ref.	Models	Techniques	Domain
2023	[249]	Bart, BERT, T5, RoBERTa, Electra, Bigbird, ProphetNet, LED, Blenderbot, and Pegasus. GPT3.5-turbo.	Fine-tuning, Prompting technique (Zero-Shot)	Temperature, Energy consumption, Customer flow
2023	[250]	Bart, Pegasus, Bigbird	Fine-tuning	Energy load
2023	[261]	GPT-4 and Open LLaMA	Prompting technique (Zero-Shot, Few-Shot)	Stock price
2023	[39]	GPT-3.5	Prompting technique (Zero-Shot)	Electricity, Traffic, Weather, News, Finance
2024	[111]	BERT4ST	Fine-tuning	Wind power
2024	[72]	LLaMA 2, GPT-3	Prompting technique (Zero-shot)	Temperature, Energy consumption, Customer flow

Most studies in this field focus on fine-tuning LLMs, which is effective but requires substantial time and computational resources, even for small datasets. Fine-tuning adjusts the model's settings through repeated optimization. In contrast, prompting techniques allow the use of LLMs directly, without changing their settings or requiring additional training. These methods can be highly effective by providing the model with sequences of input values and asking it to predict future outcomes. However, most studies that use prompting consider only historical data, thereby missing opportunities to incorporate other features that could provide better insights. Relying solely on past data underscores the need to incorporate diverse input types to improve time-series forecasting.

Table 1.5: Strengths and limitations of time series forecasting models

Method	Advantages	Limitations
Gravity [219]	Simple and robust predictive power. Effective for analysing export potential and trade agreements.	Lacks temporal dynamics. Cannot capture evolving trade patterns over time.
ARIMA [33]	Good for linear time series data. Handles trends and seasonality well.	Struggles with external shocks. Requires extensive pre-processing and parameter tuning.
Prophet [216]	Flexible and user-friendly. Captures non-linear trends and seasonal patterns.	Limited performance with highly complex and high-dimensional data.
RF [34]	Handles large datasets and complex relationships. Automates feature selection.	Poor at extrapolation. Disregards the sequential nature of time series data.
LSTM [84]	Captures dependencies between time steps. Effective for long historical datasets.	Struggles with long-range dependencies. Requires significant computational resources.
GNN [193]	Models complex trade relationships. Captures structural and feature-based context.	Scalability issues. High computational cost. Limited interpretability.
Informer[280]	Efficient for long sequences Reduces memory usage	May overlook short-term dependencies. Limited interpretability.
Autoformer [243]	Captures periodic patterns. Suitable for seasonal data. Introduce a decomposition architecture and an auto-correlation mechanism.	Struggles with irregular or non-seasonal data.
TFT [124]	Gating Mechanisms. Captures short- and long-term dependencies. Interpretable. Handles multivariate data. Excels at multi-horizon forecasting.	High computational cost. Requires large datasets. Sensitive to hyperparameter tuning.
LLM (Prompting technique) [147]	Generate predictions directly from input sequences without additional training. Offering a more resource-efficient alternative.	Results cannot be interpreted due to the model's black-box nature. Small wording changes in the prompt can substantially affect the results. The model is constrained by its token window, which limits the amount of historical context that can be provided.

1.3.3 Evaluation Metrics

In this section, we present the measures commonly used to evaluate the results of TSF. The chosen metrics encompass mean squared error (MSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE), the mean percentage error (MPE), and the coefficient of determination R-square defined respectively by (5.1), (5.2), (5.3), (5.4), and (1.10).

1. Mean squared error (MSE)
   MSE measures the average squared error between predicted and actual values. It quantifies the variance of the residuals and is sensitive to large errors.

$$MSE = \frac{1}{I}\sum_{i=1}^{I}(y_i - \hat{y}_i)^2 \tag{1.6}$$

2. Mean absolute error Mean Absolute Error (MAE) is a commonly used metric to measure the accuracy of a regression model. It computes the mean of the absolute differences between predicted and actual values.

$$MAE = \frac{1}{I}\sum_{i=1}^{I}|y_i - \hat{y}_i| \tag{1.7}$$

3. Mean absolute percentage error (MAPE) Mean Absolute Percentage Error (MAPE) calculates the average absolute percentage difference between the predicted and actual values. It expresses prediction errors as a percentage, making it easy to interpret across different scales.

$$MAPE = \frac{1}{I}\sum_{i=1}^{I}\frac{|y_i - \hat{y}_i|}{y_i} \tag{1.8}$$

4. Mean percentage error (MPE) Mean Percentage Error (MPE) is a statistical metric used to evaluate the forecast bias of a predictive model. It is calculated as the average of the percentage differences between the actual values and the predicted values

$$MPE = \frac{1}{I}\sum_{j=1}^{I}\frac{y_i - \hat{y}_i}{y_i} \tag{1.9}$$

5. R-square (R^2), also known as the coefficient of determination, is a statistical measure that indicates how well a regression model explains the variability of the target (dependent) variable.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{1.10}$$

   In this context, y_i denotes the observed values, $\hat{y}_i$ represents the predicted or estimated values, and $\bar{y}$ represents the mean of the observed values. Each metric plays a distinct role in quantifying prediction accuracy, precision, and overall goodness-of-fit by measuring discrepancies between predicted and actual values.

1.4 Conclusion

This chapter has surveyed the principal methods for trend analysis in link prediction and time series forecasting, highlighting their strengths and limitations. Traditional similarity-based approaches, such as Common Neighbors and Jaccard Index, are interpretable and straightforward but often insufficient for capturing complex structural dynamics. Integrating these methods with ML models enhances predictive capability by leveraging latent patterns in the data. In international trade, transformer-based architectures such as TFT have demonstrated superior performance through attention and gating mechanisms, particularly in economic applications. Furthermore, LLMs, exemplified by GPT, offer additional forecasting potential via prompting, enabling adaptation and generalization without full retraining.

Despite these advances, existing methods often address structural and temporal challenges separately, limiting their ability to exploit complementary information across domains fully. In the following chapters (3, 4.5, 5), the present work proposes a hybrid approach that combines similarity-based techniques with ML for link prediction, and leverages TFT and TFT with LLM-based methods for international trade forecasting. By integrating these approaches, the research aims to enhance trend analysis capabilities, capturing complex patterns over time and across networks, thereby providing a more comprehensive, AI-driven framework for predictive analytics in high-dimensional, dynamic domains.

Chapter 2

Background

2.1 Introduction

This chapter presents the essential background required to understand the methodological foundation of this thesis. We begin by reviewing core ML and DL approaches commonly used for predictive modeling, with particular attention to trend analysis in time series forecasting and link prediction. We then introduce the Transformer architecture and, more specifically, the TFT, which forms the basis of our forecasting framework. Finally, we discuss the role of LLMs and prompting strategies, highlighting their emerging relevance in prediction tasks.

It is important to note that the methods covered in this chapter represent only the subset most pertinent to this research. Numerous other predictive techniques exist and may be better suited for applications beyond the scope of the present study.

2.2 Machine Learning (ML)

ML, a pivotal branch of AI, enables computational systems to automatically identify patterns and infer relationships from data to support predictive and decision-making processes. In the realm of trend analysis and temporal data modeling, ML provides robust methodologies for extracting hidden structures and temporal dependencies that govern the evolution of complex systems. Depending on the nature of the learning signal, ML techniques are broadly categorized into supervised learning, which involves training models on labeled data to predict future outcomes; unsupervised learning, which aims to discover latent structures within unlabeled datasets; and reinforcement learning, which learns optimal decision policies through interaction with dynamic environments [70]. In the next subsection reviews representative supervised learning algorithms that have been widely applied in trend analysis: link prediction and trade forecasting.

2.2.1 Support Vector Machines(SVM)

A Support Vector Machine[226] is a supervised learning algorithm designed primarily for classification tasks, where it aims to identify an optimal hyperplane that best separates data points belonging to different classes. When the data are not linearly separable in the original feature space, SVM employs kernel functions to map the input features into a higher-dimensional space, enabling linear separability. The key strength of SVM lies in its margin maximization principle, which seeks to maximize the distance between the decision boundary and the nearest data points, thereby improving the model's generalization capability [10].

2.2.2 Decision Tree(DT)

A Decision Tree [177] is a supervised ML algorithm used for both classification and regression tasks. It constructs a hierarchical, tree-like structure in which each internal node represents a test or decision on a feature, each branch corresponds to an outcome of that test, and each leaf node denotes a predicted class label (for classification) or a continuous value (for regression). The algorithm recursively partitions the dataset into smaller subsets based on feature values, aiming to minimize data impurity or uncertainty through splitting criteria such as information gain or Gini impurity. Decision Trees are intuitive and interpretable; however, they are prone to overfitting, particularly when the tree grows too deep or the data is noisy.

2.2.3 K-NearestNeighbors (kNN)

The K-Nearest Neighbors algorithm [55]—also known as an instance-based or lazy learning method—is a non-parametric approach that classifies data points based on their similarity to nearby examples in the feature space. For a given instance, the algorithm determines its k nearest neighbors using a distance metric such as Euclidean distance, and assigns the class label most common among these neighbors. KNN has been widely applied in domains such as crash prediction and injury severity classification, owing to its simplicity and intuitive design. However, its performance depends on the choice of k and the distance metric, and it can become computationally expensive for large datasets.

2.2.4 Random Forest (RF)

Random Forest [34] is a widely used ensemble learning algorithm that enhances predictive performance by aggregating the outputs of multiple decision trees. Introduced by Breiman, the method constructs an ensemble of trees, each trained on a randomly sampled subset of the data using the bootstrap aggregating (bagging) technique. At each tree split, a random subset of features is selected, thereby enhancing model diversity and mitigating overfitting. The final prediction is obtained by averaging predictions (for regression) or by majority voting (for classification) across all trees, yielding a model that is both robust and scalable across diverse predictive tasks.

2.2.5 Logistic Regression (LR)

Logistic Regression [85] is a widely used statistical and ML algorithm primarily designed for binary classification tasks, where the goal is to estimate the probability that an instance belongs to one of two possible classes (e.g., linked vs. not linked). Unlike Linear Regression, which predicts continuous outcomes, Logistic Regression models the log-odds of the dependent variable using the sigmoid (logistic) function, thereby constraining predictions to the [0, 1] interval. The model is valued for its simplicity, interpretability, and efficiency on linearly separable data; however, its performance can degrade when relationships between variables are highly nonlinear or complex.

2.2.6 Naive Bayes (NB)

Naïve Bayes classifiers [28], including the Gaussian Naïve Bayes (GNB) variant [28], are probabilistic learning algorithms founded on Bayes' theorem. They operate under the simplifying assumption that the predictor variables are conditionally independent given the target class. In the Gaussian variant, each continuous feature is modeled as a standard (Gaussian) distribution, enabling efficient computation of class-conditional probabilities. By applying Bayes' theorem to a set of input features, Naïve Bayes models estimate the posterior probability of each class and make predictions based on the most probable class. Owing to their simplicity, computational efficiency, and robustnes.

2.2.7 Artificial NeuralNetwork (ANN)

Artificial Neural Networks [149] are computational models inspired by the structure and functioning of biological neurons in the human brain. An ANN typically consists of an input layer, one or more hidden layers, and an output layer, in which interconnected processing units—called neurons—transform input features into outputs via weighted connections. Each input feature is multiplied by an associated weight and combined with a bias term to produce a weighted sum, which is then passed through an activation function to introduce nonlinearity. The network learns by adjusting these weights using an iterative optimization process known as backpropagation, which minimizes the error between predicted and actual outputs. Owing to their ability to model complex nonlinear relationships, ANNs have been successfully applied in a wide range of predictive tasks, including trend analysis, link prediction, and time-series forecasting.

In summary, classical ML algorithms have provided effective solutions for a wide range of predictive modeling problems, including trend analysis, link prediction, and time-series forecasting. Despite their success, these models often depend on extensive feature engineering and struggle to capture complex temporal dependencies and nonlinear interactions among variables. As datasets become increasingly high-dimensional and dynamic, such limitations hinder their scalability and accuracy.

These challenges have driven the development of DL, which extends the capabilities of traditional ML by employing multi-layer neural architectures that automatically learn hierarchical feature representations from raw data. The following section explores fundamental DL architectures—most notably CNNs for spatial pattern extraction and LSTM networks for modeling sequential dependencies—laying the foundation for more advanced architectures such as Transformers and TFT.

2.3 Deep Learning (DL)

DL is a subfield of ML that employs ANNs with multiple processing layers—commonly referred to as deep neural networks—to automatically learn hierarchical patterns and feature representations from data. These layers enable the model to capture increasingly abstract and complex relationships within the input, thereby mimicking, to some extent, the way the human brain processes information. In contrast to traditional ML models that rely heavily on manual feature engineering, DL systems can learn directly from raw data, such as images, text, or numerical time-series data.

Owing to this powerful capability to model complex and nonlinear dependencies, DL has become particularly effective in a wide range of predictive modeling tasks, including trend forecasting, classification, and regression. By leveraging large datasets and high computational power, DL models can achieve superior predictive accuracy and generalization compared to classical ML approaches, making them a cornerstone of modern artificial intelligence research.

2.3.1 Convolutional Neural Network (CNN)

Convolutional Neural Networks [9] are a class of DL architectures designed to automatically learn spatial or structured patterns from input data through the use of convolutional and pooling layers. While CNNs are best known for their success in image recognition and computer vision, they have also demonstrated strong performance in analyzing structured numerical and sequential data. By capturing local dependencies and spatial relationships among input features, CNNs can effectively distinguish between classes in classification tasks and detect complex interaction patterns within the data.

2.3.2 Long Short Term Memory (LSTM)

Long Short-Term Memory networks [71] are an advanced type of RNN designed to model and learn from sequential or time-dependent data. They are particularly effective for predictive modeling tasks in which capturing temporal patterns and long-range dependencies is critical. Unlike standard RNNs, which suffer from the vanishing gradient problem, LSTMs incorporate memory cells and gating mechanisms—including input, output, and forget gates—that selectively retain or discard information over extended sequences. This architecture enables LSTMs to maintain context over time, yielding more accurate predictions from historical data. Owing to these properties, LSTMs have achieved strong performance on various time-series forecasting and sequence prediction tasks, including forecasting weather conditions, energy consumption, and financial market trends. Moreover, their predictive capabilities can be further enhanced by integrating them with other DL techniques, such as attention mechanisms and pre-trained neural architectures, which enable the model to focus on the most relevant temporal features and improve generalization across domains.

While CNNs and LSTMs have achieved significant success in capturing spatial and temporal dependencies, both architectures exhibit limitations in modeling long-range dependencies and parallel dependencies in large-scale sequential data. To overcome these challenges, the Transformer architecture was introduced, representing a breakthrough in DL. Unlike RNN-based models, Transformers rely entirely on attention mechanisms, enabling them to capture global context efficiently and process data in parallel. The following section provides a detailed overview of the Transformer framework and its extensions for predictive modeling tasks.

2.4 Transformers

The Transformer model, introduced by Vaswani et al. [227], represents a breakthrough in DL architectures, providing a highly efficient alternative to traditional recurrent and convolutional neural networks. Central to the Transformer's success is its self-attention mechanism, which enables the model to process entire sequences in parallel rather than sequentially, thereby capturing long-range dependencies more effectively. As illustrated in Figure 2.1, the Transformer architecture consists of two primary components: an encoder and a decoder.

- The encoder consists of two primary components: a multi-head self-attention mechanism, and a position-wise feed-forward network (FFN). Within each encoder block, the self-attention module enables the model to capture contextual dependencies across all tokens in the input sequence. At the same time, the FFN applies nonlinear transformations independently to each position to produce richer token representations. To maintain stable and scalable training, each sub-layer is enclosed within a residual connection [82] followed by Layer Normalization [24], a design that facilitates effective gradient propagation and enhances training robustness [128]. The final encoder representations generated by the stacked encoder layers serve as the input to the decoder.

- The decoder consists of three primary components: a masked multi-head self-attention layer, an encoder–decoder (cross) attention layer, and a position-wise FFN. The masked self-attention ensures that each position can attend only to previous tokens, preventing the model from accessing future information during generation. The cross-attention layer enables the decoder to attend to the encoder outputs, aligning the generated tokens with relevant parts of the input sequence. As in the encoder, every sub-layer is wrapped with a residual connection followed by layer normalization, facilitating stable optimization and deeper stacking of decoder blocks. Together, these components allow the decoder to integrate contextual information from both past output and encoded input representations [98].

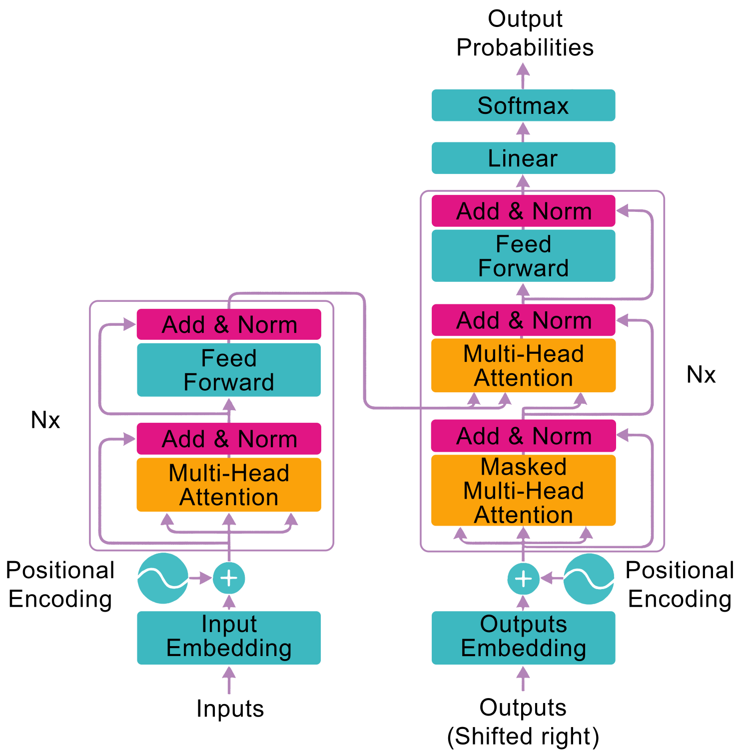

Figure 2.1: Transformer model architecture .

The Transformer has become a foundational model in DL, achieving widespread adoption across domains such as NLP, CV, SA, and TSF. Within the TSF domain, several Transformer-based architectures have been proposed that effectively capture temporal dependencies and complex patterns in sequential data. Notable examples include the Autoformer [243], which introduces an auto-correlation mechanism for efficient sequence modeling; the Informer [280], optimized for handling long input sequences with reduced computational cost; and the TFT [124], which combines attention mechanisms with interpretable outputs for multivariate time-series forecasting.

These advancements highlight the flexibility and adaptability of the Transformer framework in modeling temporal dynamics across various domains. The following section provides a detailed overview of the TFT architecture.

2.4.1 Temporal Fusion Transformers (TFT)

The Temporal Fusion Transformer [124] is an advanced DL architecture designed to model complex temporal relationships and provide interpretable multivariate time-series forecasts. Building upon the sequential modeling strengths of LSTM, the TFT integrates several specialized components to capture both short- and long-term dependencies in data. Key elements of the architecture include support for multiple input types, a Variable Selection Network for dynamically determining feature relevance, a Gated Residual Network for nonlinear transformation, and a Multi-Head Attention Mechanism to capture global temporal patterns. A simplified representation of the TFT architecture is provided in Figure 2.2.

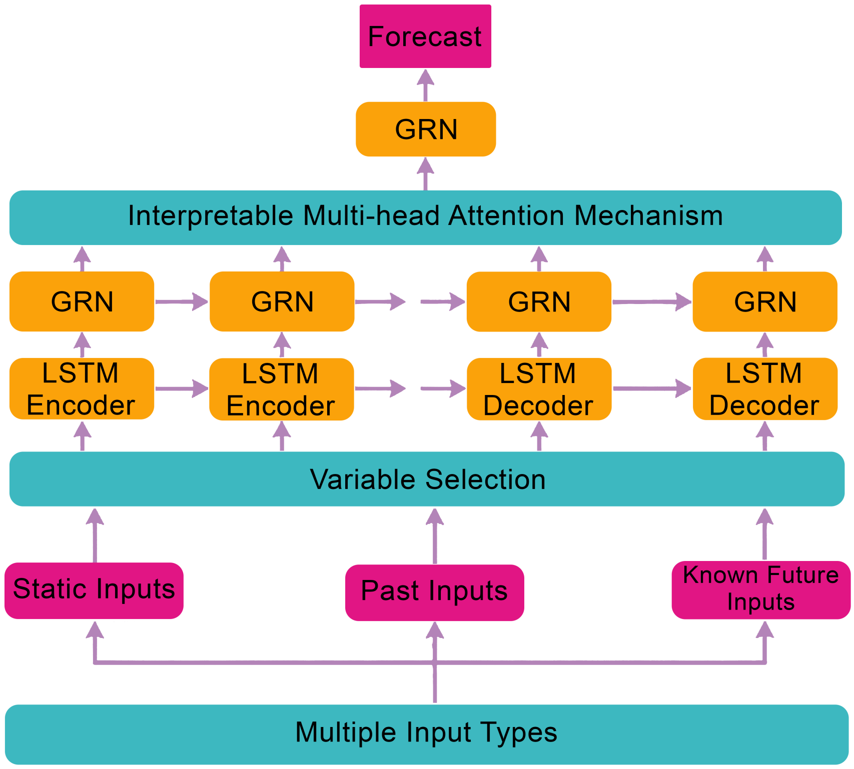

Figure 2.2: TFT architecture

2.4.1.1 Multiple Input Types

The TFT model uses three input categories [124]: past input (historical, unknown inputs), known future inputs, and static inputs. These inputs are processed through separate variable selection networks (Sec.2.4.1.2).

- Past input: past values of the target variable within a look-back window of length k.
- Known future inputs: indicators that can be predicted accurately in advance, such as calendar events.
- Unknown inputs: indicators that can only be measured during observation and collected through statistical methods; they cannot be predicted for future values.
- Static inputs: indicators reflecting the inherent properties of the forecast object, such as products and countries.

The presented equation is:

$$\hat{y}_i(q,t,\tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i) \tag{2.1}$$

Where:

- i: denotes the prediction objects within the system
- t: current moment.

- k: step length.
- τ: the prediction horizon.
- $\hat{y}_i(q, t, \tau)$: the predicted value for quantile q, at time t, for prediction horizon τ.
- f: the function that generates the prediction.
- y: historical target .
- z: unknown inputs.
- x: known inputs.
- s: static inputs.

Although multiple variables can be used as inputs to a forecasting model, the relationships among these variables and their influence on predictive performance are often uncertain. The Variable Selection Network (VSN) introduced in the Temporal Fusion Transformer (TFT) [124] dynamically assesses the relevance of each input feature at each time step. It filters out noisy or redundant information that could negatively affect forecasting accuracy. This mechanism enhances both the model's performance and interpretability.

To represent input features, the TFT employs entity embeddings for categorical variables and linear transformations for continuous variables. Let

2.4.1.2 Variable Selection Network

Although multiple variables can be used as inputs to a forecasting model, the relationships among these variables and their influence on predictive performance are often uncertain. The Variable Selection Network introduced in TFT dynamically determines the relevance of each input feature at each time step and filters out noisy or redundant information that could negatively affect forecasting accuracy. This mechanism enhances both the model's performance and interpretability.

To represent input features, the TFT employs entity embeddings for categorical variables and linear transformations for continuous variables.

Let $n_t^{(j)} \in \mathbb{R}^{d_{\text{model}}}$ be the flattened version of the j-th variable at time t and m denote the total number of input variables. Then,

$$N_t = \left[n_t^{(1)} \| \cdots \| n_t^{(m)} \right]^\top$$

is the flattened vector of all past input combinations at time t. The input N_t and external environment vector c_s are applied to the GRN, followed by a Softmax layer:

$$v_t = \text{Softmax}\left(\text{GRN}_\text{t}(N_t, c_s)\right) \tag{2.2}$$

Where $v_t \in \mathbb{R}^m$ is a variable selection weights vector. A static covariate encoder obtains the environment vector c_s and includes static information. At each time step, the input $n_t^{(j)}$ is fed into a nonlinear GRN, as shown in Eq.2.3:

$$\tilde{n}_t^{(j)} = \text{GRN}_{\tilde{n}_t^{(j)}}(n_t^{(j)}) \tag{2.3}$$

Although each variable has a separate $\mathrm{GRN}_{\tilde{n}_t^{(j)}}$, the weights are shared at all time steps t. The final features are obtained as the weighted sum of each variable and its variable selection weight, as shown in Eq.2.4:

$$\tilde{n}_t = \sum_{j=1}^{m} v_t^{(j)} \tilde{n}_t^{(j)} \tag{2.4}$$

Where $v_t^{(j)}$ is the j-th element of v_t.

2.4.1.3 Gated Residual Network (GRN)

Forecasting models often struggle to capture the exact relationships between external inputs and target variables, making it difficult to determine the degree of nonlinearity required for effective modeling. To address this limitation, Lim et al. [124] introduced the GRN (illustrated in Fig. 2.3) as a central component of the TFT to provide flexible nonlinear processing capabilities.

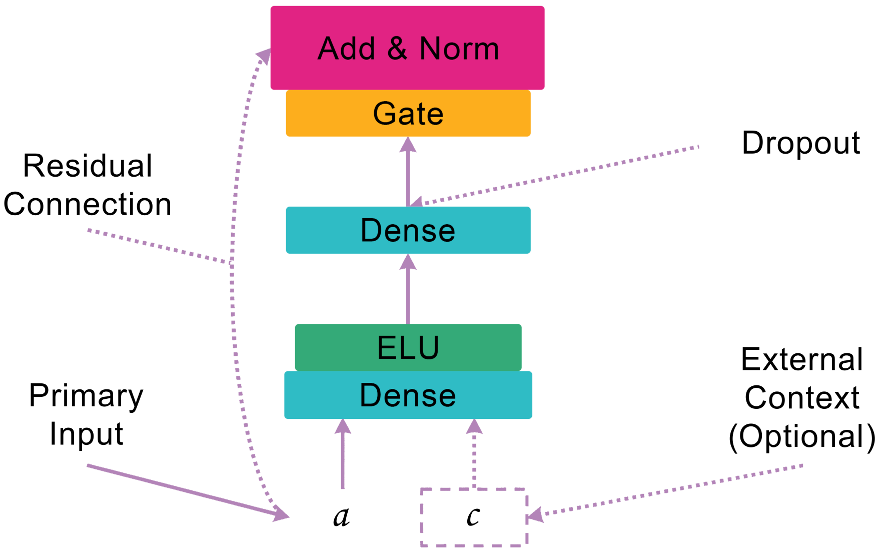

Figure 2.3: Gated residual network structure

The GRN takes a primary input a and an optional context vector c, and computes its output as:

$$\mathrm{GRN}_\omega(a, c) = \mathrm{LayerNorm}\left(a + \mathrm{GLU}_\omega(n_1)\right) \tag{2.5}$$

where

$$n_1 = W_{1,\omega} n_2 + b_{1,\omega} \tag{2.6}$$

and

$$n_2 = \mathrm{ELU}\left(W_{2,\omega} a + W_{3,\omega} c + b_{2,\omega}\right). \tag{2.7}$$

Here, GLU denotes a Gated Linear Unit, ELU represents the Exponential Linear Unit activation function, and LayerNorm refers to standard Layer Normalization. The intermediate layers $n_1, n_2 \in \mathbb{R}^{d_{\text{model}}}$, and ω is an index used to denote shared weights across different GRN instances.

To control the level of nonlinearity and minimize the influence of redundant model components, the GRN incorporates a GLU. Given an input $\gamma \in \mathbb{R}^{d_{\text{model}}}$, the GLU is defined as:

$$\text{GLU}_\omega(\gamma) = \sigma(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega}) \tag{2.8}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $W_{(\cdot)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $b_{(\cdot)} \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters, and $\odot$ denotes the element-wise (Hadamard) product.

The GLU allows the network to modulate the nonlinear contribution of the GRN. If necessary, the model can suppress the nonlinear transformation by driving the GLU output toward zero, effectively skipping the layer. When no context vector is provided, the GRN treats c as zero (i.e., $\mathbf{c} = 0$ in Eq. 2.7). During training, dropout is applied between Eq. (2.5) to regularize learning and prevent overfitting.

2.4.1.4 LSTM Encoders and Multi-Head Attention Mechanism

The TFT leverages both LSTM encoders and a multi-head attention mechanism to capture diverse temporal dependencies in the data. The encoder-decoder component models short-term sequential dependencies, focusing on recent observations and local temporal patterns. In contrast, the multi-head attention mechanism is designed to capture long-range dependencies by identifying relationships among temporally distant observations. This attention framework is derived from transformer-based architectures [227, 120] and allows the TFT to model global temporal context efficiently while maintaining interpretability.

In general, the attention mechanism computes the relevance between inputs by scaling the *values* $V \in \mathbb{R}^{N \times d_V}$ according to the similarity between corresponding *keys* $K \in \mathbb{R}^{N \times d_{\text{attn}}}$ and *queries* $Q \in \mathbb{R}^{N \times d_{\text{attn}}}$, as expressed by:

$$\text{Attention}(Q, K, V) = A(Q, K)V, \tag{2.9}$$

where $A(Q, K)$ is a normalization function and N denotes the number of time steps. A common choice is the **scaled dot-product attention** [227]:

$$A(Q, K) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{attn}}}}\right), \tag{2.10}$$

where d_{attn} is the dimensionality of the attention space. The scaling factor $\sqrt{d_{\text{attn}}}$ mitigates the problem of minimal gradients that may arise when d_{attn} is large. This formulation enables efficient parallel computation of the attention operation, significantly improving training speed.

The **multi-head attention** mechanism extends this idea by employing multiple attention heads, each learning distinct relationships between Q, K, and V. The outputs of the individual heads are concatenated and linearly transformed as follows:

$$\text{MultiHead}(Q, K, V) = [H_1, \ldots, H_{m_H}]W_H, \tag{2.11}$$

where:

$$H_h = \text{Attention}(QW_h^{(Q)}, KW_h^{(K)}, VW_h^{(V)}), \tag{2.12}$$

and $W_h^{(Q)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}$, $W_h^{(K)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}$, $W_h^{(V)} \in \mathbb{R}^{d_{\text{model}} \times d_V}$ are the head-specific projection matrices, while $W_H \in \mathbb{R}^{(m_H \cdot d_V) \times d_{\text{model}}}$ represents the final linear transformation.

By combining recurrent encoding for short-term dependencies with attention for long-range context, the TFT achieves a robust balance between **local temporal modeling** and **global interpretability**, enabling it to capture complex temporal dynamics across varying time horizons.

2.4.1.5 Quantile Outputs and Quantile Loss

The TFT produces quantile-based probabilistic forecasts, enabling it to estimate prediction intervals rather than relying solely on point predictions. This design enables the model to quantify uncertainty and generate forecasts across different confidence levels, typically using quantiles such as the 10th, 50th, and 90th percentiles for each forecast horizon. These quantile estimates are generated through linear transformations applied within the temporal fusion decoder. The TFT is trained by jointly minimizing the quantile loss function [227, 120], which asymmetrically penalizes over- and under-predictions according to the target quantile. The overall training objective is defined as

$$\mathcal{L}(\Omega, \mathbf{W}) = \frac{1}{M\tau_{\max}} \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{\max}} QL(y_t, \hat{y}(q, t-\tau, \tau), q), \tag{2.13}$$

where the quantile loss is given by

$$QL(y, \hat{y}, q) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+, \tag{2.14}$$

and $(x)_+ = \max(0, x)$. Here, Ω denotes the domain of training samples, $\mathbf{W}$ represents the learnable model weights, $Q = \{0.1, 0.5, 0.9\}$ is the set of quantile levels, M is the number of training instances, $\tau_{\max}$ is the maximum forecasting horizon, y_t and $\hat{y}$ are the true and predicted target values respectively, and q denotes a specific quantile level.

This formulation encourages the TFT to produce calibrated quantile predictions that reflect the underlying uncertainty of future outcomes. During evaluation, the model's performance is often assessed using the normalized quantile loss (*q-Risk*), which compares forecast accuracy across quantiles over the prediction horizon. By incorporating quantile forecasting and quantile loss optimization, the TFT provides robust, interpretable, and uncertainty-aware predictions for complex forecasting tasks, such as trade flows, energy demand, and financial time series.

2.4.1.6 Interpretability

The TFT provides built-in interpretability through its modular design, enabling both feature-level and temporal-level analysis of model behavior. Interpretability is achieved through five main components:

1. **Variable Selection Networks:** The variable selection network computes instance-wise feature importance for static inputs, past inputs, and known future inputs. At each time step t, variable selection weights $v_t^{(j)}$ quantify the relative contribution of each input variable j. Aggregated selection weights across the dataset allow for a global ranking of feature relevance.

2. **Static Covariate Encoders:** Static context variables (e.g., entity identifiers, categorical attributes) are encoded and used to condition both variable selection and temporal attention modules. This enables entity-dependent adaptation of model parameters and facilitates interpretability across different groups or contexts.

3. **Temporal Attention Layer:** The multi-head attention mechanism computes attention weights over historical encoder outputs, identifying the most influential past time steps for each forecast. Attention scores indicate temporal relevance and can be visualized to reveal lag effects, seasonality, or regime shifts.

4. **Gating Mechanisms (GRNs and GLUs):** Gated Residual Networks (GRNs) and Gated Linear Units (GLUs) regulate information flow by suppressing uninformative signals and allowing residual connections to pass relevant information. This improves both model robustness and interpretability by preventing overfitting to irrelevant features.

5. **Quantile-Based Outputs:** The output layer produces probabilistic forecasts by making multiple quantile predictions (e.g., the 10th, 50th, and 90th percentiles). These quantile estimates represent predictive uncertainty, providing interpretable prediction intervals and facilitating risk-aware decision-making.

Together, these mechanisms provide a transparent mapping between model inputs, temporal dependencies, and forecast outputs, making the TFT a high-performing yet interpretable DL framework for multivariate time-series forecasting.

TFT represents a significant step forward in time-series forecasting by integrating recurrent neural networks, attention mechanisms, and gating mechanisms into a unified, interpretable DL framework. This combination enables the model to capture both short- and long-term dependencies, dynamically select relevant input features, and provide meaningful insights into variable importance and temporal patterns. While the TFT achieves strong performance and interpretability, its architecture remains specialized for forecasting tasks and can be computationally demanding on large or heterogeneous datasets. Building on the Transformer foundation, recent advances in LLMs extend these capabilities to a broader range of domains through prompting and fine-tuning, offering enhanced scalability and generalization.

2.5 Large language model

2.5.1 Overview

Language models (LMs) are advanced DL models primarily based on the Transformer architecture. Large volumes of text data are used to train the LM, enabling it to adapt to specific tasks. Pre-trained language models (PLMs) employ a ”pre-train and fine-tune” methodology, which first trains a model on a general-purpose dataset, then fine-tunes it for specialized tasks. However, without this task-specific ”fine-tuning,” PLMs struggle to effectively generalize to unseen downstream tasks [99], such as BERT [58], RoBERTa [136], and Electra [52]. LLMs are more advanced PLMs that can process larger datasets without fine-tuning. This is mainly because they are based on DL and use transformer-based architectures. Meta-training techniques further refine these models to align with human preferences. Due to their large size and extensive pretraining on vast amounts of text data, LLMs demonstrate what is known as ‘emerging abilities’ [237, 194]. These abilities allow them to perform exceptionally well on many natural language processing tasks, even without task specific training [98]. LLMs have evolved along two dimensions:

Table 2.1: Overview of various LLMs

Model Name	Trained on Data Up Until	Tokens Input	Provider	Number of Parameters	Open Source
GPT-3 [35]	October 2019	2,048	OpenAI	175 billion	No
GPT-3.5 [50]	June 2021	4,096	OpenAI	175 billion	No
GPT-3.5-turbo (0125) [3]	September 2021	16,385	OpenAI	175 billions	No
GPT-4 (0613) [3]	September 2021	8,192	OpenAI	1 trillion	No
GPT-4-turbo (2024-04-09) [3]	December 2023	128,000	OpenAI	1 trillion	No
LLaMA 2 [221]	July 2023	4,096	Meta	7 billion, 13 billion, 33 billion, 65 billion	Yes
LLaMA 3 [61]	8 billion: March 2023, 70 billion: December 2023	8,000	Meta	8 billion, 70 billion	Yes
Llama 4 Scout (17Bx16E) [203]	August 2024	10M	Meta	17B (Activated) 109B (Total)	Yes

- Closed-source models, such as the ChatGPT family, only allow access via Application Programming Interfaces (APIs) and do not publicly release their model weights, making it impossible for external developers to load or modify their underlying parameters.
- Open-source models, such as the LLaMA family [221], provide access to their model weights, allowing for greater flexibility in experimentation and customization.

Closed-source and open-source models, each offering distinct advantages and use cases.

Various types of LLMs differ in architecture, training data, and the number of parameters, all of which affect their language comprehension and task performance. Table 2.1 overviews the most prominent LLMs.

Language models have demonstrated remarkable effectiveness across various fields, such as education and research [86], healthcare [195, 165, 242], and financial modeling [121, 275]. Recently, their capabilities have expanded into time series forecasting. Advances in language model technology are now being applied to predict future values from historical data with greater accuracy and insight. To adapt LLMs for specific applications, prompting and fine-tuning are among the most widely adopted approaches. The following sections provide an overview of these techniques.

2.5.2 Architecture of Large Language Model

LLM encompass a wide family of architectures that differ in size, training objectives, and implementation details. Despite these variations, most modern LLMs share a common architectural foundation derived from the Transformer framework. In this section, we present the core components that constitute the basic architecture of LLMs. While individual models may introduce modifications or additional mechanisms, the elements described here constitute the essential pipeline underlying most contemporary LLMs.

2.5.2.1 Data Preprocessing

The construction of LLMs critically depends on the availability of high-quality training data, as model performance is directly linked to the dataset's reliability and richness. Consequently, the first stage of the LLM architecture involves assembling a large, diverse corpus from books, academic publications, web pages, and online forums. These sources vary widely in style, structure, and quality, making it essential to preprocess the collected data systematically to ensure that the model learns from accurate and representative content. The preprocessing pipeline typically includes several key steps [182, 169]:

- Data cleaning: where invalid characters, incomplete entries, formatting inconsistencies, outliers, and missing values are corrected or removed;
- Data filtering: which excludes noisy, irrelevant, or low-quality text and reduces computational burden during training;
- Data de-duplication: which eliminates redundant samples to prevent overfitting and promote richer linguistic diversity;
- Data privacy preservation: which removes or anonymizes personally identifiable information (PII) to meet ethical and legal standards.

Together, these steps transform raw textual data into a clean, secure, and structured corpus, forming a reliable foundation for subsequent processes such as tokenization, embedding, and large-scale model training.

2.5.2.2 Tokenization

Tokenization is a fundamental step in the LLM preprocessing pipeline, converting raw text into smaller, machine-readable units called tokens. These tokens may represent characters, subwords, symbols, or entire words, depending on the model's design and objectives [182]. While simple tokenization methods split text based on white spaces, such approaches suffer from the out-of-vocabulary problem, as they can only recognize words present in a predefined dictionary. To address this limitation and increase vocabulary coverage, LLMs rely on subword-based tokenization methods such as Byte Pair Encoding, WordPiece, and Unigram Language Model [169]. These algorithms break rare or unseen words into smaller subword units, enabling the model to process a large variety of linguistic forms—including words not present in the training data—while maintaining efficiency and reducing vocabulary size. Tokenization therefore, plays a crucial role in bridging raw text and numerical model inputs, ensuring consistent representation across languages, domains, and tasks.

2.5.2.3 Embedding Layer and Position Encoding

After tokenization assigns an ID to each token, the embedding layer converts these token IDs into numerical vectors. Each token is mapped to a learned embedding that represents its meaning in a mathematical space. Tokens that occur in similar contexts tend to have similar vector representations. During training, these embeddings are refined to better capture relationships among words or subwords. However, token order is crucial for meaning, and transformer architectures do not inherently encode sequence structure. Positional encoding is introduced to represent the position of each token within the input sequence. These positional vectors are added to the token embeddings, enabling the model to distinguish between tokens based on their order and thus interpret the sentence in a grammatically and semantically appropriate manner [182].

2.5.2.4 Transformer Block

Transformers serve as the foundational architecture for most modern LLMs due to their scalability, flexibility, and suitability for large-scale pretraining. Their self-attention mechanism enables efficient modeling of long-range dependencies within text, allowing the model to capture complex linguistic patterns. As a result, Transformer-based LLMs have driven significant advances across a wide range of natural language processing tasks. A detailed description of the Transformer architecture is provided in Section 2.4.

2.5.3 LLM Adaptation Techniques

Large Language Models can be adapted to different tasks using two main approaches that vary in complexity, flexibility, and computational requirements. These approaches provide complementary means to guide or modify model behavior, depending on the application's needs and resource availability. Before detailing each method, it is essential to introduce the general idea behind how LLMs can be customized, ranging from light-weight instruction-based techniques to more specialized training procedures. The following subsections present the two most widely used adaptation strategies: prompting and fine-tuning.

2.5.3.1 Prompt Engineering

In this section, we define the concept of a prompt and outline prompt engineering techniques, including few-shot, zero-shot, chain-of-thought, tree-of-thought, and self-consistency. These techniques are essential for guiding LLMs to generate accurate and relevant responses across diverse applications.

A prompt serves as the input a user provides to an LLM to influence its responses. This input can take various forms, such as questions, statements, examples, or detailed instructions [151]. The prompt's clarity, structure, and specificity heavily influence the effectiveness of the LLM's output. Prompt engineering involves designing and refining prompts to maximize the quality and relevance of LLM-generated responses [151]. It aims to enhance the LLM's performance and ensure that its responses are more accurate and meaningful. This process shows how important prompt quality is for improving LLMs' generative abilities because it directly affects the type of output that is made [137, 202, 147]. Below, we explore several notable and widely used prompt engineering techniques.

- Zero-Shot-Prompting [178]: It is a technique where the LLM is given a task with only a description or query, without any examples or task-specific training. The model generates a response based solely on its general knowledge and understanding acquired during training. This method assesses the model's capacity to generalize and execute tasks for which it has not received explicit training.

- Few-Shot-Prompting [146]: It builds upon zero-shot prompting by providing the LLM with a few examples, including the input and the corresponding output, alongside the task description. These examples serve as references to help clarify the task's format, expectations, and desired outcome. By presenting the model with input examples and their corresponding outputs, few-shot prompting guides the LLM to generate more accurate and context-aware responses. This approach is beneficial for more complex or ambiguous tasks.

- Chain-of-Thought (CoT) prompting [238]: It is a technique that makes the implicit reasoning process of the LLMs explicit. By outlining the necessary steps for reasoning, CoT guides the model to produce a logical, well-reasoned output, particularly in situations that require more than simple information retrieval or pattern recognition. There are two main ways to apply CoT prompting:

  - Zero-Shot-CoT: This technique directs the model to "think step by step" by breaking down a problem and explaining each reasoning step without any examples provided beforehand.
  - Few-Shot-CoT: This technique provides the model with a few examples of problems and their step-by-step solutions. These examples guide the model in structuring its reasoning process for similar tasks.

  CoT prompting often outperforms traditional, zero-shot, and one-shot methods. It helps the model perform complex reasoning more effectively and yields more accurate, well-structured answers, especially for problems that require a deeper understanding.

- Self-Consistency (SC) [234]: It is a technique that improves the accuracy of LLMs by generating multiple reasoning paths for a single prompt and selecting the most consistent answer. It leverages the observation that complex problems often have multiple valid solutions, and consistent answers across different paths increase confidence in the correctness of the response. This technique enhances both the reliability and accuracy of the model's output.

- Tree of Thought (ToT) [257]: It is a technique used to improve problem-solving in complex tasks, like mathematics. It involves breaking down a problem into multiple steps or "thoughts" and exploring different paths to find the best solution. ToT helps tackle complex issues by organizing intermediate steps into a tree-like structure, enabling more thorough and accurate reasoning.

2.5.3.2 Fine Tuning

There are various strategies for fine-tuning LLM. In this section, we define transfer learning and instruction-tuning [168].

- Transfer learning refers to the process of adapting a pre-trained LLM to a specific downstream task. While pre-trained LLMs demonstrate strong general capabilities across a variety of tasks [35], their performance on a particular application can be significantly enhanced through additional training on task-specific datasets[179, 251]. This process of reusing and fine-tuning a model for a new, specialized task is known as transfer learning.

- Instruction-tuning is a specialized form of fine-tuning in which a pre-trained model is trained on datasets composed of instructions, input, and output. These instructions span a wide range of tasks and are written in plain language to guide the model in accurately following prompts.

As LLMs continue to grow in size, fine-tuning becomes increasingly expensive because it requires updating and storing a massive number of parameters for every new task, posing challenges for memory usage, computational cost, and practical deployment. To address these limitations, more parameter efficient fine tuning methods[59] have been proposed. Among them, Low-Rank Adaptation (LoRA) [88] offers a scalable solution by preserving the core idea of adapting the model to new tasks while drastically reducing the number of trainable parameters. LoRA introduces a low-rank constraint to approximate weight updates in dense layers, thereby enabling model adaptation with substantially fewer trainable parameters.
When fine-tuning a weight matrix W, the update is typically expressed as $W \leftarrow W + \Delta W$. LoRA freezes the original matrix $W \in \mathbb{R}^{m \times n}$ and represents the update term using a low-rank decomposition, $\Delta W = AB^\top$, where $A \in \mathbb{R}^{m \times k}$ and $B \in \mathbb{R}^{n \times k}$ are the only trainable components, with $k \ll \min(m, n)$ controlling the degree of parameter reduction.

2.6 Feature Selection

In ML, feature selection is a fundamental preprocessing step that identifies the most relevant and informative features in a given dataset. High-dimensional data often contain redundant, irrelevant, or noisy attributes, which can degrade model performance by increasing computational complexity, leading to overfitting, and reducing interpretability.

To address these challenges, feature selection aims to reduce the feature space by selecting a subset of features that most significantly contribute to the model's predictive accuracy. This not only enhances learning efficiency and model generalization but also simplifies the model, making it easier to interpret and deploy, reduces the data's dimensionality, resulting in faster training times, and lower computing costs.

Feature selection typically comprises two components: a search strategy to explore candidate feature subsets, and an evaluation mechanism to assess their quality. Based on this, feature selection methods are commonly divided into two categories: filter-based and wrapper-based approaches. Filter methods apply statistical measures—such as information gain [255], mutual information [176], or principal component analysis [153], independent of the learning algorithm. These techniques are computationally efficient but may overlook feature-classifier interactions. In contrast, wrapper methods incorporate an ML model to evaluate feature subsets directly, often achieving higher predictive accuracy but at the cost of increased computation. One of the most prominent wrapper-based algorithms is the binary Butterfly Optimization Algorithm, which we describe in the next section.

2.6.1 Binary Butterfly Optimization Algorithm (BBOA)

The Binary Butterfly Optimization Algorithm (BBOA) [15] is a discrete variant of the original Butterfly Optimization Algorithm (BOA) [16], a bio-inspired metaheuristic based on the food foraging behavior of butterflies. While BOA operates in a continuous search space, BBOA has been specifically adapted for binary optimization tasks, such as feature selection. In BBOA, each butterfly represents a potential solution encoded as a binary vector, where each bit signifies the selection (1) or exclusion (0) of a specific feature.

The algorithm retains BOA's fundamental biological inspiration, where butterflies emit a "fragrance" based on their fitness, calculated using:

$$\mathrm{pf}_i = c \cdot I^a \tag{2.15}$$

Here, pf_i represents the fragrance of the i-th butterfly, I is a stimulus intensity proportional to the fitness value (e.g., classification accuracy), c is a sensory modality constant, and a is a power exponent that regulates fragrance diffusion.

BBOA operates using two primary search mechanisms:

- **Global Search (Exploration)**
  With probability p, a butterfly is attracted to the globally best solution found so far. The movement is formulated as:

  $$\mathbf{F}_i^{t+1} = (r \cdot \mathbf{g}^* - \mathbf{x}_i^t) \cdot \mathrm{pf}_i \tag{2.16}$$

  where r is a random number in $[0,1]$, and $\mathbf{g}^*$ denotes the best-known binary solution. This allows butterflies to explore promising regions of the search space, thereby promoting global exploration.

- **Local Search (Exploitation)**
  If global search is not selected (i.e., $r \geq p$), a butterfly performs local search by interacting with two randomly chosen butterflies j and k. The local movement is calculated as:

  $$\mathbf{F}_i^{t+1} = (r \cdot \mathbf{x}_j^t - \mathbf{x}_k^t) \cdot \mathrm{pf}_i \tag{2.17}$$

  This enables focused exploration around similar solutions, enhancing local exploitation and aiding the algorithm in escaping local optima.

Since these update rules produce continuous intermediate values, BBOA applies a transfer function, typically the sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2.18}$$

to map them to probabilities. These are then binarized using:

$$x_i^{t+1} = \begin{cases} 1 & \text{if } S(x) > r \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

where r is a uniformly generated random number in $[0,1]$.

This binary adaptation enables BBOA to effectively search for optimal feature subsets, thereby reducing dataset dimensionality while maximizing classification performance.

By balancing global exploration and local exploitation through probabilistic switching, BBOA achieves a substantial trade-off between intensification and diversification, making it particularly effective for high-dimensional binary optimization tasks.

2.7 Interpretability and Explainability

AI models have revolutionized various fields by delivering exceptional performance across tasks such as IR, NLP, and TSF. However, a significant limitation persists: their inherent black-box nature. These models, composed of deep, highly parameterized neural architectures, often produce highly accurate outputs while offering little to no transparency into how those results are derived. This opacity makes it challenging for users and stakeholders to understand, validate, or trust the model's decision-making process [189]. This issue becomes particularly critical in sensitive domains such as finance [198, 173] and healthcare [271, 181], where decisions informed by ML models can have significant ethical, legal, or economic consequences. In both sectors, model transparency is not just a desirable feature; it is a fundamental requirement for building trust, ensuring accountability, and complying with legal and ethical standards. To address these concerns, the concepts of **explainability** and **interpretability** have gained increasing importance in the development and deployment of AI systems. **Explainability** refers to the ability to provide understandable and human-readable justifications for a model's predictions [189]. On the other hand, **Interpretability** denotes the extent to which a human can comprehend the internal mechanics of a model, such as understanding the weights, activations, or structure that drive the prediction. Together, these concepts aim to open the black box of DL, enabling traceability of how and why decisions are made, ultimately enabling safer and more responsible AI adoption in high-impact areas [189].To operationalize these concepts, the field of XAI has developed a range of techniques to enhance model transparency. Notable methods include LIME [184], which approximates complex models locally with simpler interpretable models. CAM [279], which highlights important regions in input data (e.g., image pixels) that influence predictions, SHAP [140], which quantifies feature contributions based on cooperative game theory; SHAP computes Shapley values to quantify the marginal contribution of each feature to a model's prediction, thereby enabling both local and global interpretability. It satisfies key properties, including local accuracy, missingness, and consistency, making it effective for revealing how individual features influence predictions.

These XAI techniques have been successfully applied across a wide range of applications and have become essential tools in making AI systems more interpretable, trustworthy, and transparent.

2.8 Conclusion

In this chapter, we reviewed the fundamental methodologies that support this research, covering classical ML, DL, Transformer architectures, feature extraction techniques, and XAI. The discussion began with the role of these methods in trend analysis, which is central to understanding dynamic behaviors in complex systems. Building on this foundation, we highlighted techniques directly relevant to the two main tasks examined in this work: link prediction and time series forecasting, with a specific focus on applications to international trade networks and trade flow dynamics. The concepts presented here establish the theoretical basis for the models and contributions developed in the subsequent chapters.

Chapter 3

Efficient Link Prediction in Social Networks Using a BBOA-Inspired Heuristic and Similarity-Based Machine Learning

3.1 Introduction

Social networks are vital for enabling user interactions and information exchange, in which users and their relationships form nodes and links in the network. A key challenge in analyzing these networks is link prediction—the task of forecasting potential connections based on existing patterns, which also involves analyzing trends to understand how relationships evolve. With the rise of large-scale social platforms and the growing complexity of data, there is a strong trend toward combining traditional similarity measures with advanced ML and feature selection techniques to improve predictive accuracy. Applications range from recommendation systems to community detection and social influence analysis. This chapter explores a hybrid approach that combines similarity-based and supervised learning methods, enhanced by Binary Butterfly Optimization for feature selection and SHAP for model interpretability, to address current demands for both accuracy and explainability. Methodology, datasets, and experimental analysis of various models on real-world networks are also presented.

3.2 Methodology

This section outlines the main stages of our research methodology. We began by selecting two social media datasets, then preprocessed the data, extracted features, and selected the most relevant attributes. Using these selected features, we finally split the data into training and testing sets and trained several ML models.

3.2.1 Datasets

Experiments are conducted using two real-world datasets: Facebook [117] and Email-Eu-Core [260]. The Facebook dataset comprises user profiles and their ego networks, including friend circles. In contrast, the Email-Eu-Core-Eu-core dataset represents internal Email-Eu-Core communications among members of a European institution, in which edges indicate at least one exchanged message. Detailed statistics for the datasets are presented in Table 3.1.

Table 3.1: The detailed statistics of Facebook & Email-Eu-Core dataset

Datasets	**Nodes**	**Edges**
Email-Eu-Core	1.005	25.571
Facebook	4.039	88.234

3.2.2 Data Representation and Feature Extraction

Algorithm 1 explains the detailed process of data presentation and feature extraction. The first part of the algorithm, from line 3 to line 18, outlines the procedure for converting the raw input dataset X into a graph representation G. Specifically, the algorithm starts by extracting the node set V and edge set E from the dataset (lines 4–5). These are used to initialize the graph $G = (V, E)$ (line 6). The number of observed edges is recorded (line 7), and an equal number of random non-edges are generated to serve as negative samples (line 8) to avoid class imbalance bias. The loop from lines 9 to 11 labels all existing edges $(u, v) \in E$ with a class label of 1, signifying that a true link exists. Subsequently, the loop from lines 12 to 17 examines the generated non-edge pairs. For each pair (u, v), if it does not already exist in the edge set E, it is added to the graph as a non-edge (line 14) and assigned a label of 0 (line 15), representing the absence of a relationship. This prepares a binary-labeled graph dataset suitable for training and evaluation in link prediction tasks.

The second part of the algorithm, from line 20 to line 33, is encapsulated in the CalculateFeatures(G) function. This stage computes structural similarity metrics for the graph's nodes and edges. For each node $u \in G$, the clustering coefficient is calculated (lines 21–23), which quantifies the tendency of u's neighbors to form tightly knit groups. The loop from lines 24 to 30 then computes a set of edge-based similarity measures for each edge pair (u, v). These include well-established network metrics such as: Common Neighbors, Adamic-Adar Index, Jaccard Coefficient, Preferential Attachment, Salton Index, Sørensen Index, Hub Promoted Index, Resource Allocation Index, Random Walk Index, Leicht-Holme-Newman Index (LHNI), and the Matrix Forest Index (MFI), which is derived from matrix-forest theory. Finally, in line 31, all computed features are normalized using MinMaxScaler to ensure a consistent scale across metrics, enabling fair comparisons and improved convergence during model training. The fully annotated graph G, now enriched with structural features and binary labels, is returned for use in downstream ML tasks (line 32).

3.2.3 Feature Selection

Our simplified version of the Binary Butterfly Optimization Algorithm (BBOA) retains its population-based search and mutation mechanisms. Still, it removes components such as fragrance modeling, global–local switching, and probabilistic migration rules. Instead, it uses a mutation-driven local search guided by the best-performing solution in each generation. This adaptation reduces complexity while maintaining effective feature selection exploration in link prediction tasks. Algorithm 2 presents the inspired BBOA approach to feature selection. Lines 1-6 define the algorithm's inputs and outputs. Specifically,

Algorithm 1 Feature Extraction

```
Input: X                                              ▷ Dataset containing pairs of nodes
Output: Graph G with calculated features
procedure CreateGraph(X)
    V ← Extract nodes from X
    E ← Extract edges from X
    G ← (V, E)                                        ▷ Initialize graph with nodes and edges
    |E_target| ← |E|
    E_random ← Generate random non-edges of size |E_target|
    for each edge (u, v) ∈ E do
        Label (u, v) in X as 1                        ▷ Edge exists
    end for
    for each pair (u, v) ∈ E_random do
        if (u, v) ∉ E then
            Add (u, v) to G as non-edge
            Label (u, v) in X as 0                    ▷ Random non-edge
        end if
    end for
    return G
end procedure
function CalculateFeatures(G)
    for each node u in G do
        ClusteringCoeff(u) ← ClusteringCoefficient(G, u)
    end for
    for each edge (u, v) in G do
        CommonNeighbors(u, v) ← CommonNeighbors(G, u, v)
        AdamicAdar(u, v) ← AdamicAdar(G, u, v)
        JaccardCoeff(u, v) ← JaccardCoefficient(G, u, v)
        PreferentialAttach(u, v) ← PreferentialAttach(G, u, v)
        Other similarity measures
    end for
    Normalize features using MinMaxScaler
    return G
end function
```

the training and testing datasets (X_train, y_train, X_test, y_test), the number of optimization iterations (N), and the population size (S) are provided as inputs. The output is the optimal subset of features, denoted as best_features. In line 8, the total number of features (F) is determined from the training data (X_train). Line 9 initializes the population (P), which consists of S binary vectors, each of length F. These vectors represent candidate feature subsets, where a value of 1 at a given index indicates the inclusion of the corresponding feature, and 0 denotes its exclusion.

The initialization phase (Lines 7–10) initializes the search space for the optimization process by randomly generating binary solutions.

The central part of the algorithm is the optimization loop (starting at Line 11), which runs for a predefined number of iterations (N).

Within this loop, the fitness of each individual in the population is evaluated (Lines 13–15). The fitness is determined by the performance of the objective function (see Algorithm 3), which uses kNN as a classifier. We selected it as the fitness evaluator because it is non-parametric, fast to train, and sensitive to feature relevance, making it a suitable proxy for assessing feature subset quality during optimization. The classifier is trained and tested using only the features selected (i.e., marked with a 1) in the individual's binary vector. The fitness metric used is the classification accuracy on the test dataset (X_test). From the evaluated population, the individual with the highest fitness score is selected as p_best (Line 17), representing the current best subset of features. A mutation operation is then applied to p_best (Lines 18–19) by randomly flipping one bit in its binary vector, producing a new individual p_mut. The fitness of p_mut is then computed (Line 19). If p_mut outperforms p_best, it replaces p_best in the population (Lines 20–21).

This evaluation-and-selective-mutation process continues at each iteration (Lines 12–24), allowing the algorithm to exploit the best solutions while exploring new potential feature combinations via mutation.

Once the specified number of iterations (N) is completed, the algorithm returns best_features (Line 26), which corresponds to the individual with the highest observed fitness throughout the process. This feature subset is expected to yield the best classification performance and is selected as the optimal set. The optimal features selected for the Email-Eu-Core dataset include, Clustering_Coefficient, Preferential_Attachment, Leicht–Holme–Newman Index (LHNI_Index), and Matrix Forest Index (MFI_Index) In contrast, the Facebook dataset benefits from a broader feature set, including Common_Neighbors, Adamic_Adar, Preferential_Attachment, Resource_Allocation_Index, Leicht–Holme–Newman Index (LHNI_Index), and Matrix Forest Index (MFI_Index). These features achieved the highest classification performance during feature selection.

3.2.4 Data Split

The dataset is divided into training and testing subsets, with 80% allocated to training and 20% to testing. This maximizes model performance and obtains the best possible results.

3.2.5 Training Model

After selecting the most relevant features using BBOA, a range of supervised ML models are trained. These included **SVC** [53], **DT** [64], **ANN** [187], **RF** [83], **kNN** [55] , **LR**[63],

Algorithm 2 Inspired Binary Butterfly Optimization for Feature Selection

Inputs:
    $X_{\text{train}}, y_{\text{train}}$: Training data features and labels
    $X_{\text{test}}, y_{\text{test}}$: Testing data features and labels
    N: Number of iterations
    S: Population size
Output: best_features: Optimal set of features
procedure INITIALIZE
    $F \leftarrow$ number of features in X_{train}
    $P \leftarrow$ population of S binary vectors of length F
end procedure
procedure OPTIMIZATIONLOOP
    for $n = 1$ to N **do**
        for each p_i in P **do**
            $\text{fit}(p_i) \leftarrow \text{OBJECTIVEFUNCTION}(p_i, X_{\text{train}}, y_{\text{train}},$
                $X_{\text{test}}, y_{\text{test}})$
        end for
        $p_{\text{best}} \leftarrow$ butterfly in P with highest fitness
        $p_{\text{mut}} \leftarrow$ mutate p_{best} by flipping one random bit
        Compute $\text{fit}(p_{\text{mut}})$
        if $\text{fit}(p_{\text{mut}}) > \text{fit}(p_{\text{best}})$ **then**
            Replace p_{best} in P with p_{mut}
        end if
        Update best_features if current best fitness improves
    end for
end procedure
return best_features

Algorithm 3 Objective_Function

function OBJECTIVE_FUNCTION(p_i, X_{train}, y_{train}, X_{test}, y_{test})
    Train a kNN classifier on $X_{\text{train}}[p_i], y_{\text{train}}$
    Evaluate accuracy on $X_{\text{test}}[p_i], y_{\text{test}}$
    Store accuracy as fitness $\text{fit}(p_i)$
    return $\text{fit}(p_i)$
end function

and **GNB** [188].

In addition to these traditional models, a CNN is implemented to explore deep learning approaches for the classification task. The CNN architecture is designed as follows:

- **Conv1D Layer:** Consists of 32 filters with a kernel size of 3 and ReLU activation, which captures key patterns in sequential data.
- **MaxPooling1D Layer:** Reduces the dimensionality of the feature maps, preserving important features while decreasing computational cost.
- **Flatten Layer:** Transforms the pooled feature maps into a one-dimensional array to serve as input to the dense layers.
- **Dense Layers with Dropout:** Two fully connected layers with 64 and 32 neurons respectively, each followed by a dropout layer with a dropout rate of 0.5 to reduce overfitting and improve generalization.
- **Output Layer:** A single neuron with a sigmoid activation function that outputs the probability of a connection between a given pair of nodes.

The CNN model is compiled with the Adam optimizer and binary cross-entropy loss, both of which are well-suited for binary classification.

3.3 Results and Evalution

3.3.1 Setups

The models are executed on a workstation with an Intel Core i9-13900KF CPU, 128 GB of RAM, and an Nvidia GeForce RTX 3060 12G graphics card. The experiments were conducted on a Windows 11 system using Anaconda Jupyter Notebook, Python 3.10, and essential libraries, including scikit-learn, TensorFlow, NetworkX, and SHAP.

3.3.2 Result

Table 3.2 presents the performance of various ML algorithms on two distinct datasets: Email-Eu-Core and Facebook, evaluated using four key classification metrics: accuracy, precision, recall, and F1-score.

On the Email-Eu-Core dataset, the RF and CNN models demonstrated superior performance, each attaining an accuracy and F1-score of (0.854). In contrast, the GNB classifier, although achieving the highest precision (0.926), exhibited a substantially lower recall (0.393), which resulted in a diminished F1-score of (0.541). This discrepancy suggests that GN was highly conservative in its predictions, favoring precision at the cost of failing to identify a significant proportion of true positives. The ANN model performed suboptimally, with remarkably low precision (0.243) and an F1-score of only (0.331), indicating challenges in generalizing effectively on this dataset—possibly due to data sparsity, overfitting, or insufficient model complexity. In comparison, performance across all models on the Facebook dataset was substantially higher. The CNN model achieved the highest accuracy (0.978), followed closely by kNN, ANN, and RF, all with accuracy values between (0.96) and (0.97). Notably, all models achieved F1-scores above (0.92), demonstrating effective generalization on this dataset. The improved results are attributed to the more structured and less noisy nature, as well as the larger size, of the Facebook dataset, which may allow models to learn distinguishing features more effectively.

Additionally, classical models such as LR and SVC performed competitively on the Facebook dataset, both attaining (0.949) accuracy and F1-scores, showing that even simpler models can achieve high performance given clean and linearly separable data. Unlike its performance on the Email-Eu-Core dataset, GNB also performed reasonably well here, with an F1-score of0.921%.

In summary, the CNN model consistently performed among the best across both datasets, highlighting its strength in capturing local patterns in the data. However, the Facebook dataset appears more amenable to classification, with all models showing improved performance compared to the Email-Eu-Core dataset.

Table 3.2: Performance of ML algorithms on two datasets

Dataset	**Algorithm**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Email-Eu-Core	SVC	0.792	0.848	0.701	0.767
	LR	0.832	0.841	0.799	0.824
	DT	0.800	0.801	0.800	0.800
	GN	0.681	0.926	0.393	0.541
	kNN	0.808	0.816	0.796	0.809
	RF	**0.852**	0.852	0.852	0.852
	ANN	0.799	0.243	0.500	0.331
	CNN	**0.854**	**0.859**	0.841	0.852
Facebook	SVC	0.953	0.977	0.928	0.949
	LR	0.949	**0.989**	0.912	0.952
	DT	0.951	0.951	0.951	0.951
	GNB	0.933	0.979	0.867	0.921
	kNN	0.973	0.966	0.964	0.964
	RF	0.964	0.964	0.964	0.964
	ANN	0.969	0.964	0.945	0.957
	CNN	**0.978**	0.972	0.963	0.963

3.3.3 Interpretation the Model

The results in Table 3.2 show that the CNN model achieved the highest accuracy on both the Email-Eu-Core and Facebook datasets. To further interpret the CNN model's decision-making process, SHAP is employed, a global model-agnostic interpretability method that quantifies the contribution of each input feature to the model's predictions.

Figures 3.1 and 3.2 illustrate bar plots of the mean absolute SHAP values for each feature, providing insight into their overall contribution to the CNN model's predictions on the Email-Eu-Core and Facebook datasets, respectively.

In Figure 3.1, which corresponds to the Email-Eu-Core dataset, the Preferential_Attachment feature exhibits the highest mean SHAP value (0.28), indicating it has the most significant influence on the model's output. This suggests that the CNN heavily relies on this structural feature when making classification decisions. The Clustering_Coefficient follows with a mean SHAP value of approximately 0.03, showing a moderate impact. In contrast, features such as LHNI_Index and MFI_Index display substantially lower SHAP values (0.01), implying a relatively minor role in shaping the predictions.

These findings highlight the importance of global structural properties—particularly Preferential Attachment—in guiding the CNN's classification process for link prediction tasks.

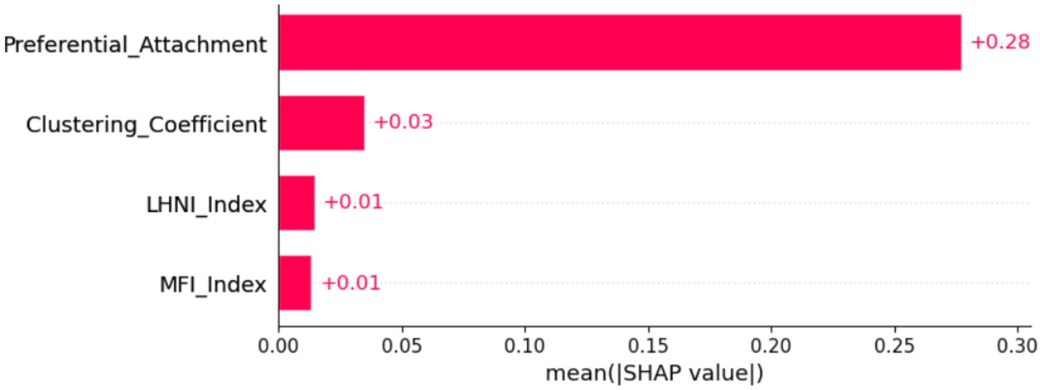

Figure 3.1: The bar plot of the mean SHAP of the Email-Eu-Core dataset

Figure 3.2 presents a bar plot of the mean absolute SHAP values across features, highlighting their relative importance in the CNN model's predictions for the Facebook dataset. The Common_Neighbors feature exhibits the highest mean SHAP value (0.21), suggesting it has the most decisive influence on the model's output. This is followed by Adamic_Adar and Resource_Allocation_Index, both with mean SHAP values of approximately 0.14, indicating they also contribute significantly to the model's predictions. In contrast, Preferential_Attachment shows a substantially lower mean SHAP value (0.04), while LHNI_Index and MFI_Index have mean SHAP values close to zero. These findings suggest that local similarity-based features play a more prominent role in the CNN's classification performance on the Facebook dataset than global structural metrics.

These figures, 3.3 and 3.4, present a detailed SHAP summary plot for the CNN model on the Email-Eu-Core and Facebook datasets. The x-axis represents the SHAP value, indicating the impact of each feature on the model's output. Positive SHAP values show features pushing the prediction higher, while negative SHAP values indicate features pushing the prediction lower. The color of each dot represents the feature value, with blue indicating low values and red indicating high values. Figure 3.3 illustrates the distribution of SHAP values for each feature in the CNN model applied to the Email-Eu-Core dataset, offering insights into both the direction and magnitude of their influence on the model's predictions. The Preferential_Attachment feature exhibits the broadest range of SHAP values, spanning from approximately($-0.4 to + 0.6$). This indicates that it can strongly influence the prediction in either direction, depending on the input value. Notably, higher feature values (depicted in red) tend to increase the predicted probability, whereas lower values (in blue) are associated with a decrease in the prediction.

The Clustering_Coefficient also shows a moderate spread in SHAP values, though most are centered around zero. This suggests that, in many instances, it has a minimal effect on the model's output. However, in specific cases, both high and low values can contribute to either a positive or negative shift in the prediction.

In contrast, LHNI_Index and MFI_Index exhibit minimal overall impact. Their

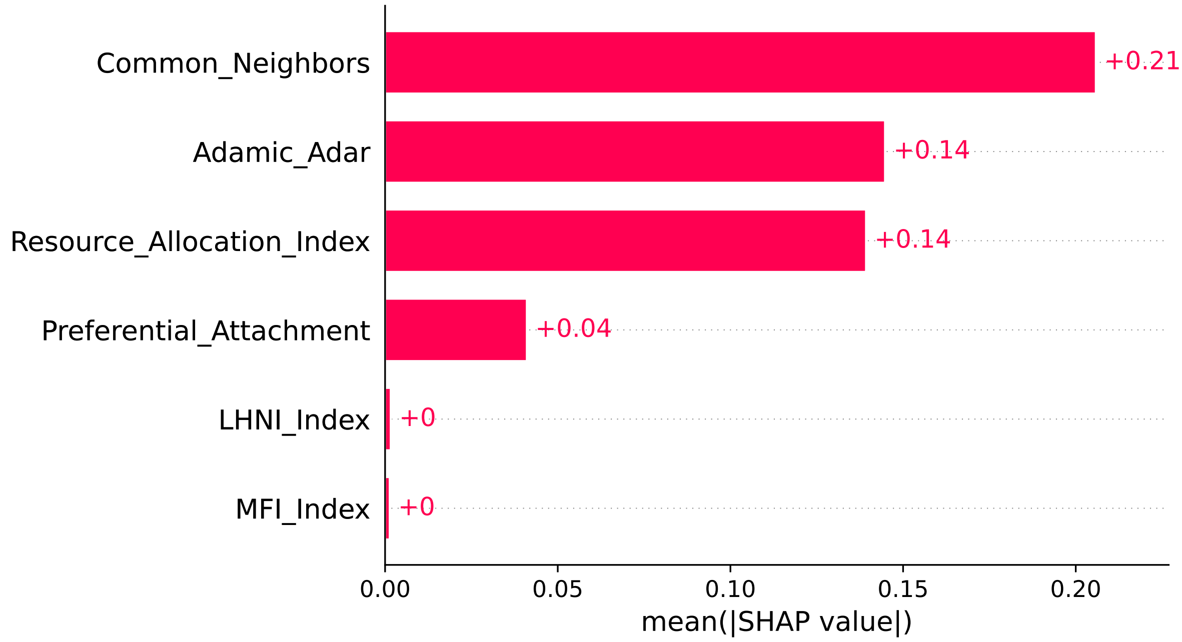

Figure 3.2: The bar plot of the mean SHAP of the Facebook dataset

SHAP values are concentrated near zero, indicating limited influence on the model's decisions, although a few high-value instances exhibit a slight positive effect. These observations reinforce the dominant role of Preferential_Attachment and Clustering_Coefficient in the CNN's performance on the Email-Eu-Core dataset, particularly in capturing structural patterns within the underlying graph.

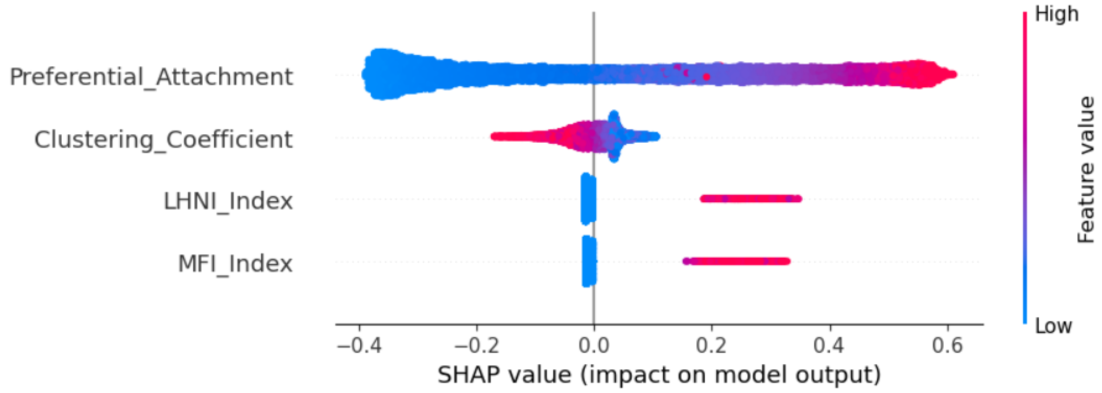

Figure 3.3: Summary plot of the CNN model for the Email-Eu-Core dataset.

Figure 3.4 shows that the Common_Neighbors feature has a varied impact with points scattered across the SHAP value range, indicating that it can have positive and negative effects on the prediction. High values (red) tend to increase the prediction, while low values (blue) tend to decrease it.Adamic_Adar and Resource_Allocation_Index features also display a broad distribution of SHAP values, reflecting their complex influence on the model's output. They have a mix of positive and negative impacts depending on their value. The Preferential_Attachment feature has a wide range of SHAP values, from approximately (-0.4 to 0.4), indicating it can both significantly increase and decrease the model's predictions depending on its value. Higher values of Preferential_Attachment (in red) generally increase the prediction, while lower values (in blue) tend to decrease it.LHNI_Index and MFI_Index features have a minor impact on the prediction. The SHAP values are mainly close to zero, indicating minimal influence on the model's output. However, in a few cases, high values (red) can slightly increase the prediction.

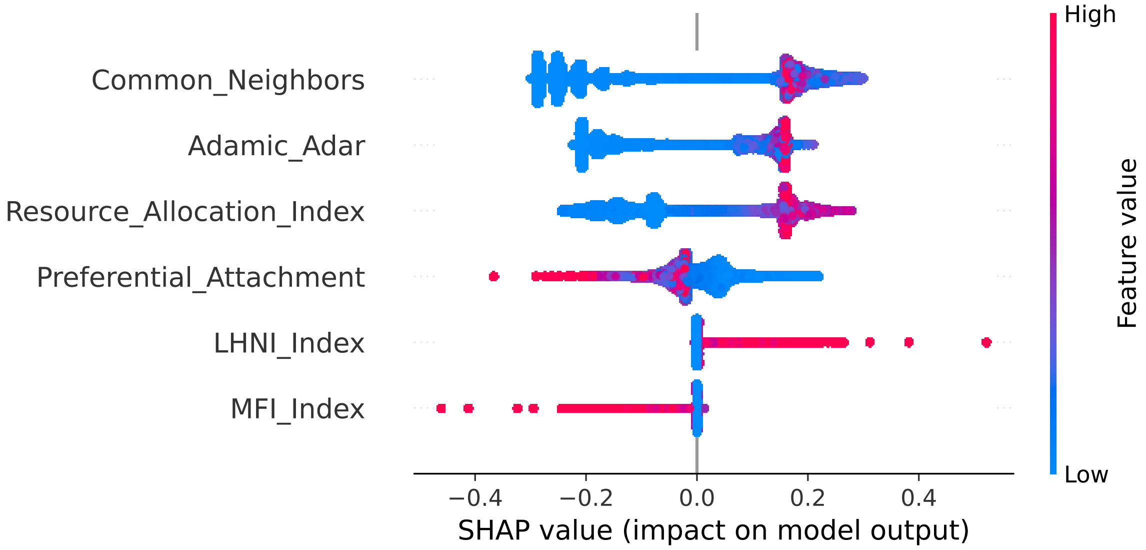

Figure 3.4: Summary plot of the CNN model for the Facebook dataset.

3.4 Conclusion

This chapter presented a hybrid link prediction framework that integrates similarity measures, supervised ML models, and an enhanced Binary Butterfly Optimization–inspired feature selection strategy. The proposed approach improves prediction accuracy, reduces feature redundancy and computational complexity, and SHAP-based explainability supports transparent interpretation of model outputs. Experimental evaluations on real-world social network datasets demonstrate the effectiveness of the framework, with the CNN model achieving the best performance, reaching 98% accuracy on the Facebook dataset and 85% on the Email-Eu-Core dataset.

Chapter 4

High-Accuracy Prediction of International Raw Material Trade Flows Using Temporal Fusion Transformer

4.1 Introduction

In this chapter, we present our contribution to international trade forecasting by leveraging the TFT model. Our approach aims to accurately predict raw material trade flows between countries. We construct a rich, hybrid dataset encompassing trade, economic, and geographic features by integrating data from multiple trusted sources, including the UN Comtrade database, the CEPII Gravity dataset, and the World Bank. The application of TFT enables us to capture complex temporal dependencies while offering model interpretability, which is crucial for identifying key influencing factors. This chapter details the methodology, experimental setup, results, and implications of our model, highlighting its robustness under volatile trade conditions, including those observed during the COVID-19 pandemic and geopolitical disruptions such as the Ukraine war.

4.2 Methodology

This section outlines our research contribution through three key components, as shown in Figure 4.1: Data integration and pre-processing, model training, and evaluation. We start by describing the data used and the steps to prepare it for analysis. We then discuss the model's training process and conclude with an evaluation of its performance.

4.2.1 Data Integration and Pre-processing

The first step in our approach was to collect and clean the necessary datasets. We integrated data from the UN Comtrade database. (2023) `https://comtradeplus.un.org/`, a detailed source of international trade information, the CEPII gravity data CEPII Gravity Database, Version: CEPII,2022-2, `http://www.cepii.fr/`, and the World Bank Open Data World Bank Open Data, `https://databank.worldbank.org/`, which offers a wide range of global economic and social statistics, to form a comprehensive dataset.

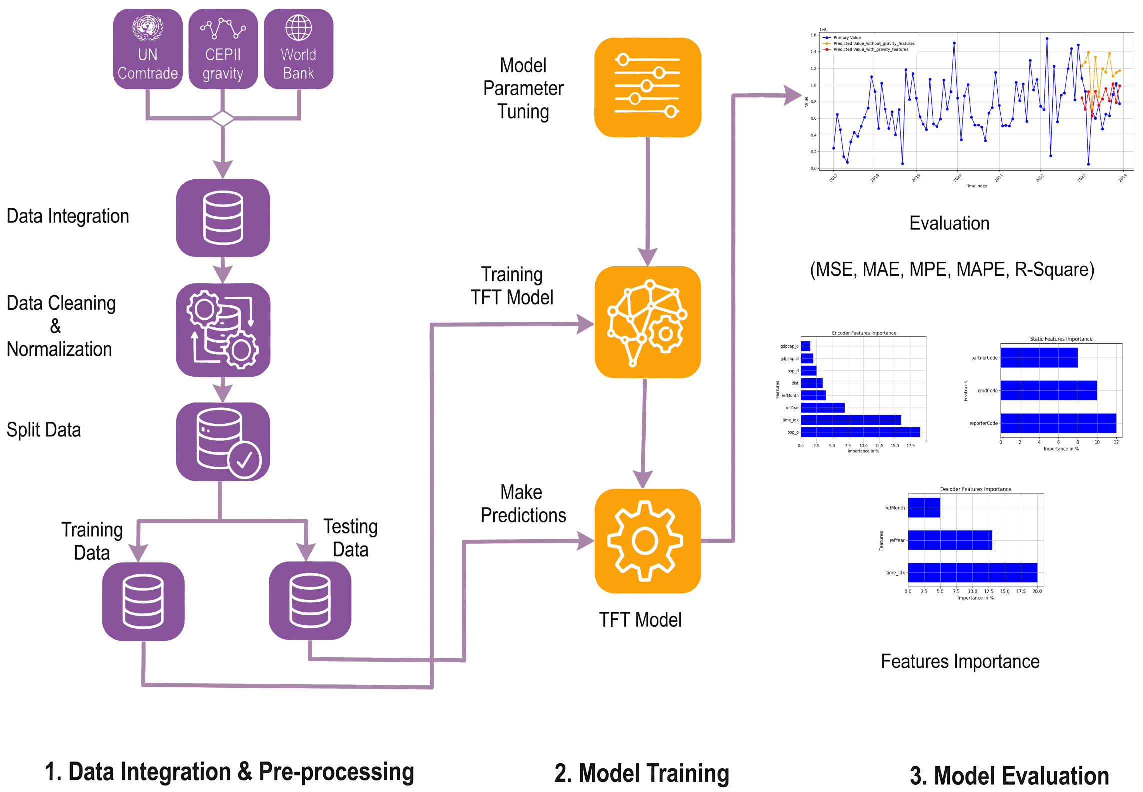

Figure 4.1: Research framework architecture.

We preprocessed these datasets to address inconsistencies and impute missing values, ensuring that our analysis is accurate.

The United Nations Comtrade Database is widely acknowledged as the most authoritative global data source for international trade in goods. It is an original, highly standardized database for monitoring international goods trade, covering bilateral trade among 250 countries and territories. The database categorizes over 5,000 products at the 6-digit HS product code level and comprehensively covers global trade. Each database entry represents a well-structured bilateral trade flow record, including the reporting country, partner country, traded product, trade value, and quantity. Official national entities collect the data, ensuring accuracy and reliability.

The database provides an extensive record of global trade flows, capturing import and export details for a wide array of goods and commodities. This dataset provides detailed information on the commodities exchanged between countries. It includes details on the trading countries, product codes, quantities, and values.

For this study, we selected a variety of raw materials for our experiments. This selection is based on [97], which developed trade codes for non-food, non-fuel raw materials and their products at the HS6 level. The chosen materials include those critical to the EU's economy and supply chain security, encompassing abiotic, biotic, and synthetic categories. Abiotic raw materials include metals, industrial minerals, and construction minerals, such as aluminum, cobalt, lithium, magnesium, nickel, and rare earth elements. These are vital for key industries and technologies and carry a high supply risk due to the EU's import dependency. Biotic raw materials include natural substances such as natural rubber, cork, and industrial roundwood, which are crucial to various manufacturing sectors. Additionally, synthetic materials, such as artificial graphite, rubber, and carbon fibers, are examined for their technological importance. In total, 282 raw materials at the HS6 level were studied.

CEPII Gravity Database is a widely employed and comprehensive resource for estimating gravity models and exploring bilateral trade flows. This dataset encompasses a range of features that facilitate a deeper understanding of international trade dynamics. The dataset includes key macroeconomic indicators, including GDP, GDP per capita, and population. Additionally, the dataset consists of geographic features, such as bilateral distances and shared borders, as well as other elements, such as shared languages and historical colonial ties. The CEPII Gravity Dataset spans the period from 1948 to 2021 and provides extensive geographic coverage, with data for 252 countries and territories.

The CEPII Gravity Dataset collects data from diverse sources. The World Bank's World Development Indicators (WDI), the International Monetary Fund (IMF), and the World Trade Organization (WTO) contribute to the dataset's richness. At the same time, CEPII's databases directly source a substantial portion of the data.

The World Bank Open Data offers a wide range of global economic and social statistics, including crucial indicators on the Gross Domestic Product (GDP) and population. It also provides extensive data on critical areas, including poverty, education, and gender.

Combining the UN Comtrade dataset with gravity and World Bank features enhances its depth and scope. This hybrid dataset, comprising traded products and macroeconomic and geographic variables, serves as the foundation for our trade flow prediction analysis.

4.2.1.1 Data Integration

Harmonizing diverse datasets based on common elements, such as commodity and country codes, is crucial for integrating gravity features with UN Comtrade data. This alignment creates a unified framework that combines information on traded goods with broader gravity-model features. The resulting integrated dataset underpins our analysis, enabling a comprehensive examination of the interplay between macroeconomic indicators and commodity-specific details in international trade dynamics. The resulting hybrid dataset includes a range of factors, as summarized in Table 4.1.

Table 4.1: Factors in the hybrid dataset

Factor	Description
Trade value	The real value of goods exchanged between two nations, expressed in US Dollars (USD). Derived from the UN Comtrade dataset.
Product code	Harmonised System (HS) codes representing specific traded goods, facilitating an in-depth examination of trade connections.
Population	Total number of residents in each trading nation are essential for gauging market size and potential import demand.
GDP per capita	Economic performance per individual are calculated by dividing the GDP by the total population. Provides insights into the average economic well-being and purchasing power of citizens.
Distance	Geographical distance between trading countries.

4.2.1.2 Data Cleaning

The data cleaning process involved applying various criteria to ensure the dataset's accuracy and relevance. Focusing on the UN Comtrade monthly data from 2017 to 2023 under the HS 2012 revision, we curated a subset by retaining export data (flowCode X) associated with all customs procedure codes (CustomsCode C00), all modes of transport (MotCode 0), and all last known destinations (Partner2Code 0).

To enrich the dataset, we integrated additional features from the World Bank dataset, including GDP and GDP per capita for exporting and importing countries, as well as the populations of the origin and destination countries. Furthermore, the geographical distance between countries (dist), sourced from the CEPII gravity dataset, was included to provide context for trade relationships.

We addressed data completeness by removing groups with missing timestamps, thereby ensuring a continuous, uninterrupted time series. Maintaining a constant time series ensures robust forecasting.

A crucial step in the data preprocessing was excluding specific HS codes that introduced significant volatility. Specifically, we removed HS code 271121 (Petroleum gases and other gaseous hydrocarbons; in a gaseous state, natural gas) and HS code 271111 (Petroleum gases and other gaseous hydrocarbons; liquefied, natural gas). This decision was motivated by substantial fluctuations in natural gas prices during the study period, which could distort the results and complicate forecasting.

The rationale for excluding these commodities is their high sensitivity to external market factors, including geopolitical events, supply chain disruptions, and seasonal demand variations. Natural gas prices exhibited pronounced spikes and troughs, resulting in extreme outliers that could overshadow more stable trends in other commodities. This exclusion improves generalization and forecasting stability.

To assess the impact of this exclusion, we compared the dataset with and without natural gas by analysing monthly averages over the study period. Figure 4.2, which includes natural gas data, shows significant volatility, with an average peak of approximately 5.0 million USD per HS code in 2022. This high level of fluctuation reflects the sensitivity of natural gas to external price shocks, contributing to significant variability that can hinder the accuracy of forecasting models.

In contrast, excluding natural gas presents a more stable trend. The values gradually rose from 2017, peaking just above 3.5 million USD per HS code in 2022, followed by a moderate decline in 2023. Removing natural gas data significantly reduces the impact of external market fluctuations, resulting in a remarkably stable dataset that is highly reliable for predictive modeling.

This comparison underscores the advantages of excluding natural gas data, leading to a more reliable and stable dataset. By minimizing the influence of extreme outliers, the exclusion enhances the overall accuracy and robustness of predictive models. After data cleaning and preparation, the final dataset is summarized in Table 4.2, which details the features and their corresponding measurements.

4.2.1.3 Data Normalization

This essential step ensures that numerical features are uniformly scaled, covering macroeconomic and product-specific factors. By applying MinMax scaling to the distance, population, GDP per capita, and GDP columns, we created a standardized and consistent framework for analysis, mitigating potential biases from differing feature scales.

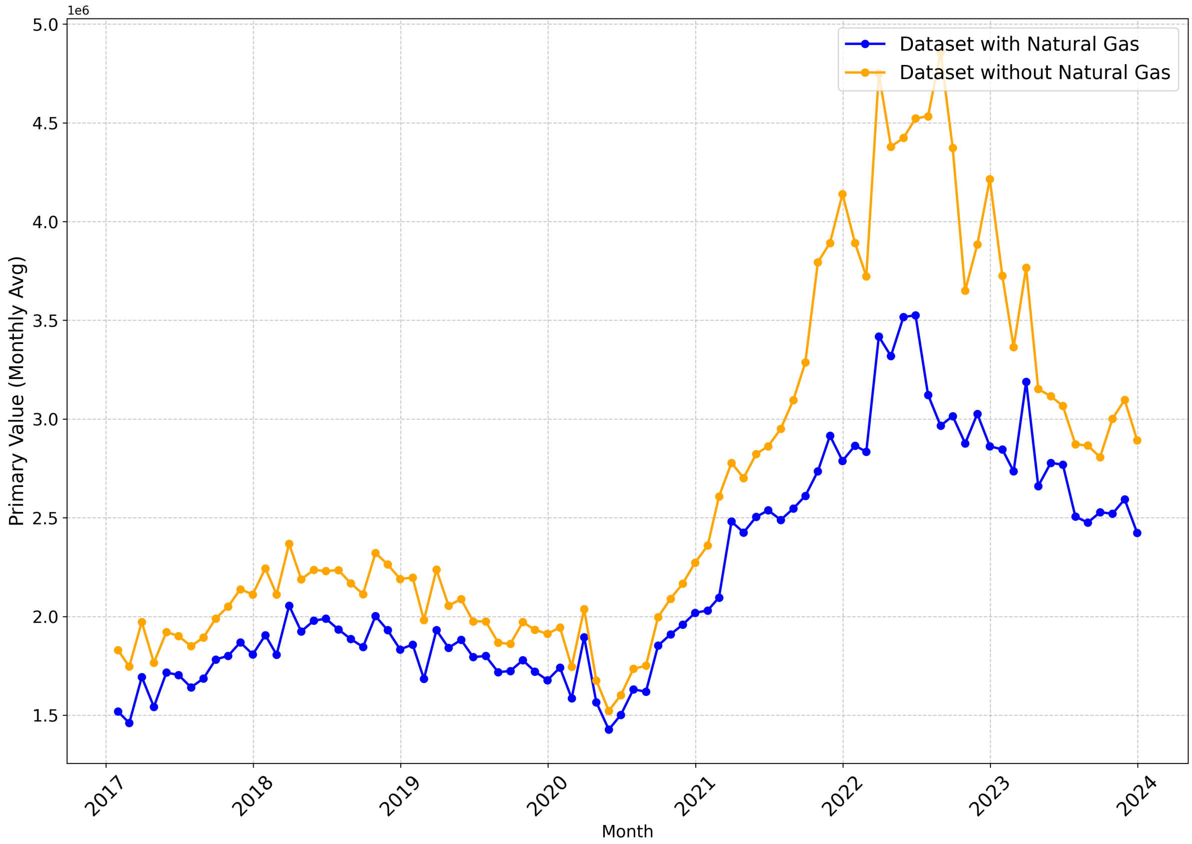

Figure 4.2: Comparison of the monthly average trade value per HS code, with and without natural gas, in USD (2017–2023).

4.2.1.4 Train-Test Split

To evaluate the model's performance, we employed a temporal split for the dataset using the Python module "TimeSeriesDataSet". Data from January 2017 to December 2022 were used for training, while data from January 2023 to December 2023 were reserved for testing. Ensures realistic out-of-sample evaluation.

4.2.2 Model Training

This section details the parameters and inputs used to train the TFT model, highlighting the specific configurations and data preparation steps that contribute to its performance. As outlined in Table 4.3, we used random search and carefully selected and tuned various parameters to achieve optimal performance for our prediction tasks. Our experimentation evaluated multiple window lengths across different parameter settings, both with and without additional feature inputs. This approach enabled us to assess the impact of various configurations on model performance systematically.

To better understand the model's behavior, we explored different historical data windows, with encoder lengths of 6, 12, and 24 months, to predict trade values for 2023. By varying these window lengths, we observed how different time horizons influenced the model's accuracy and generalization. This exploration provided valuable insights, identifying the most effective configurations and demonstrating the model's robustness across diverse setups.

Table 4.2: Bilateral trade datasets

Parameter	Value
Period	January 2017 - December 2023
Frequency	Monthly
Reporter Code	44
Partner Code	133
Group (Reporter-Partner-Commodity)	13,823
Commodity Code	264
Size	1,159,536
Features	GDP per capita, Population, Distance

We also experimented with incorporating additional feature inputs, observing how the integration of gravity features affected model performance. Reporter, partner, and commodity codes were treated as static covariate variables, remaining constant throughout the prediction process. In contrast, dynamic variables such as GDP per capita, population, and distance were treated as gravity features, thereby introducing variability that the model learned to predict.

Hyperparameter optimization was conducted through extensive experimentation to ensure an optimal balance between model complexity and predictive performance. An iterative tuning process was employed to refine key architectural choices, enhancing the model's ability to capture temporal dependencies and complex patterns in the data. A learning rate of 0.0003 was selected to maintain stability while ensuring effective convergence. The hidden layer size was tested at 128 and 64, and attention head sizes at 64 and 32 were explored to assess their impact on capturing intricate dependencies. Larger attention heads improved the model's ability to extract meaningful relationships but increased the risk of overfitting, whereas smaller attention heads reduced the risk of overfitting but could lead to underfitting.

To further enhance temporal feature extraction, Long Short-Term Memory (LSTM) layers were set to 2 and 4, enabling the model to learn both short-term and long-term dependencies. Additionally, hidden continuous sizes of 32 and 64 were examined to evaluate their influence on performance. A dropout rate of 0.1 was used to mitigate overfitting, and training was conducted for 50 epochs with a fixed batch size of 128, balancing computational efficiency and generalization. Specifically, two model hypotheses were tested: Hyp1, where the hidden size was set to 128, LSTM layers to 4, attention head size to 64, and hidden continuous size to 64; and Hyp2, where the hidden size was reduced to 64, LSTM layers were set to 2, attention head size to 32, and hidden continuous size to 32. These configurations were designed to evaluate the effects of varying network depth, feature-extraction capacity, and attention mechanisms on predictive accuracy. The comparative analysis of these hypotheses provides insights into the trade-offs between model expressiveness and computational efficiency, guiding the selection of an optimal architecture for time-series forecasting.

One key feature of TFT is that it provides users with a confidence interval. This helps in understanding the range within which the value should lie with a given confidence level. This is usually done using quantiles, in which multiple outputs correspond to different quantile predictions. In our approach, we used seven output quantiles to provide a comprehensive understanding of the prediction range. To select the most accurate forecast, we evaluated the performance of each quantile using three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R). The median value (50th quantile) was chosen as the primary prediction because it strikes the best balance, minimizing MAE and MSE while maximizing R. This quantile-based approach enables the model to capture the distribution of possible outcomes, yielding more robust and informative forecasts. In scenarios such as international trade forecasting, this range is particularly useful, as it provides practical estimates for data-driven economic decision-making.

Table 4.3: Hyperparameters of the model

Parameter	Value
Past Target	Trade value
Known Variables	Year, Month
Unknown Variables	GDP per capita, Population, Distance
Static Covariates	Reporter Code, Partner Code, Commodity Code
Encoder Length	[6,12,24]
Decoder Length	12
Learning Rate	0.0003
Hidden Size	[64,128]
Attention Head Size	[32 ,64]
LSTM Layers	[2,4]
Dropout	0.10
Hidden Continuous Size	[32,64]
Epochs	50
Batch Size	128

4.2.3 Evaluation

This section provides an overview of the evaluation process for our model, detailing the experimental setups used for training and testing, the baseline models used for comparison, and the evaluation metrics used to assess the model's performance.

4.2.3.1 Setups

We run the models on an Apple Mac with an M2 Ultra 24-core CPU and 192GB of RAM, using Jupyter Notebook and Python 3.9, and leverage essential libraries such as Keras (version 2.3.1), PyTorch (2.3.1), PyTorch Forecasting (1.0.0), and PyTorch Lightning (2.2.1).

4.2.3.2 Baselines

We evaluate the prediction performance of the TFT model on trade flow by comparing it against Random Forest and Graph Attention Network. Both baseline models utilize the UN Comtrade dataset, ensuring a consistent basis for comparison.

- Random Forest: As proposed in [218], the model builds multiple decision trees during the training phase and provides the average prediction of all the individual trees. The authors highlighted that Random Forest outperformed traditional trade gravity approaches in predicting international trade. This model is recognized for its robustness to overfitting and its ability to handle large datasets with high dimensionality.
- Graph Attention Network: As proposed in [196], the model utilizes an attention mechanism to aggregate features from neighboring nodes in a graph. This model assigns varying importance to neighboring nodes, enabling more nuanced feature extraction than traditional graph convolutional networks.

Both models demonstrated good performance on the annual UN Comtrade dataset; in our comparison, we evaluate their performance on more granular monthly data. By comparing TFT with these baseline models, we aim to highlight the efficacy and advantages of our proposed approach to predicting trade values.

4.2.3.3 Features Importance

Built-in interpretability in the TFT model enables a clear understanding of the importance of features in its predictions. Through its architecture, the model facilitates the calculation and visualization of feature importance, thereby identifying the variables with the most significant impact on forecasted trade flows. The TFT model classifies feature importance into three distinct categories: Encoder features relate to past targets and covariates; decoder features focus on future covariates; and static features involve fixed covariates. The interpretation results demonstrate that each feature type plays a distinct role in shaping the model's outputs. Static features, such as reporter, partner, and commodity codes, provide a stable context for the model's predictions, grounding the analysis in fixed attributes of trade relationships. Encoder features capture historical context, including past economic indicators like GDP per capita, population, and trade distances, which inform the model's understanding of future trade flows. Meanwhile, decoder features emphasize forward-looking information, highlighting how future covariates, such as relative time indices, influence the model's forecasts. This interpretability not only enhances the model's transparency but also provides valuable insights into the key drivers of its predictions, facilitating more informed decision-making.

4.3 Results

Table 4.5 presents TFT's performance evaluation with and without gravity features (population, GDP per capita, and distance between countries) compared with its competitors, RF and GAT. The results indicate that the GAT model achieved the highest R-squared value of 0.93, suggesting that it explains the greatest portion of the variance in trade values. However, it also had the highest MAPE 32.7×10^3 and MAE 7.9×10^6, indicating significant sensitivity to outliers and larger average errors.

The TFT model with gravity features presents a well-balanced performance. With an R-squared of 0.85, it performs reasonably well in terms of explaining variance, though slightly lower than GAT. Its MAPE of 2.7×10^3 and MAE of 1.1×10^6 are significantly lower than those of GAT and RF, underscoring its effectiveness in minimizing relative and

Table 4.4: Model Performance Using Test Dataset in 2023 with Different Parameters

Hyperparameter	Windows	R-squared	MAPE	MPE	MSE	MAE
With Gravity Features						
Hyp1	(6,12)	0.83	3.1×10^3	-3.1×10^3	1.3×10^{14}	1.23×10^6
	(12,12)	0.84	3.1×10^3	-3.1×10^3	1.3×10^{14}	1.14×10^6
	(24,12)	**0.85**	2.7×10^3	-2.7×10^3	1.2×10^{14}	1.1×10^6
Hyp2	(6,12)	0.83	4.2×10^3	-4.2×10^3	1.3×10^{14}	1.2×10^6
	(12,12)	0.82	5.17×10^3	-5.2×10^3	1.4×10^{14}	1.4×10^6
	(24,12)	0.82	4.1×10^3	-4.1×10^3	1.4×10^{14}	1.3×10^6
Without Gravity Features						
Hyp1	(6,12)	0.81	3.6×10^3	-3.6×10^3	1.1×10^{14}	1.1×10^6
	(12,12)	0.84	2.8×10^3	-2.9×10^3	1.5×10^{14}	1.3×10^6
	(24,12)	**0.84**	2.7×10^3	-2.6×10^3	1.3×10^{14}	1.2×10^6
Hyp2	(6,12)	0.78	5.6×10^3	-5.6×10^3	2.2×10^{14}	1.6×10^6
	(12,12)	0.82	2.7×10^3	-2.7×10^3	1.2×10^{14}	1.1×10^6
	(24,12)	0,83	3.0×10^3	-2.9×10^3	1.4×10^{14}	1.1×10^6

average absolute errors. Similarly, the MSE for TFT with gravity features (1.2×10^{14}) is lower than RF but comparable to GAT, indicating moderately significant errors. The TFT model without gravity features performs similarly, with an R-squared of 0.84 and comparable MAPE 2.7×10^3 and MAE 1.2×10^6. This indicates that while gravity features contribute to performance improvements, the TFT model without them still provides competitive results, showing robustness in handling trade predictions.

The RF model, with an R-squared of only 0.83, lags behind the other models. It also has the highest MSE 1.4×10^{14}, MAPE 48.2×10^3, and MAE 1.4×10^6, indicating weaker performance, particularly in handling outliers and producing accurate predictions.

While TFT consistently outperforms RF and remains a competitive alternative to GAT, its performance in international trade prediction is influenced by multiple factors, including feature selection, historical window sizes, and hyperparameter configurations. A detailed evaluation was conducted to analyze how these elements shape the model's accuracy and robustness. The findings, as presented in Table 4.4, show how different configurations affect key evaluation metrics, including R-squared, MAPE, MPE, MSE, and MAE. In the following paragraphs, we analyze how gravity-related features, different input window sizes, and hyperparameter choices contribute to TFT's effectiveness in predicting international trade values.

- Effectiveness of Gravity-Related Features: One of the key findings of this study is the substantial improvement in predictive performance achieved by integrating gravity-related features (e.g., population, GDP per capita, and distance between countries) into the TFT model. These features enhance the ability to capture intricate spatial-temporal dependencies, leading to lower MAPE, MPE, MSE, and MAE values. The higher R-squared values observed in models with gravity-related inputs further confirm their positive impact on predictive reliability.

- Effectiveness of Different Window Sizes: The choice of window size plays a crucial role in determining the model's ability to extract meaningful temporal patterns. The results indicate that larger window sizes $(24, 12)$ consistently yield lower error values, reinforcing the hypothesis that longer input sequences enable the model to capture deeper temporal dependencies. Conversely, shorter window sizes $(6, 12)$ lead to higher prediction errors, likely due to their limited historical context, which limits the model's ability to generalize over time. Interestingly, the advantages of longer windows are even more pronounced when coupled with gravity-related features. The synergy between extended input sequences and enriched feature representations enables the model to achieve greater predictive accuracy.

- Effectiveness of Different Parameter Configurations: Beyond feature selection and temporal context, model performance is also influenced by hyperparameter choices. A comparison between Hyp1 and Hyp2 reveals that increased model complexity enhances predictive accuracy. Specifically, Hyp1, which utilizes a larger hidden size, deeper LSTM layers, and a higher attention head dimension, consistently outperforms Hyp2 across all evaluation metrics. This suggests that deeper architectures facilitate more effective feature extraction and representation learning. The superiority of Hyp1 is particularly evident when gravity features are included, as the larger model capacity allows for better utilization of enriched input representations. However, this comes at the cost of greater computational demands, which may pose challenges for real-time applications where efficiency is critical. In contrast, Hyp2, with its reduced complexity, offers a more resource-efficient alternative while still maintaining reasonable predictive performance.

Table 4.5: Model performance using test dataset in 2023

Model	R-squared	MAPE	MPE	MSE	MAE
TFT with gravity features	0.85	2.7×10^3	-2.7×10^3	1.2×10^{14}	1.1×10^6
TFT without gravity features	0.84	2.7×10^3	-2.6×10^3	1.3×10^{14}	1.2×10^6
RF	0.83	48.2×10^3	-48.2×10^3	1.4×10^{14}	1.4×10^6
GAT	**0.93**	32.7×10^3	-11.6×10^3	1.3×10^{14}	7.9×10^6

We analyze the TFT model results (with gravity features) to assess the significance of these features. This analysis provides insights into how these features influence the model's predictions, highlighting their contributions to understanding trade dynamics. Figure 4.3 shows the importance of the encoder features. This analysis highlights the dominant role of the population of the origin country (pop_o), which has the highest importance score and contributes approximately 20% to the model's predictive accuracy. The time index (time_idx), which represents the sequential time steps in the forecasting model, follows closely with a score of approximately 17%. Other worthy-of-mention features include the reference year (refYear, 4%) and the reference month (refMonth, 3%), which capture seasonal trends and temporal structure. Economic variables such as the GDP per capita of the origin (gdpcap_o, 1%) and destination (gdpcap_d, 1%) countries, as well as the distance between them (dist, 2%), also provide contextual economic information, though with lower impact. The destination country's population (pop_d, 2%) contributes moderately to the overall predictions.

Figure 5.1 shows the importance of the decoder feature. The time index, with an importance score of 20%, typically represents the absolute position in the time series and helps the model understand the sequence of events. Following the time index, the reference year 13% and reference month 5% also play important roles in the model's predictions. The reference year helps the model account for yearly trends and seasonal pat-

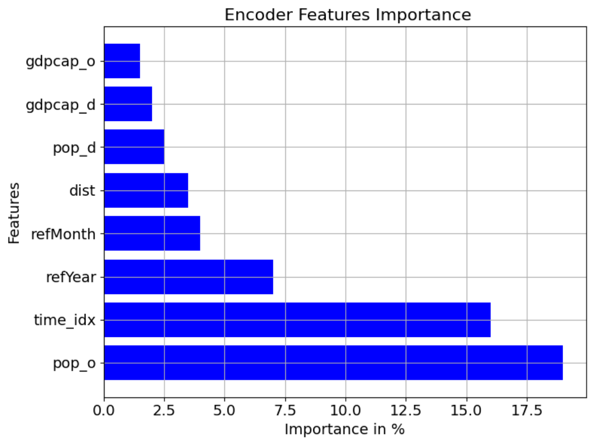

Figure 4.3: Encoder features importance

terns, while the reference month provides additional granularity to understand monthly variations.

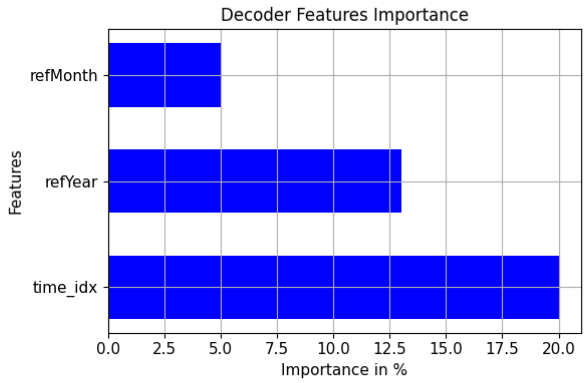

Figure 4.4: Decoder feature importance

Figure 4.5 illustrates the static feature importance. The reporter code is the most critical feature, with a strong influence on prediction accuracy and an importance score of 12%. In comparison, the partner code 8% and commodity code 10% have a moderate impact, indicating their roles are less substantial but still relevant.

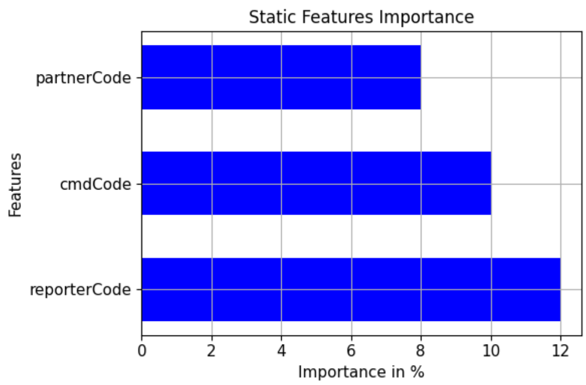

Figure 4.5: Static feature importance

Overall, this visual representation of feature importance provides key insights into how different variables contribute to the TFT model's predictive power.

4.4 Discussion

Regarding model performance, the TFT demonstrates a significant advantage over traditional machine learning models, such as Random Forest, and more sophisticated architectures, such as Graph Attention Networks. The TFT model consistently yields lower relative and absolute errors, as evidenced by superior evaluation metrics such as Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). This strong performance highlights its ability to effectively capture the intricate, nonlinear relationships governing international trade flows while adeptly managing multi-horizon temporal dependencies.

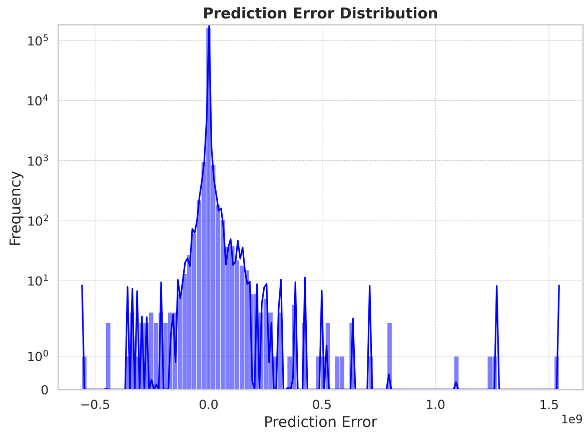

Figure 4.6: Analysis of Prediction Error Distribution

However, despite these strengths, certain limitations remain. As shown in Figure 4.6, the histogram of prediction error distribution follows a normal distribution, with most errors concentrated around zero, indicating high model accuracy. This suggests that the model performs well in predicting trade flows, as mistakes are symmetrically distributed with minimal bias. However, the presence of outliers—occasional large deviations from the mean—suggests that the model may struggle in specific cases, potentially due to extreme economic shocks, data sparsity, or unexpected trade fluctuations. These anomalies highlight the need for further refinement, such as improved feature engineering and the inclusion of additional macroeconomic indicators. Addressing these challenges could further strengthen the TFT model's robustness and reliability in forecasting international trade patterns.

Incorporating data from global disruptions such as the COVID-19 pandemic and the ongoing war in Ukraine further demonstrates the model's robustness. These events have caused unprecedented volatility and disruptions in international trade, significantly impacting supply chains, trade policies, and market demand. Despite this complexity, the TFT model has shown a strong capacity to adapt to these changes, effectively accounting for the sudden shifts and uncertainties in global trade patterns. As a result, it not only provides reliable forecasts during periods of relative stability but also offers valuable insights into managing crises and unexpected disruptions. This adaptability highlights the TFT model's potential as a tool for future crisis management, aiding policymakers and businesses in anticipating and navigating market fluctuations more effectively.

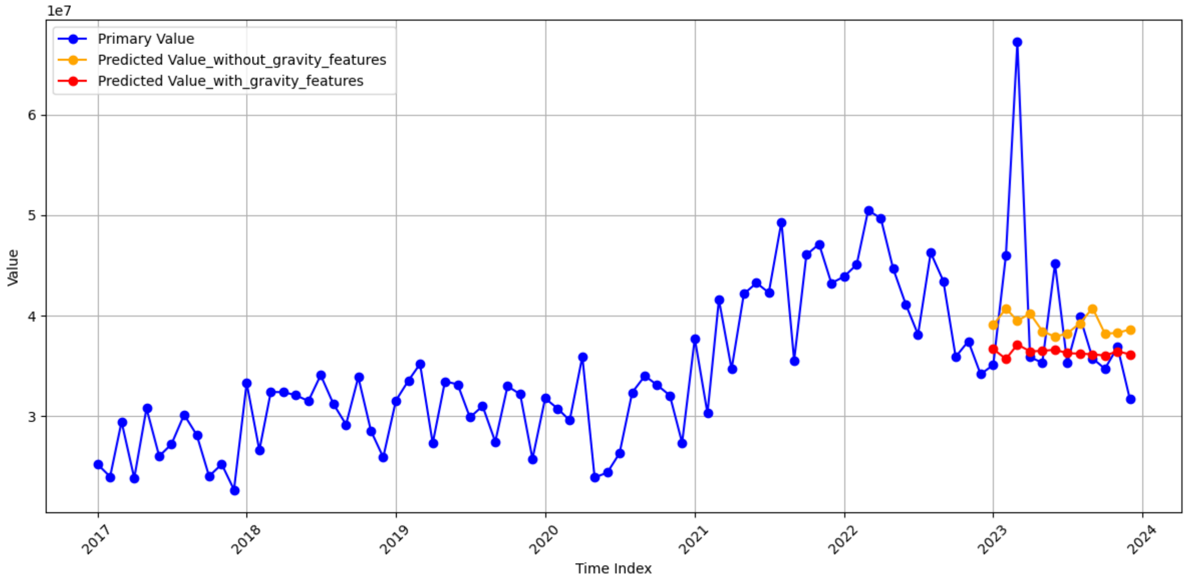

Figure 4.7: Comparison of Predicted and Primary Values for Chemical Products (HS 382499) between Germany and the United States of America using TFT Model

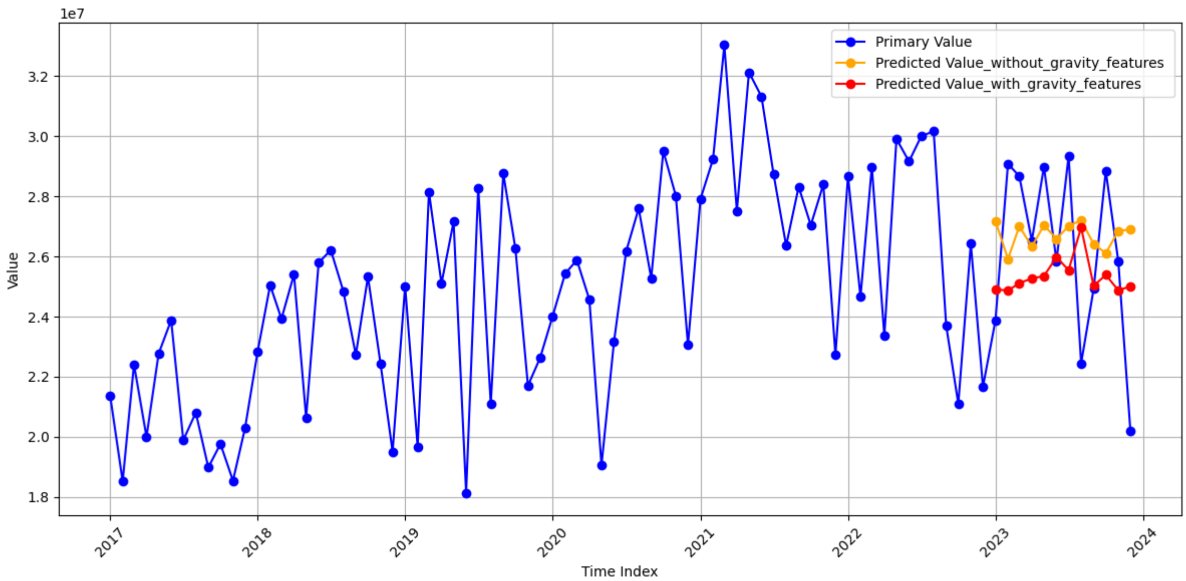

Figure 4.8: Comparison of Predicted and Primary Values for Chemical Products (HS 382499) between Germany and the Netherlands using the TFT Model

To illustrate the model's predictive performance, Figures 4.7 to 4.10 present comparisons of the predicted and actual trade values for specific commodities between selected countries. In Figure 4.7, the TFT model accurately captures the trade patterns of chemical products (HS 382499) between Germany and the United States. Several factors may explain the sharp increase in actual trade values in early 2023. Most plausibly, it reflects one-off, large-scale, or strategic batch shipments, possibly related to inventory restocking

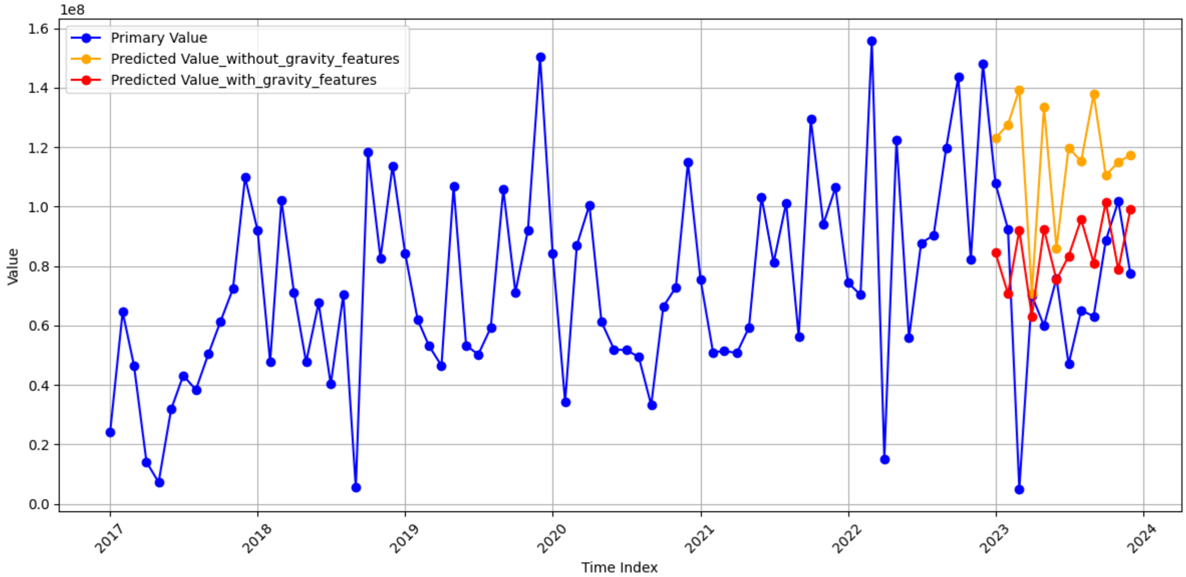

Figure 4.9: Comparison of Predicted and Primary Values for Zinc ores and concentrates (HS 260800) between Australia and China using TFT Model

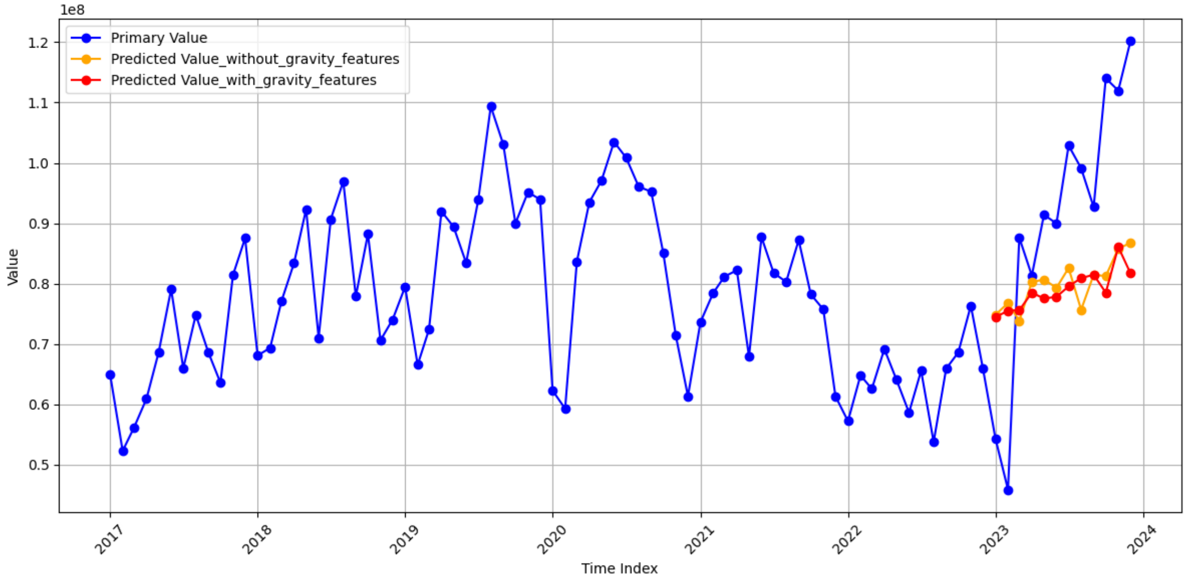

Figure 4.10: Comparison of Predicted and Primary Values for Aluminium ores and concentrates (HS 260600) between Australia and China using TFT Model

or to contractual obligations postponed from prior months. Given the volatility in global trade during this period—including ongoing supply chain disruptions and geopolitical uncertainty—such fluctuations are not uncommon. However, without detailed shipment-level data or additional contextual information (e.g., firm-level transactions), it is difficult to definitively attribute the spike to a single cause. From a modeling perspective, the TFT model performs well in tracking the overall trend, as reflected in the predicted values, which closely align with the actual values for the remainder of the time series.

Similarly, Figure 4.8 depicts the trade flows of the same chemical products between Germany and the Netherlands. The model successfully predicts the fluctuations and seasonal variations in trade values, highlighting its capability to adapt to intra-European Union trade dynamics.

Figure 4.9 showcases the trade of zinc ores and concentrates (HS 260800) between Australia and China. The model effectively captures the upward trend and the volatility in trade values, reflecting the strong demand for raw materials in China's manufacturing sector and Australia's role as a key supplier.

Figure 4.10 presents the trade flows of aluminum ores and concentrates (HS 260600) between Australia and China. The TFT model closely aligns with actual trade values, albeit with a tendency to predict slightly lower trade figures, for Figures 4.9 and 4.10, some deviations are also evident. As noted in the discussion of Figure 4.7, lower values in specific months may be balanced by shipments recorded in earlier or later months. Additionally, inaccuracies in the UN Comtrade data—sometimes corrected only at a significantly later stage—may contribute to these discrepancies. Despite this, from a modelling standpoint, the TFT model performs reasonably well in capturing the overall trend, with predicted values aligning reasonably well with actual values for the remainder of the time series.

In various ways, policymakers and companies can use insights from the TFT model to inform strategic planning. For example, they can identify and prioritize trade relationships with countries with higher demand potential, thereby tapping into new markets and diversifying their trade portfolios. This approach not only promotes economic growth but also mitigates the risks associated with reliance on a limited number of trading partners. In addition, stakeholders can optimize supply chain management by identifying vulnerabilities and adjusting sourcing and logistics strategies accordingly. This may involve diversifying suppliers, investing in alternative transportation routes, or improving inventory management to protect against potential disruptions. By incorporating these strategies, businesses can build more resilient operations, while policymakers can create supportive frameworks that facilitate smoother trade flows, even in times of uncertainty. Consideration of potential market demand and supply is particularly crucial for economies that may be too small to compete on volume and, therefore, must focus on high-value niches in the global market [101, 217].

The TFT model's feature importance analysis provides valuable economic insights into international trade flows. The GDP per capita of the destination country emerged as the most influential feature in the model's predictions, suggesting that wealthier countries have a greater propensity to import raw materials. This aligns with economic theories that posit that higher-income countries exhibit a greater demand for raw materials due to their advanced, complex industrial sectors and higher consumption levels. Policymakers in exporting countries can leverage this information to target more prosperous nations when formulating trade policies or marketing strategies for raw material exports. Additionally, businesses can use this insight to identify markets with higher demand potential, allowing them to focus their efforts on countries with greater expected demand.

The model also emphasizes the importance of temporal factors, including seasonality and economic cycles, indicating that changes in global economic conditions affect trade volumes. This knowledge can help governments and businesses optimize their trade strategies according to economic conditions.

Population size in origin and destination countries also plays a crucial role, reflecting market size and labor force availability.

The moderate importance of geographic distance is consistent with the gravity model of trade, underscoring the value of regional trade agreements and infrastructure improvements in reducing trade costs.

Static features, such as reporter and partner codes, indicate that the identities of trading countries significantly affect trade flows, driven by trade agreements and production capacities.

The importance of commodity codes suggests that certain raw materials are more vital in international trade, guiding countries to diversify their trade portfolios.

4.5 Conclusions

In this chapter, we introduced a novel application of the TFT model for forecasting international trade flows, with a particular focus on raw materials. By integrating data from the UN Comtrade database, the CEPII Gravity dataset, and the World Bank—spanning the years 2017 to 2023 and encompassing major global disruptions like the COVID-19 pandemic and the war in Ukraine—we demonstrated the TFT model's robustness, adaptability, and superior predictive accuracy compared to traditional methods such as Random Forest and even advanced models like Graph Attention Networks. Despite its strong performance, the TFT model requires substantial computational resources, time-consuming hyperparameter tuning, and careful calibration, which can limit its practicality in large-scale or real-time forecasting scenarios. To address these limitations, we propose exploring LLMs, which have demonstrated strong performance on time-series tasks. Leveraging their contextual reasoning and scalable architectures, LLMs offer a promising alternative that may provide faster, more flexible, and equally accurate forecasting capabilities. The next chapter delves into this exciting direction, marking a strategic evolution in our approach to modeling global trade flows.

Chapter 5

TransFusion-LM: Revolutionizing Trade Flow Forecasting with LLMs and TFT Integration

5.1 Introduction

In this chapter, we extend our exploration of international trade forecasting by introducing a novel hybrid approach that leverages the power of LLMs in combination with TFT. Building on TFT's strengths in handling diverse feature sets and temporal dynamics, we propose TransFusion-LM. This model improves predictive accuracy by applying advanced prompting techniques to LLMs, including GPT-3.5-turbo, GPT-4-turbo, LLaMA 2, and LLaMA 3. This approach allows us to integrate not only historical trade data, but also a wide range of economic and geographic indicators, without requiring extensive fine-tuning. By embedding TFT outputs into LLM prompts, we reduce reliance on large-scale training and enable more efficient, context-aware forecasting. This chapter outlines the motivation, architecture, prompting strategies and performance evaluation of our TransFusion-LM framework, demonstrating its effectiveness in forecasting international trade flows and its adaptability to emerging challenges in global trade.

5.2 Methodology

This section outlines our research contribution through three key components: data, TransFusion-LM, and evaluation. We begin by describing the dataset, including how we sampled it and converted it to text. Next, we explain our approach. Finally, we evaluate the performance of the model.

5.2.1 Data

In our previous research [18], in which we applied the TFT model to forecast international trade flows, we explicitly focused on raw materials. The model utilized a comprehensive dataset from 2017 to 2023, integrating information from the UN Comtrade database [223], the CEPII Gravity dataset [40], and the World Bank [240]. This dataset includes a variety of features, including geographical attributes (e.g., distance between trading partners), demographic data (e.g., population sizes) and economic indicators (e.g., GDP and GDP per capita). We trained the TFT model using both temporal and static features in the data to predict trade flows for 2023. The TFT model was trained on historical data from 2017 to 2022 and evaluated on a separate test dataset from 2023. However, in our analysis, we rely exclusively on 2022 as the historical dataset and focus on predicting trade flows for the 2023 test dataset.

5.2.1.1 Data sampling

In this paper, we selected a representative sample of 100 groups (Reporter–Partner–Commodity combinations) from the test dataset for evaluation. Several key considerations, including computational feasibility, sample diversity, generalizability, and statistical rigor, guided this choice. The selected combinations were drawn from entries with fully available, complete data across all variables, including trade values, GDP, population, and bilateral distance.

- Computational Feasibility: LLMs have inherent constraints on input token limits, especially when employing advanced prompting techniques like Few-Shot-CoT. Generating predictions for a large dataset with diverse input features—such as historical trade values, GDP, population, and TFT-predicted outputs—requires significant computational resources. Selecting 100 groups balances computational efficiency with a sufficiently large sample for robust evaluation.

- Diversity and Representativeness: The 100 groups were selected via random sampling to capture heterogeneity in trade flows across countries, commodities, and economic indicators. As summarized in Table 5.1, this sample includes:

  - A diverse range of reporter and partner countries that reflect global trade dynamics.
  - Various commodity codes, ensure that the sample is not biased toward specific industries or sectors.
  - A mix of high- and low-GDP countries, large and small populations, and varying geographical distances.

- Generalizability: The selected groups represent a broad cross-section of international trade flows, enabling the findings to be generalized to the wider dataset. The random sampling approach ensures that the sample mirrors the distribution of features (e.g., economic and geographic diversity) in the full dataset.

- Statistical Rigor: By focusing on 100 groups, we can analyze the proposed approach's performance on a manageable yet statistically meaningful subset of the data. This subset size enables us to calculate reliable evaluation metrics (e.g., MSE, MAE, R^2) and assess the model's performance comprehensively across diverse contexts.

For the few-shot approach, an additional ten instances were randomly selected from the test dataset as learning examples. These examples are independent of the primary sample and were chosen to provide the model with a diverse set of references for learning. By leveraging these additional instances, the model can generalize more effectively and improve its predictive accuracy.

While computational and token limitations constrain the sample size, the careful selection process ensures that it remains representative of the dataset's overall diversity and complexity. This balance allows us to draw meaningful insights while optimizing resource usage.

Table 5.1: Sample overview

Parameter	Value
Reporter Codes	23
Partner Codes	44
Commodity Codes	60
Features	GDP per Capita, Population, Distance
Period	January - December 2023
Size	2400 (100 group * 24 months)

5.2.1.2 Converting Time Series Data to Textual Formats

It is essential first to transform the dataset from numerical formats into natural-language sentences to apply language models to time-series forecasting. In this study, we begin by converting numerical data into textual descriptions. For example, we translate numerical month values such as 01 and 02 into their corresponding text names, "January" and "February." Similarly, we replace numerical country codes (ISO) with their respective country names and map commodity codes to descriptive names. We then introduce a template, presented in Table 5.2, which provides the template structure and examples for the transformed dataset.

The template consists of two primary components: the input and the correct value. The input prompt includes feature descriptions, such as GDP, population, and distance, as well as TFT model predictions and the corresponding trade values. LLMs can adapt this template to generate informative responses in the form of direct question-and-answer pairs or descriptive input-output pairs. These "Trade-Prompts" are saved in a single JSON file with two fields—prompt and correct value—to facilitate evaluation and enable efficient use in subsequent steps.

5.2.2 TransFusion-LM (TFL)

In this study, we apply the TransFusion-LM approach based on LLMs to predict international trade flows, building on the strong forecasting performance of the TFT model [18]. While the TFT model delivers accurate predictions, incorporating its forecasts into the LLMs as additional input enables further refinement and improves the accuracy of trade value predictions. This integration leverages the strengths of both models to deliver more precise, contextually relevant forecasts. To achieve this, we use two primary approaches: prompting and fine-tuning. For the prompting approach, we use two distinct techniques: TFL-Zero-Shot-CoT and TFL-Few-Shot-CoT, along with TFL-self-consistency across various LLMs, including GPT 3.5-turbo, GPT 4-turbo, LLama 2, and LLama 3. For the fine-tuning approach, only LLama 2 and LLama 3 are used, as GPT-3.5 and GPT-4 are proprietary and do not permit fine-tuning due to their closed-source nature.

Table 5.2: Templates for transforming Trade-Numerical to Trade-Prompt

Type	Template	Example
Input	Information about International Trade Flow in Y_{t-1} **Trade Item:** *cmdCode* **Exporter:** *reporterCode* **- GDP per capita:** $gdpcap_{E,t-1}$ **- Population:** $pop_{E,t-1}$ **Importer:** *partnerCode* **- GDP per capita:** $gdpcap_{I,t-1}$ **- Population:** $pop_{I,t-1}$ **Distance Between Countries:** *dist* **Historical Trade Values of** $t-1$ **from January to December** H_{t-1}: $[h_{t-1,1}, h_{t-1,2}, h_{t-1,3}, h_{t-1,4}, h_{t-1,5}, h_{t-1,6}, h_{t-1,7}, h_{t-1,8}, h_{t-1,9}, h_{t-1,10}, h_{t-1,11}, h_{t-1,12}]$ **Prediction Value Using TFT model from January to December in** $P_{TFT,t}$: $[p_{TFT,t,1}, p_{TFT,t,2}, p_{TFT,t,3}, p_{TFT,t,4}, p_{TFT,t,5}, p_{TFT,t,6}, p_{TFT,t,7}, p_{TFT,t,8}, p_{TFT,t,9}, p_{TFT,t,10}, p_{TFT,t,11}, p_{TFT,t,12}]$	Information about International Trade Flow in 2022 **Trade Item:** Graphite or other carbon-based preparations; in the form of pastes, blocks, plates, or other semi-manufactures. **Exporter:** China **- GDP per capita:** 12,662.58 **- Population:** 1,412,175,000 **Importer:** Germany **- GDP per capita:** 48717.99 **- Population:** 83,797,985 **Distance Between Countries:** 8,415 km **Historical Trade Values of 2022 from January to December:** [645,979.0, 501,432.0, 387,049.0, 1,092,873.0, 1,093,486.0, 485,573.0, 1,272,255.0, 785,140.0, 1,014,682.0, 407,413.0, 514,621.0, 657,128.0] **Prediction Value Using TFT model from January to December in 2023:** [828,008.75, 867,900.44, 775,645.6, 967,281.94, 871,645.1, 902,057.75, 925,329.8, 835,478.44, 892,364.4, 743,031.7, 750,624.7, 762,700.3]
The correct trade	The Corrects trade value from January to December in Y_t are: $[y_{t,1}, y_{t,2}, y_{t,3}, y_{t,4}, y_{t,5}, y_{t,6}, y_{t,7}, y_{t,8}, y_{t,9}, y_{t,10}, y_{t,11}, y_{t,12}]$	The Correct trade value from January to December in 2023 are: 645,979.0, 501,432.0, 387,049.0, 1,092,873.0, 1,093,486.0, 485,573.0, 1,272,255.0, 785,140.0, 1,014,682.0, 407,413.0, 514,621.0, 657,128.0

Furthermore, we added a system prompt to our trade prompt to clarify the model's role and guide it in systematically analyzing the input. The system prompt is formulated as: "You are an expert system in international trade analysis. Based on the following input, analyze it step by step and predict the correct trade value for each month in 2023, from January to December."

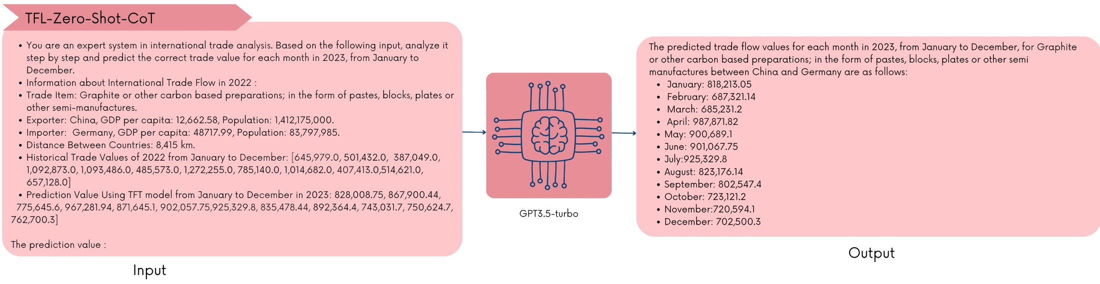

Figure 5.1: Example of TFL-Zero-Shot-CoT

5.2.2.1 TFL prompting

The prompting branch of TransFusion-LM focuses on reasoning-based forecasting, in which the model generates predictions directly from structured textual inputs without requiring additional parameter updates. Through progressive prompting strategies, ranging from zero-shot reasoning to few-shot analogical learning and self-consistency, we evaluate how different levels of contextual guidance influence the LLM's forecasting behavior.

1. **TFL-Zero-Shot-CoT (TFL-ZSC)**: We use LLMs to predict international trade flows for the twelve months of 2023. We employ a structured trade prompt to guide the model through a step-by-step analysis, thereby enhancing predictive accuracy. Our approach leverages Zero-Shot-CoT, enabling the model to make predictions without prior examples, and chain-of-thought reasoning, which allows it to break down the analysis into manageable steps, thereby improving logical reasoning. Figure 5.1 presents an example of the TFL-ZSC approach using the GPT 3.5-turbo model to predict the trade value of graphite between China as an exporter and Germany as an importer from January to December. As background information, the prompt includes key economic indicators such as 2022 GDP per capita and population data, as well as the predicted TFT value for each month in 2023.

2. **TFL-Few-Shot-Cot (TFL-FSC)**: In the Few-Shot-CoT approach, we enhance the LLM's predictions by providing a set of example prompts and corresponding correct trade values. This method offers the LLM with the same input prompt as TFL-ZSC, but enhances it with 10 learning examples. These learning examples serve as a reference for the model, help LLMs better understand context, and improve their predictions.

   Each learning example includes economic and geographic features, TFT-predicted values, and actual trade values for all months in 2022. The LLM leverages these examples to improve its response generation, aiming to produce more accurate trade-value predictions for 2023 based on the provided input. We then extract the predicted values directly from the LLM's response. Figure 5.2 presents an example of the TFL-FSC approach using GPT-3.5-turbo to predict the trade value between China as an exporter and Germany as an importer in 2023. Table 5.3 presents the Few-shot Prediction, the Correct Value, and the Residual/Error for trade value predictions between China as an exporter and Germany as an importer in 2023. The residuals are calculated as the difference between each month's predicted and correct trade values.

Month	Few-shot Prediction	Correct Value	Residual/Error
1	705,124.25	645,979.0	59,145.25
2	600,001.25	501,432.0	98,569.25
3	527,102.2	387,049.0	140,053.2
4	989,842.09	1,092,873.0	-103,030.91
5	990,975.19	1,093,486.0	-102,510.81
6	526,610.0	485,573.0	41,037.0
7	845,652.0	1,272,255.0	-426,603.0
8	845,652.22	785,140.0	60,512.22
9	978,654.14	1,014,682.0	-36,027.86
10	754,125.68	407,413.0	346,712.68
11	558,124.65	514,621.0	43,503.65
12	645,872.55	657,128.0	-11,255.45

Table 5.3: Few-shot Predictions, Actual Values, and Residuals for Trade Value Predictions in 2023 (China, Germany).

3. **TFL-Self-Consistency**: In our approach, we ran multiple LLMs with identical prompts to evaluate their output consistency. Each model was prompted using different techniques, but all runs were conducted under similar conditions to ensure uniformity in input. We then collected and analyzed the results, recording the frequency of each output across runs. We observed that each model consistently produced the most frequent output across runs, indicating high self-consistency in its predictions. This approach enables us to assess the stability and reliability of models' responses, ensuring robust, reproducible predictions across diverse LLM architectures.

   In Figure 5.3, we use the TFL-ZSC approach with the self-consistency technique. We provided a single prompt to the LLM and executed it three times, generating multiple predictions. We then analyzed the answers and selected the most

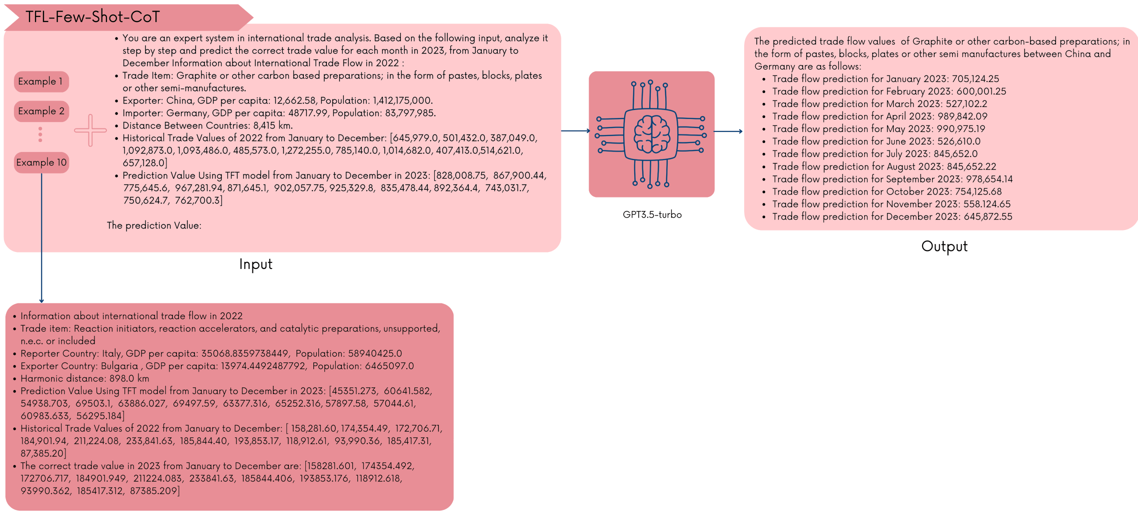

Figure 5.2: Example of TFL-Few-Shot-Cot

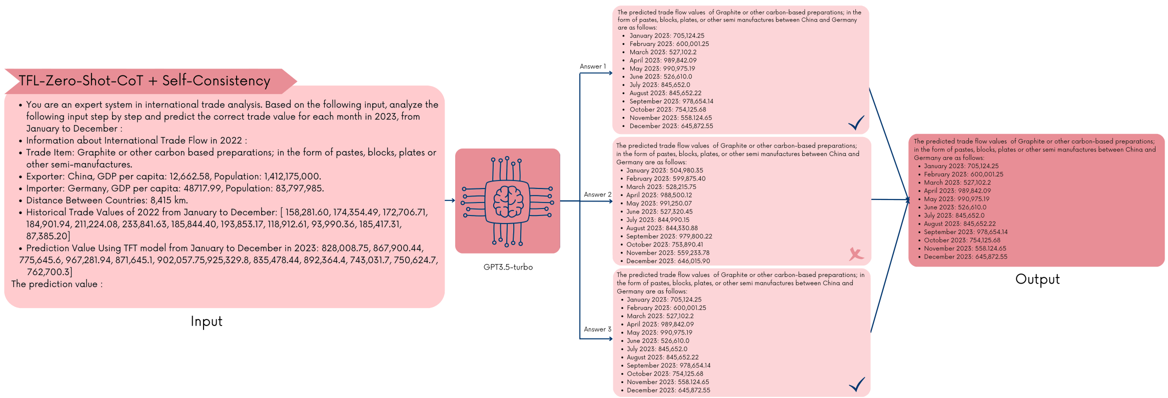

Figure 5.3: Example of TFL-zero-shot-cot with self-consistency

frequently occurring prediction based on the principle of self-consistency. This approach improved the accuracy of the results by assuming that the most consistent answer across multiple outputs is most likely reliable. Through this method, we reduce variability and improve the overall robustness of the model's predictions.

5.2.2.2 TFL Fine-Tuning

In the TransFusion-LM Fine-Tuning approach, we fine-tuned LLama 2 (13b) and LLama 3 (70b) (Table 2.1) to enhance their predictive performance for international trade flows. Model training was performed on a stratified random sample of 100 groups (Reporter-Partner-Commodity combinations) from the 2023 test data (see section 5.2.1), with all groups previously used in the evaluation of TFL explicitly excluded to prevent data leakage and ensure unbiased training. The models were trained using the same input features, GDP, GDP per capita, population, distance, predicted TFT values in 2023, and historical trade values in 2022, and evaluated using the same test dataset used for the prompting approach.

To enable efficient fine-tuning of these large models, we employed QLoRA, which enables task-specific adaptation with a significantly reduced number of trainable parameters, thereby reducing computational cost and memory requirements. Table 5.4 summarizes the hyperparameters used for fine-tuning, including a LoRA rank (r) of 8, a scaling factor (α) of 8, and a dropout rate of 0.01 applied to the model's output layers, enabling task-specific adaptation with a minimal number of trainable parameters and reduced risk of overfitting. The target weight matrices for training include q_proj, k_proj, v_proj, and lm_head. The rank and scaling factor were set to relatively low values due to the modest dataset size, thereby limiting overfitting while maintaining sufficient adaptation capacity. The AdamW optimizer was employed to enhance training stability and prevent overfitting by incorporating decoupled weight decay regularization. The models were trained for 100 epochs with a batch size of 4 and a learning rate of 2×10^{-4} to ensure stable optimization and convergence. To support training stability and reproducibility, model checkpoints were saved every 50 steps.

Table 5.4: Fine-Tuning and QLoRA Hyperparameters Used for LLama 2 and LLama 3

Hyperparameter	Value
QLoRA Parameters	
Rank (r)	8
Scaling factor (α)	8
LoRA dropout	0.01
Training Parameters	
Number of epochs	100
Batch size (per device)	4
Gradient accumulation steps	1
Learning rate	2×10^{-4}
Checkpoint saving steps	50
Optimizer	Adamw

5.2.3 Evaluation

This section provides an overview of our approach's evaluation process, detailing the experimental setups and evaluation metrics used to measure the models' performance.

5.2.3.1 Setups

The models are executed on a computing environment equipped with an NVIDIA 3090 GPU, 16 CPU cores, and 24 GB of RAM. All experiments are conducted using Python 3.10, with essential libraries such as PyTorch (2.3.1), PyTorch Forecasting (1.0.0), PyTorch Lightning (2.2.1), peft (0.17.1), and OpenAI (1.36.1) to ensure optimal resource utilization for high-performance model training and evaluation. This setup enables efficient handling of computational demands, supporting robust and scalable model execution.

5.2.3.2 Baselines

To evaluate the predictive performance of the TFL framework, we benchmark its accuracy against three established machine learning models: Random Forest (RF), Graph Attention Network (GAT), and eXtreme Gradient Boosting (XGBoost). All models were trained on the same dataset used to fine-tune LLama 2 and LLama 3, with the same explanatory variables (GDP, GDP per capita, population, bilateral distance, historical trade in 2022, and TFT-predicted trade for 2023). For methodological consistency, the same test subset used for TFL models was applied to all baselines. The configurations used for each model are summarized below:

- RF model was trained using 100 decision trees as the ensemble size and a maximum depth of 25 as a constraint on tree complexity, ensuring stable predictions while preventing overfitting.
- XGBoost model was trained using 1000 boosting rounds as the number of sequential trees and a maximum tree depth of 5 as a measure to control model complexity, ensuring stable learning without overfitting. A learning rate of 0.05 and a subsample ratio of 0.8 were used to stabilize updates and improve generalization.
- GAT model was configured with three GAT layers, each containing 128 hidden units and four attention heads, followed by a 64-unit Multi-Layer Perceptron (MLP) for the final prediction. The model was trained using the Adam optimizer, with a learning rate of 1×10^{-3} and a weight decay of 5×10^{-4} to prevent overfitting, for 300 epochs.

5.2.3.3 Evaluation metrics

To rigorously assess the predictive performance of our approach for trade values, we employ a suite of quantitative evaluation metrics that enable robust comparisons with different models. The selected metrics for this study include the Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Percentage Error (MPE), and the Coefficient of Determination (R-squared). These metrics are formally defined as follows: equations (5.1), (5.2), (5.3), (5.4), and (5.5), respectively.

$$MSE = \frac{1}{I}\sum_{i=1}^{I}(y_i - \hat{y}_i)^2 \tag{5.1}$$

$$MAE = \frac{1}{I}\sum_{i=1}^{I}|y_i - \hat{y}_i| \tag{5.2}$$

$$MAPE = \frac{1}{I}\sum_{i=1}^{I}\frac{|y_i - \hat{y}_i|}{y_i} \tag{5.3}$$

$$MPE = \frac{1}{I}\sum_{j=1}^{I}\frac{y_i - \hat{y}_i}{y_i} \tag{5.4}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{5.5}$$

In these expressions, y_i represents the observed trade values, $\hat{y}_i$ denotes the predicted trade values generated by the model, and $\bar{y}$ is the mean of the observed values. Each metric provides a distinct perspective on the model's performance. Using these evaluation metrics, we aim to comprehensively assess the predictive accuracy, precision, and robustness of the proposed model in capturing the intricate dynamics of international trade flows. The collective use of these metrics enables a nuanced understanding of the model's performance in different contexts and under varying data conditions, thereby ensuring thorough validation of our approach against the established benchmarks.

5.3 Results

As shown in Table 5.5, the TFL-FSC-GPT-4-turbo significantly outperforms the TFT and GPT-3.5-turbo in time series forecasting on various measures. Notably, it achieves an

R^2 score of 0.94, an improvement of 0.11 over the TFT model's 0.83. Additionally, TFL-FSC-GPT-4-turbo significantly improves other metrics relative to TFT, underscoring its superior accuracy and lower error rates in time-series forecasting.

Also, TFL-FSC-GPT-4-turbo demonstrates significantly better forecasting accuracy than TFL-FSC-GPT-3.5-turbo across all performance metrics. Specifically, TFL-FSC-GPT-4-turbo achieves an R^2 score of 0.94, which is a 0.9 improvement over TFL-FSC-GPT-3.5-turbo's R^2 score of 0.85. Additionally, TFL-FSC-GPT-4-turbo reduces MSE from 1.91×10^{13} to 1.23×10^{12}, and MAE from 1.23×10^{6} to 2.96×10^{5}. These improvements highlight the significant advantage of using GPT-4-turbo for time series forecasting, particularly when comparing the TFL-FSC approach across GPT-4 and GPT-3.5.

Moreover, TFL-ZSC-GPT-4-turbo achieves an R^2 score of 0.89, which is 0.05 lower than TFL-FSC-GPT-4-turbo. Similarly, TFL-ZSC-GPT-3.5-turbo records an R^2 of 0.83, slightly trailing the TFL-FSC-GPT-3.5-turbo's R^2 score of 0.85. These results emphasize the importance of leveraging the TFL-FSC approach to optimize LLM performance in trade forecasting. Overall, the findings illustrate that providing the model with a few examples (learning examples) can significantly enhance its predictive accuracy compared to TFL-ZSC. This underscores the value of TFL-ZSC in improving forecasting outcomes across different model configurations.

TFT serves as a strong benchmark for time series forecasting, with an R^2 score of 0.83. In contrast, TFL-ZSC demonstrates notable predictive capabilities, leveraging the model's ability to generalize from its extensive training data without relying on prior examples. TFL-ZSC-GPT-3.5-turbo matches TFT's R^2 score of 0.83, while TFL-ZSC-GPT-4-turbo outperforms it with an R^2 score of 0.89. Overall, TFL-ZSC demonstrates the potential to improve time-series forecasting performance compared to the established TFT model.

The TFL-fine-tuned LLama 3 model demonstrates competitive performance in time-series forecasting. Specifically, it achieves an R^2 score of 0.83, matching the TFT model, with an MSE of 2.42×10^{13} and MAE of 1.38×10^{6}. While TFL-fine-tuning-LLama3 performs well compared to classical machine-learning models such as RF, XGBoost, and GAT, it does not reach the predictive accuracy of TFL-FSC-GPT-4-turbo. These results suggest that while TFL-fine-tuning of LLama 3 can provide reasonable forecasting accuracy, the GPT-4-turbo model demonstrates superior performance due to its greater model capacity, advanced architecture, and extensive training corpus. Consequently, the TFL-FSC approach with GPT-4-turbo remains markedly superior, highlighting the advantage of leveraging GPT-4-turbo's TFL-FSC prompting capability over fine-tuning smaller models, such as LLaMA 3, for complex time series forecasting tasks.

For comparative completeness, the classical machine-learning models GAT, RF, and XGBoost were also evaluated. RF achieved an R^2 of 0.76 with an MSE of 3.94×10^{13}, while XGBoost attained an R^2 of 0.71 with an MSE of 4.79×10^{13}. The GAT-based model produced relatively low predictive accuracy, with an R^2 of 0.60. Although these models are widely used for time-series forecasting, their error magnitudes are substantially higher and their explanatory power is notably lower than that of the TFL.

Overall, the results indicate that TFL-FSC-GPT-4-turbo delivers the highest forecasting accuracy among all evaluated models, outperforming TLF-FT-LLama 3, the Transformers model (TFT), deep-learning baselines (GAT), classical machine-learning algorithms (RF, XGBoost), and TFL-ZSC using GPT-3.5 and GPT-4. These findings underscore the potential of LLM-driven forecasting frameworks and highlight the efficacy of the TFL-FSC strategy in improving predictive performance in complex temporal domains.

Technique	Model	MSE	MAE	MAPE	MPE	R^2 Score
Prompting	TFL-FSC-GPT-4-turbo	1.23×10^{12}	2.96×10^{5}	105.00	-102.00	**0.94**
	TFL-ZSC-GPT-4-turbo	2.42×10^{12}	4.34×10^{5}	115.65	-105.52	0.89
	TFL-FSC-GPT-3.5-turbo	1.91×10^{13}	1.23×10^{6}	120.25	-102.18	0.85
	TFL-ZSC-GPT-3.5-turbo	2.92×10^{13}	1.45×10^{6}	167.43	-138.92	0.83
Fine-tuning	TFL fine-tuning (LLama 3)	2.42×10^{13}	1.38×10^{6}	139.04	-110.32	0.83
ML	TFT	2.42×10^{13}	1.38×10^{6}	139.04	-110.32	0.83
	RF	3.94×10^{13}	1.56×10^{6}	9407.01	-9391.68	0.76
	XGBoost	4.79×10^{13}	1.28×10^{6}	3807.54	-1250.86	0.71
	GAT	2.87×10^{6}	1.38×10^{6}	8.75×10^{6}	-3.93×10^{6}	0.60

Table 5.5: Models' performance evaluation

5.4 Discussion

In this section, we discuss the performance and interpretability of the TransFusion-LM framework for time series forecasting. The discussion is organized into two subsections. The first, General Overview of Forecasting Results, provides a comprehensive evaluation of the framework's predictive accuracy, strengths, limitations, and the strategies, such as fine-tuning and prompt-based techniques, that contributed to its performance. The second, Strategies for Interpreting TFL Predictions, addresses the framework's interpretability by detailing how local explanations (prompt-based) and global explanations (fine-tuning-based) can be used to understand and analyze the model's reasoning behind its forecasts.

5.4.1 General Overview of TransFusion-LM f Results

We evaluate the effectiveness of the LLama 2 and LLama 3 models for time series forecasting. These models were selected for their open-source nature, which ensures free access and avoids reliance on proprietary platforms such as OpenAI, thereby providing greater flexibility. However, both models exhibited hallucinations when using the prompting approach, generating information (trade values) that was either inaccurate or entirely fabricated, as illustrated in Figure 5.4, shows an example of the TFL-ZSC approach using the LLama 3 model to predict the trade value of graphite between China (exporter) and Germany (importer) from January to December; however, the model produces a hallucinated response.

In the context of language models, hallucination refers to the production of content that lacks grounding in the input or real-world facts, resulting in misleading or erroneous responses [183, 95]. This issue is particularly concerning in critical fields such as economics and medicine, where accuracy is paramount. For instance, the models occasionally presented erroneous financial data, misrepresented medical findings, or fabricated statistics—errors that could lead to significant real-world repercussions in these domains. Although LLaMA 3 improved over LLaMA 2, hallucinations persisted, motivating the implementation of targeted mitigation strategies in this study.

To address these reliability challenges, we applied several corrective measures to reduce hallucinations and improve factual consistency. First, we fine-tuned LLama 2 and LLama 3 on domain-specific trade data using the QLoRA method, thereby enabling

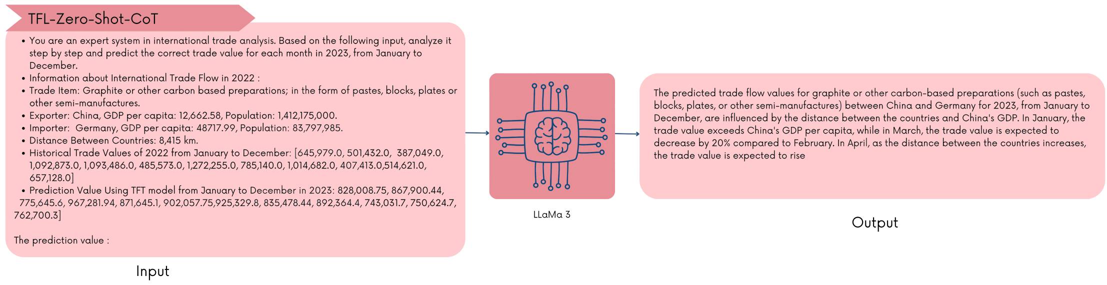

Figure 5.4: Example of TFL-Zero-Shot-CoT using LLama3 Model

the models to align their outputs with factual economic relationships better and reduce the generation of random or fabricated values. Second, we applied the CoT prompting technique, which aggregates multiple reasoning paths to identify the most recurrent and stable prediction, thereby minimizing stochastic variability across runs. Together, these strategies improved both factual grounding and reproducibility in trade forecasts. After fine-tuning, LLama 3 demonstrated the capacity to generate forecasts without hallucinations, producing predictions that closely mirror those of the TFT model. In contrast, LLama 2 continued to exhibit persistent hallucination behavior, likely due to its smaller model capacity and more restricted pretraining corpus (limited to data up to June 2023; see Table 2.1). Nevertheless, when benchmarked against GPT-3.5 and GPT-4, with TFL-FT-LLama 3 remained outperformed by the TFL-FSC GPT models. We also observed that GPT-4's superior performance can be attributed to its larger architecture and broader pretraining corpus, which contribute to stronger internal factual representations and lower hallucination rates compared with open-source counterparts. It is important to note that the GPT-3.5 and GPT-4 models used in this study correspond to the 2024 versions, before the integration of Retrieval-Augmented Generation (RAG) techniques that enable dynamic web access.

The results demonstrate that the TFL-FSC approach significantly enhances LLM prediction accuracy. Importantly, LLMs do not need to replicate TFT outputs directly; instead, they can refine their forecasts using diverse input types and additional learning examples, allowing them to forecast time series using a variety of inputs effectively, not just historical values as other studies focus on—and produce results that surpass those of powerful time-series models such as TFT. By integrating the outputs of the TFT model [18] as inputs to LLMs, this framework leverages TFT's robust temporal modeling capabilities while simultaneously exploiting LLMs' adaptive reasoning and flexibility. This synergistic integration harnesses the advantages of both state-of-the-art architectures, enabling LLMs to produce more accurate, reliable, and computationally efficient forecasts, ultimately improving overall predictive performance.

Employing LLMs for time-series forecasting offers several notable advantages. Unlike traditional time series models, which often require extensive feature selection, normalization, and large datasets [18, 278], LLMs simplify forecasting. By using prompting techniques, LLMs can effectively learn from fewer examples without fine-tuning, thereby reducing the need for complex feature engineering and extensive data preprocessing. Moreover, LLMs require neither substantial computational resources nor extensive parameter tuning, unlike conventional models. Their ability to deliver accurate predictions with limited training data and reduced computational requirements, enhanced by advanced prompting methods, makes them well-suited for time series forecasting. This approach offers a more efficient and versatile solution for managing temporal data, thus improving both performance and practical usability.

Addressing prompt engineering challenges is crucial to improving LLM performance. Recent research highlights their significance in optimizing LLM outcomes [137, 147]. Our study investigated various prompting strategies to identify the most effective approaches to improve results. We tested complex and straightforward prompts to assess their impact on the LLM's ability to understand and analyze inputs. We communicated effectively with the model by ensuring clarity and precision in the prompts, using a human-like writing style.

Furthermore, we employed advanced techniques, such as Chain of Thought (CoT), which was particularly useful for handling complex cases and scenarios requiring multi-step reasoning. We also experimented with the Tree of Thoughts (ToT) prompting technique; however, it did not yield the desired results. ToT effectively addresses complex mathematical problem-solving situations.

Despite the promising results, several challenges and limitations remain. One significant concern is the interpretability of LLMs. While models like GPT-4 exhibit high predictive accuracy, their "black box" nature may limit our understanding of how they make predictions. In economic forecasting, predicting outcomes and understanding the underlying decision-making process are crucial to derive actionable insights and policy recommendations. Despite this limitation, recent research [204, 274] has proposed strategies for interpreting LLM outputs and understanding the reasoning behind their forecasts. These approaches leverage both local and global interpretability techniques to analyze how models generate predictions. In the following subsection 5.4.2, we detail how such methods can be applied to explain and interpret the outputs of the TransFusion-LM framework.

In summary, the TransFusion-LM framework demonstrates that integrating structured temporal signals with language-based reasoning enhances both performance and adaptability. Nonetheless, continued refinement in interpretability, prompt optimization, and hallucination control remains essential for reliable deployment in sensitive domains such as economics.

5.4.2 Interpretability and explainability in TransFusion-LM

Recent perspectives on LLM interpretability emphasize that modern language models redefine explainability by combining feature attribution with natural-language reasoning and dataset-level explanation capabilities [204, 274]. In line with this vision, our TransFusion-LM framework leverages both prompt-based interpretability (local explanations) and fine-tuning interpretability (global explanations) as complementary forms of explanation.

- **Prompt-based interpretability as local explanation:** Our prompting setups (TFL-ZSC, TFL-FSC, and TFL-Self-Consistency) correspond to what Singh et al. [204] classify as local interpretability, explaining a single model generation through reasoning transparency. By decomposing predictions into Chain-of-Thought reasoning, TransFusion-LM allows users to inspect how variables such as GDP, population, and TFT baselines influence trade forecasts step by step. As in the interactive natural-language explanations proposed by Singh et al., these reasoning chains can be queried, compared, and refined, bridging human understanding and model behavior. We also adopt their notion of data grounding, testing prompts both with and without features (economic features, historical trade, and TFT prediction), as well as baselines to measure factual dependence and methods to reduce hallucinated justifications. The inclusion of these features anchors the model's reasoning in quantitative evidence (as shown in Example 1, Table 5.6), helping it generate explanations that are more consistent with real-world data. This grounding makes the model's outputs more interpretable and fact-based, reflected in the performance improvement from an R^2 of -0.22 without features to 0.34 with features, and in more "faithful" textual explanations in the sense proposed by [204]. Furthermore, we incorporate the FSC technique to enhance interpretability. By providing the model with representative input–output examples, it learns to map key features to underlying economic patterns. This not only improves prediction accuracy but also ensures that the reasoning chains align more closely with economic logic. Example-based prompting acts as an interpretable analogical reasoning mechanism, allowing users to trace how the model generalizes from specific instances to new predictions (as illustrated in Example 2, Table 5.6). Additionally, the Self-Consistency technique adds another dimension of interpretability by evaluating the stability of reasoning paths across multiple generations. By selecting the most frequently occurring output, the resulting prediction reflects the model's dominant internal reasoning rather than a single stochastic path. This method, which boosts accuracy while highlighting the reasoning chains the model finds most plausible.

- **Fine-tuning-based interpretability as global explanation:** Fine-tuned LLama 3 model, adapted with QLoRA, provides what Singh et al. define as global or mechanistic interpretability, explaining how model parameters internalize patterns across an entire dataset. We extend this with feature-attribution tools (SHAP) to quantify each variable's contribution to the predictions. After fine-tuning TFL-FT-LLama 3, SHAP-based analysis was used to interpret the relative importance and directional impact of the features. Figure 5.5 presents a bar plot of the mean absolute SHAP values across features, highlighting their relative importance in TFL-FT-LLama 3 predictions. The x-axis represents the SHAP value, indicating the impact of each feature on the model's output. Positive SHAP values show features pushing the prediction higher, while negative SHAP values indicate features pushing the prediction lower. The color of each dot represents the feature value, with blue indicating low values and red indicating high values. The TFT prediction feature has the broadest range of SHAP values, from 8 to +8 million, suggesting the most decisive influence on the model's output. High values of this feature (in pink) increase the predicted result, while low values (in blue) reduce it. The TradeValue_2022 feature also has a clear positive impact, meaning that higher historical values tend to increase predictions. In contrast, features related to GDP, population, distance, and trade codes have SHAP values close to zero, suggesting they have negligible effects on the model's decisions. Overall, the results indicate that the model primarily relies on TFT_prediction and TradeValue_2022, while the other variables play only minor supporting roles in shaping the final prediction. Figure 5.6 presents the SHAP feature importance for the surrogate model, showing the contribution of each variable to the overall prediction. The TFT prediction feature dominates the model, with a mean SHAP value exceeding 3.5 million. This indicates that it has the strongest effect on the predicted output compared to all other features. The TradeValue_2022 trade feature ranks second, with a moderate positive influence, suggesting that higher historical values tend to increase the prediction. In contrast,

Table 5.6: Illustrative examples of prompt-based explainability in TransFusion-LM (TFL–TFT)

Examples	Prompt Context	Evaluation
Example 1	**Prompt (with historical trade, TFT prediction, economic features):** You are an expert system in international trade analysis. Based on the following input, analyze step by step and predict the correct trade value for each month in 2023, from January to December: Information about 2022 trade flow: Trade Item: Graphite; Exporter: China (GDP per capita, Population); Importer: Germany (GDP per capita, Population); Distance; Historical Trade Values 2022; TFT prediction values.	MSE: 5.71×10^{10} MAE:2.11×10^{5} MAPE: 35.92 MPE: 23.35 R^2 : 0.34
	Prompt (No historical trade or features provided): You are an expert system in international trade analysis. Based on the following input, analyze step by step and predict the correct trade value for each month in 2023, from January to December: Trade Item: Graphite; Exporter: China; Importer: Germany.	MSE: 1.92×10^{14}; MAE: 1.38×10^{7}; MAPE: 2.21×10^{3}; MPE: -2.27×10^{3}; R^2: -0.22
Example 2	Figure 5.1: Present TFL-Zero-Shot-CoT	MSE: 5.71×10^{10} MAE: 2.11×10^{5} MAPE: 35.92 MPE: 23.35; R^2 : 0.34
	Figure 5.2: Present TFL-Few-Shot-CoT	MSE: 3.04×10^{10}; MAE: 1.22×10^{5}; MAPE: 19.36; MPE: -9.76; R^2 : 0.65

features such as GDP-related indicators (gdpcap and gdp), distance, population, and trade codes have much smaller mean SHAP values, meaning they play only a minor role in shaping the model's final output. Overall, the results indicate that TFT_prediction and TradeValue_2022 primarily drive the model's predictive performance, while the remaining features contribute marginally to its performance. Additionally, although LLama's relatively smaller scale (7B–13B parameters) limits its implicit factual grounding compared with GPT-4's trillion-parameter architecture, domain-specific fine-tuning effectively mitigates this gap by aligning model representations with structured trade-economic data. This improvement is exemplified through a comparative evaluation of TFL-ZSC-LLaMA 3 and TFL-FT-LLaMA 3. When prompted to *"Predict the trade value between China and Germany for 2023,"* as illustrated in Figure 5.4, the TFL-ZSC-LLaMA 3 model generated a largely qualitative narrative, explaining that *"the predicted trade flow values for graphite or other carbon-based preparations between China and Germany for 2023 are influenced by distance and China's GDP. In January, the trade value exceeds China's GDP per capita, while in March it decreases by 20% compared to February, and in April, as distance increases, the trade value rises."* In contrast, the TFL-FT-LLaMA 3 model produced precise, quantitative forecasts, providing monthly trade values for 2023 of 828,008.75, 867,900.44, 775,645.6, 967,281.94, 871,645.1, 902,057.75, 925,329.8, 835,478.44, 892,364.4, 743,031.7, 750,624.7, and 762,700.3. This comparison clearly demonstrates that domain-specific fine-tuning substantially enhances both factual precision and interpretability, transforming the model from a descriptive, text-based system into a data-driven, quantitatively grounded predictive framework. These findings are consistent with the observations of Mumuni and Mumuni [162], who emphasize that dataset-aware adaptation improves both accuracy and transparency when LLMs are finetuned to specialized scientific domains.

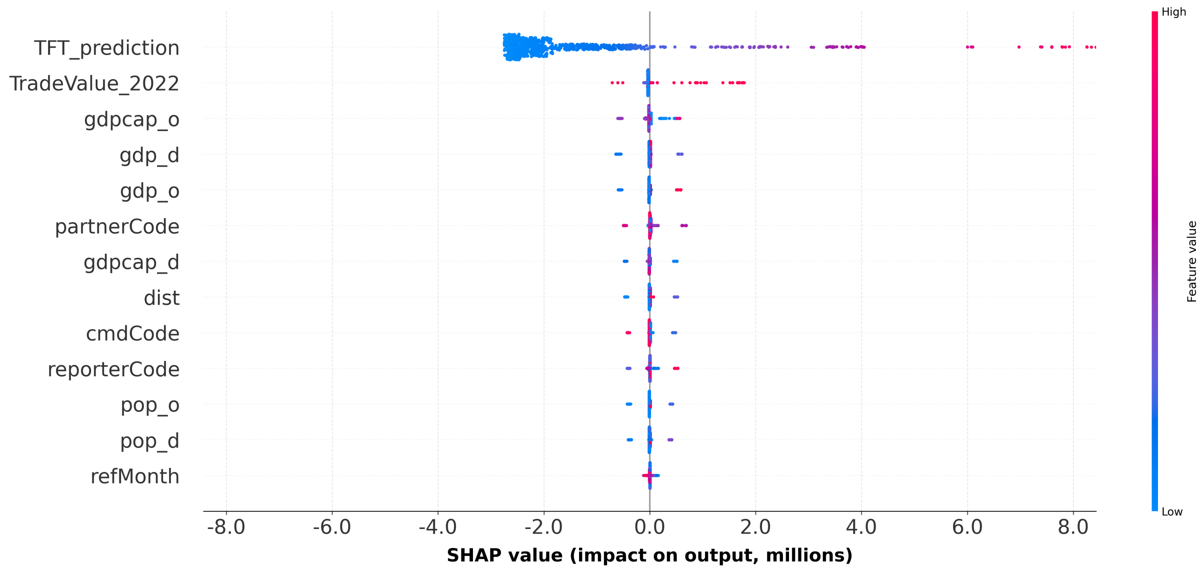

Figure 5.5: Beeswarm plot of the TFL-FT-LLaMA 3 model

- **Interactive and dataset-level interpretability:** Singh et al. [204] highlight interactive and dataset explanation as a new interpretability frontier. In TransFusion-LM, the few-shot CoT reasoning (TFL-FSC) and the expert validation of reasoning

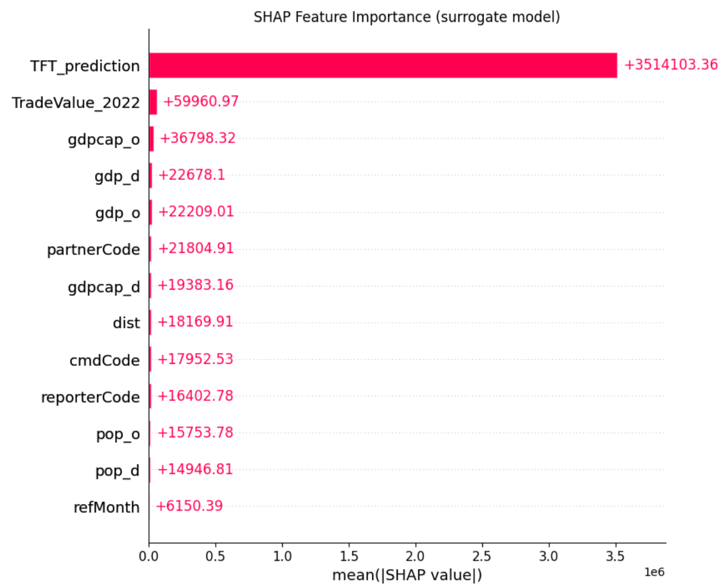

Figure 5.6: Features importance plot of the TFL-FT-LLaMA 3 model

traces act as early forms of interactive explanation. Experts reviewed a subset of reasoning outputs to confirm that the generated narratives were consistent with economic theory, thus complementing automatic attribution with human, grounded validation. Extending this approach could turn TransFusion-LM into an interactive auditing system capable of conversational explanations of trade-data relationships.

By combining prompt-based local reasoning, fine-tuned global attribution, and human-interactive evaluation, TransFusion-LM operationalizes the interpretability dimensions proposed by [204, 274], attribution, natural-language reasoning, reasoning decomposition, and data grounding. This integration transforms interpretability from a post-hoc visualization exercise into a multilevel dialogue among data, model, and domain experts —an essential step toward trustworthy LLM-based forecasting in economic science.

5.5 Conclusions

In conclusion, this study highlights the potential of TransFusion-LM, a hybrid approach that integrates the strengths of TFT and LLMs for TSF in international trade. By leveraging advanced prompting techniques such as Few-Shot and Chain-of-Thought, TransFusion-LM achieves higher accuracy and adaptability while reducing the need for extensive data and feature engineering. Although challenges like hallucination and interpretability remain, this method represents a significant step toward more flexible, intelligent, and scalable forecasting solutions. Its promising performance paves the way for broader applications in trade, finance, and beyond.

Conclusion

This thesis has developed and validated novel predictive frameworks for trend analysis through link prediction in social networks and time-series forecasting in international trade, two complementary tasks that capture structural and temporal dynamics in complex systems. Both problems are characterized by high-dimensional, heterogeneous, and dynamically evolving data, in which identifying meaningful trends is essential for understanding system behavior and enabling reliable prediction. In such settings, traditional statistical, econometric, and heuristic-based approaches often struggle to model nonlinear dependencies, long-range temporal patterns, and evolving structural relationships. The central objective of this research was therefore to design scalable, accurate, and interpretable predictive frameworks that advance trend analysis for both network-based and time-series-based problems in real-world environments.

In the context of social networks, this work demonstrated that conventional similarity-based methods and standalone machine learning models are insufficient to capture the evolving trends that govern link formation fully. To address these shortcomings, a hybrid link prediction framework was proposed that integrates similarity measures with supervised learning and an optimization-inspired feature selection strategy. By adopting a Binary Butterfly Optimization–inspired mechanism enhanced by mutation, the framework efficiently identifies the most informative structural features, allowing predictive models to focus on relevant relational trends while avoiding redundant or noisy information. The incorporation of SHAP-based explainability further improves transparency, enabling meaningful interpretation of the learned link formation patterns. Extensive experiments on real-world datasets confirm the effectiveness of the proposed approach, with the CNN-based model achieving prediction accuracies of 98% on the Facebook dataset and 85% on the Enron Email-Eu-Core dataset, substantially outperforming traditional heuristic and baseline learning methods.

In parallel, this thesis advanced time-series forecasting in international trade, a domain in which complex temporal dynamics, economic indicators, geographic factors, and geopolitical events shape long-term economic trends. Using enriched datasets constructed from the UN Comtrade database, the CEPII Gravity dataset, and the World Bank, an interpretable forecasting framework based on the Temporal Fusion Transformer was developed. The TFT model effectively captures long-range temporal dependencies and integrates heterogeneous exogenous variables, enabling the extraction of meaningful trade trends and yielding a 17% improvement in the coefficient of determination (R^2) compared to baseline models such as Random Forest and Graph Attention Networks. These results demonstrate the suitability of transformer-based architectures for modeling complex multivariate trade trends.

Building on the strengths of the TFT framework, this research introduced TransFusion-LM. This hybrid forecasting architecture enhances trend analysis by integrating large language models through prompt-based strategies without fine-tuning. By embedding

TFT outputs and structured contextual features into prompts, TransFusion-LM enables efficient and context-aware forecasting while reducing training complexity. Empirical evaluations show that prompt-based GPT-4-turbo configurations achieve the best overall performance, reaching an R^2 score of up to 94% and substantially reducing forecasting errors compared to fine-tuned language models, standalone TFT, and classical machine learning approaches. Beyond improved predictive accuracy, this framework provides interpretable, natural-language explanations of forecasting outcomes, supporting trust and transparency in trend-based decision-making for sensitive economic domains.

Overall, the findings of this thesis demonstrate that practical trend analysis in complex systems requires models that jointly capture structural evolution, temporal dynamics, and contextual information. By directly addressing the challenges identified in the introduction and validating the proposed methods through extensive real-world experiments, this work contributes novel and practical methodologies to both network-based and time-series-based trend analysis. These contributions provide valuable tools for researchers and practitioners and support more informed, transparent, and reliable decision-making in socially and economically impactful domains, such as social network analysis and international trade policy.

Future research can further advance trend analysis in complex systems by extending the proposed frameworks to more dynamic and information-rich settings. In link prediction, applying the approach to temporal social networks and incorporating node profiles and semantic features would enable the modeling of evolving relational trends and link persistence over time. In international trade forecasting, integrating external and time-varying knowledge sources such as trade policies, tariff changes, geopolitical events, and economic reports could improve the detection and interpretation of emerging trade trends that are not fully captured by historical numerical data. Combining structural, temporal, and contextual information in this manner would enhance the robustness, adaptability, and interpretability of trend-based predictive models in real-world social and economic systems.

Bibliography

[1] Faima Abbasi, Muhammad Muzammal, Kashif Naseer Qureshi, Ibrahim Tariq Javed, Tiziana Margaria, and Noel Crespi. Exploiting optimised communities in directed weighted graphs for link prediction. *Online Social Networks and Media*, 31:100222, 2022.

[2] Hossein Abbasimehr and Reza Paki. Improving time series forecasting using lstm and attention models. *Journal of Ambient Intelligence and Humanized Computing*, 13(1):673–691, 2022.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[5] Sanskar Agarwal, Shivam Sharma, Kazi Newaj Faisal, and Rishi Raj Sharma. Time-series forecasting using svmd-lstm: A hybrid approach for stock market prediction. *Journal of Probability and Statistics*, 2025(1):9464938, 2025.

[6] Asan Agibetov. Neural graph embeddings as explicit low-rank matrix factorization for link prediction. *Pattern Recognition*, 133:108977, 2023.

[7] Iftikhar Ahmad, Muhammad Usman Akhtar, Salma Noor, and Ambreen Shahnaz. Missing link prediction using common neighbor and centrality based parameterized algorithm. *Scientific reports*, 10(1):1–9, 2020.

[8] Faez Ahmed, Yaxin Cui, Yan Fu, and Wei Chen. A graph neural network approach for product relationship prediction. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 85383, page V03AT03A036. American Society of Mechanical Engineers, 2021.

[9] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.

[10] Yasir Ali, Fizza Hussain, and Md Mazharul Haque. Advances, challenges, and future research needs in machine learning-based crash prediction models: A systematic review. *Accident Analysis & Prevention*, 194:107378, 2024.

[11] Hamed Alizadegan, Behzad Rashidi Malki, Arian Radmehr, Hossein Karimi, and Mohsen Asghari Ilani. Comparative study of long short-term memory (lstm), bidirectional lstm, and traditional machine learning approaches for energy consumption prediction. *Energy Exploration & Exploitation*, 43(1):281–301, 2025.

[12] Vito Walter Anelli, Amra Delić, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, et al. Recsys 2020 challenge workshop: engagement prediction on twitter's home timeline. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 623–627, 2020.

[13] DM Anisuzzaman, Jeffrey G Malins, Paul A Friedman, and Zachi I Attia. Fine-tuning large language models for specialized use cases. *Mayo Clinic Proceedings: Digital Health*, 3(1), 2025.

[14] Anonymous. Facebook wall posts network dataset. `http://konect.cc/networks/facebook-wosn-wall/`, 2017.

[15] Sankalap Arora and Priyanka Anand. Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, 116:147–160, 2019.

[16] Sankalap Arora and Satvir Singh. Butterfly optimization algorithm: a novel approach for global optimization. *Soft computing*, 23:715–734, 2019.

[17] Djihad Arrar, Nadjet Kamel, and Abdelaziz Lakhfif. A comprehensive survey of link prediction methods. *The Journal of Supercomputing*, 80(3):3902–3942, 2024.

[18] Djihad Arrar, Chahinez Ounoughi, Nadjet Kamel, Tarmo Kalvet, Marek Tiits, and Sadok Ben Yahia. High-accuracy prediction of international raw material trade flows using temporal fusion transformer. *International Journal of Data Science and Analytics*, pages 1–21, 2025.

[19] Djihad Arrar, Chahinez Ounoughi, Nadjet Kamel, Marek Tiits, Tarmo Kalvet, and Sadok Ben Yahia. High-accuracy prediction of international raw material trade flows using temporal fusion transformer. *Neural Computing and Applications*, 2025.

[20] G Avinash, Hariom Pachori, Avinash Sharma, and SukhDev Mishra. Time series forecasting of bed occupancy in mental health facilities in india using machine learning. *Scientific Reports*, 15(1):2686, 2025.

[21] Jibouni Ayoub, Dounia Lotfi, Mohamed El Marraki, and Ahmed Hammouch. Accurate link prediction method based on path length between a pair of unlinked nodes and their degree. *Social Network Analysis and Mining*, 10(1):1–13, 2020.

[22] Ersin Aytaç. Forecasting turkey's hazelnut export quantities with facebook's prophet algorithm and box-cox transformation. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 10(1):33, 2021.

[23] Furqan Aziz, Haji Gul, Ishtiaq Muhammad, and Irfan Uddin. Link prediction using node information on local paths. *Physica A: Statistical Mechanics and its Applications*, 557:124980, 2020.

[24] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[25] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[26] Emanuele Pio Barracchia, Gianvito Pio, Albert Bifet, Heitor Murilo Gomes, Bernhard Pfahringer, and Michelangelo Ceci. Lp-robin: link prediction in dynamic networks exploiting incremental node embedding. *Information Sciences*, 606:702–721, 2022.

[27] V Batagelj and A Mrvar. Pajek datasets http://vlado. fmf. uni-lj. si/pub/networks/data/mix. *USAir97. net*, 2006.

[28] Thomas Bayes. Naive bayes classifier. *Article Sources and Contributors*, pages 1–9, 1968.

[29] Saman Behrouzi, Zahra Shafaeipour Sarmoor, Khosrow Hajsadeghi, and Kaveh Kavousi. Predicting scientific research trends based on link prediction in keyword networks. *Journal of Informetrics*, 14(4):101079, 2020.

[30] Kamal Berahmand, Elahe Nasiri, Saman Forouzandeh, and Yuefeng Li. A preference random walk algorithm for link prediction through mutual influence nodes in complex networks. *Journal of king saud university-computer and information sciences*, 34(8):5375–5387, 2022.

[31] Kamal Berahmand, Elahe Nasiri, Mehrdad Rostami, and Saman Forouzandeh. A modified deepwalk method for link prediction in attributed social network. *Computing*, 103:2227–2249, 2021.

[32] Bogdan Bochenek and Zbigniew Ustrnul. Machine learning in weather prediction and climate analyses—applications and perspectives. *Atmosphere*, 13(2):180, 2022.

[33] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[34] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[36] Ertan Bütün, Mehmet Kaya, and Reda Alhajj. Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks. *Information Sciences*, 463:152–165, 2018.

[37] Lei Cai and Shuiwang Ji. A multi-scale approach for graph link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3308–3315, 2020.

[38] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[39] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

[40] CEPII. CEPII Gravity Database, Version 2022-2. `http://www.cepii.fr/`, 2022. Accessed: 2025.

[41] Lang Chai, Lilan Tu, Xinyi Yu, Xianjia Wang, and Juan Chen. Link prediction and its optimization based on low-rank representation of network structures. *Expert Systems with Applications*, 219:119680, 2023.

[42] Shang-Wei Chao and Feng-Li Lian. A self-adaptive frequency decomposition approach for denoising to enhance data-driven learning of cyclic time series in medical signal estimation. *IEEE Access*, 2025.

[43] P CHEBOTAREV. The matrix-forest theorem and measuring relations in small social groups. *Autom. Remote Control*, 58(9):1505–1514, 1997.

[44] Guangfu Chen, Haibo Wang, Yili Fang, and Ling Jiang. Link prediction by deep non-negative matrix factorization. *Expert Systems with Applications*, 188:115991, 2022.

[45] Guangfu Chen, Chen Xu, Jingyi Wang, Jianwen Feng, and Jiqiang Feng. Nonnegative matrix factorization for link prediction in directed complex networks using pagerank and asymmetric link clustering information. *Expert Systems with Applications*, 148:113290, 2020.

[46] Guangfu Chen, Chen Xu, Jingyi Wang, Jianwen Feng, and Jiqiang Feng. Robust non-negative matrix factorization for link prediction in complex networks using manifold regularization and sparse learning. *Physica A: Statistical Mechanics and its Applications*, 539:122882, 2020.

[47] Hongxu Chen, Hongzhi Yin, Xiangguo Sun, Tong Chen, Bogdan Gabrys, and Katarzyna Musial. Multi-level graph convolutional networks for cross-platform anchor link prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1503–1511, 2020.

[48] Jinyin Chen, Jian Zhang, Xuanheng Xu, Chenbo Fu, Dan Zhang, Qingpeng Zhang, and Qi Xuan. E-lstm-d: A deep learning framework for dynamic network link prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(6):3699–3712, 2019.

[49] Ling Chen, Jun Cui, Xing Tang, Yuntao Qian, Yansheng Li, and Yongjun Zhang. Rlpath: a knowledge graph link prediction method using reinforcement learning based attentive relation path searching and representation learning. *Applied Intelligence*, 52(4):4715–4726, 2022.

[50] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[51] Kuo Chi, Hui Qu, and Guisheng Yin. Link prediction for existing links in dynamic networks based on the attraction force. *Chaos, Solitons & Fractals*, 159:112120, 2022.

[52] K Clark. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[53] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[54] Leonardo Cotta, Beatrice Bevilacqua, Nesreen Ahmed, and Bruno Ribeiro. Causal lifting and link prediction. *arXiv preprint arXiv:2302.01198*, 2023.

[55] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[56] Nur Nasuha Daud, Siti Hafizah Ab Hamid, Chempaka Seri, Muntadher Saadoon, and Nor Badrul Anuar. Scalable link prediction in twitter using self-configured framework. *arXiv preprint arXiv:2208.09798*, 2022.

[57] DemandSage. Facebook usage and statistics in 2025. `https://www.demandsage.com/facebook-statistics/?utm_source=chatgpt.com`, 2025. Accessed: 2025-12-17.

[58] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[59] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

[60] Xingbo Du, Junchi Yan, Rui Zhang, and Hongyuan Zha. Cross-network skip-gram embedding for joint network alignment and link prediction. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):1080–1095, 2020.

[61] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[62] Habib Ahmed Elsayir. Forecasting arima model for foreign trade statistics. *Standard Scientific Research and Essays*, 2(12):636–648, 2014.

[63] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

[64] Yoav Freund and Llew Mason. The alternating decision tree learning algorithm. In *icml*, volume 99, pages 124–133, 1999.

[65] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.

[66] Qing Ge. Enhancing stock market forecasting: A hybrid model for accurate prediction of s&p 500 and csi 300 future prices. *Expert Systems with Applications*, 260:125380, 2025.

[67] Hossien Ghorbanzadeh, Amir Sheikhahmadi, Mahdi Jalili, and Sadegh Sulaimany. A hybrid method of link prediction in directed graphs. *Expert Systems with Applications*, 165:113896, 2021.

[68] Riccardo Giubilei and Pierpaolo Brutti. Supervised classification for link prediction in facebook ego networks with anonymized profile information. *Journal of Classification*, pages 1–24, 2022.

[69] Pablo M Gleiser and Leon Danon. Community structure in jazz. *Advances in complex systems*, 6(04):565–573, 2003.

[70] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[71] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[72] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[73] Didem Güleryüz and Erdemalp Özden. The prediction of brent crude oil trend using lstm and facebook prophet. *Avrupa Bilim ve Teknoloji Dergisi*, (20):1–9, 2020.

[74] Feipeng Guo, Wei Zhou, Zifan Wang, Chunhua Ju, Shaobo Ji, and Qibei Lu. A link prediction method based on topological nearest-neighbors similarity in directed networks. *Journal of Computational Science*, 69:102002, 2023.

[75] Li Guo, Yijie Sun, et al. Economic forecasting analysis of high-dimensional multifractal action based on financial time series. *International Journal for Housing Science and Its Applications*, 45:11–19, 2024.

[76] Samraj Gupta, Sanchal Nachappa, and Nirmala Paramanandham. Stock market time series forecasting using comparative machine learning algorithms. *Procedia Computer Science*, 252:893–904, 2025.

[77] Huimin Han, Zehua Liu, Jiuhao Li, and Zhixiong Zeng. Challenges in remote sensing based climate and crop monitoring: navigating the complexities using ai. *Journal of cloud computing*, 13(1):1–14, 2024.

[78] Yan Han, Lihua Mi, Lian Shen, CS Cai, Yuchen Liu, Kai Li, and Guoji Xu. A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting. *Applied Energy*, 312:118777, 2022.

[79] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[80] Kazunobu Hayakawa and Hiroshi Mukunoki. The impact of covid-19 on international trade: Evidence from the first shock. *Journal of the Japanese and International Economies*, 60:101135, 2021.

[81] Chaobo He, Junwei Cheng, Xiang Fei, Yu Weng, Yulong Zheng, and Yong Tang. Community preserving adaptive graph convolutional networks for link prediction in attributed networks. *Knowledge-Based Systems*, page 110589, 2023.

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[83] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[84] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[85] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

[86] Mohammad Hosseini, Catherine A Gao, David M Liebovitz, Alexandre M Carvalho, Faraz S Ahmad, Yuan Luo, Ngan MacDonald, Kristi L Holmes, and Abel Kho. An exploratory survey about using chatgpt in education, healthcare, and research. *Plos one*, 18(10):e0292216, 2023.

[87] Desislava Hristova, Anastasios Noulas, Chloë Brown, Mirco Musolesi, and Cecilia Mascolo. A multilayer approach to multiplexity and link prediction in online geo-social networks. *EPJ Data Science*, 5:1–17, 2016.

[88] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[89] Da Huang and Fangyuan Lei. Temporal group-aware graph diffusion networks for dynamic link prediction. *Information Processing & Management*, 60(3):103292, 2023.

[90] Pham Canh Huy, Nguyen Quoc Minh, Nguyen Dang Tien, and Tao Thi Quynh Anh. Short-term electricity load forecasting based on temporal fusion transformer model. *Ieee Access*, 10:106296–106304, 2022.

[91] Md. Aliul Islam, Mst. Sharmin Akter Sumy, Md. Alhaz Uddin, and Md. Sazzad Hossain. Fitting arima model and forecasting for the tea production, and internal consumption of tea (per year) and export of tea. *International Journal of Material and Mathematical Sciences*, 2(1):8–15, 2020.

[92] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579, 1901.

[93] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, 2002.

[94] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2021.

[95] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[96] Ayoub Jibouni, Dounia Lotfi, Mohamed El Marraki, and Ahmed Hammouch. A novel parameter free approach for link prediction. In *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6. IEEE, 2018.

[97] Georgitzikis K. Trade codes of non-food, non-fuel raw materials and their products. (JRC136041), 2023.

[98] Katikapalli Subramanyam Kalyan. A survey of gpt-3 family large language models including chatgpt and gpt-4. *Natural Language Processing Journal*, page 100048, 2023.

[99] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammu: a survey of transformer-based biomedical pretrained language models. *Journal of biomedical informatics*, 126:103982, 2022.

[100] Fatemeh Karimi, Shahriar Lotfi, and Habib Izadkhah. Community-guided link prediction in multiplex networks. *Journal of Informetrics*, 15(4):101178, 2021.

[101] Rainer Kattel, Tiina Randma-Liiv, and Tarmo Kalvet. Small states, innovation and administrative capacity. In V. Bekkers, J. Edelenbos, and B. Steijn, editors, *Innovation in the Public Sector: Linking Capacity and Leadership*, pages 61–81. Palgrave Macmillan UK, London, 2011.

[102] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[103] Mohammad Mehdi Keikha, Maseud Rahgozar, and Masoud Asadpour. Deeplink: A novel link prediction framework based on deep learning. *Journal of Information Science*, 47(5):642–657, 2021.

[104] Yang-Seon Kim, Moon Keun Kim, Nuodi Fu, Jiying Liu, Junqi Wang, and Jelena Srebric. Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models. *Sustainable Cities and Society*, 118:105570, 2025.

[105] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks.

[106] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[107] Holger Kreft and Walter Jetz. Global patterns and determinants of vascular plant diversity. *Proceedings of the National Academy of Sciences*, 104(14):5925–5930, 2007.

[108] Mukesh Kumar, Shivansh Mishra, and Bhaskar Biswas. Pqklp: Projected quantum kernel based link prediction in dynamic networks. *Computer Communications*, 196:249–267, 2022.

[109] Mukesh Kumar, Shivansh Mishra, Rahul Deo Pandey, and Bhaskar Biswas. Cflp: A new cost based feature for link prediction in dynamic networks. *Journal of Computational Science*, 62:101726, 2022.

[110] Anisha Kumari, Ranjan Kumar Behera, Kshira Sagar Sahoo, Anand Nayyar, Ashish Kumar Luhach, and Satya Prakash Sahoo. Supervised link prediction using structured-based feature extraction in social network. *Concurrency and Computation: practice and Experience*, 34(13):e5839, 2022.

[111] Zefeng Lai, Tangjie Wu, Xihong Fei, and Qiang Ling. Bert4st:: Fine-tuning pre-trained large language model for wind power forecasting. *Energy Conversion and Management*, 307:118331, 2024.

[112] T Jaya Lakshmi and S Durga Bhavani. Link prediction approach to recommender systems. *Computing*, 106(7):2157–2183, 2024.

[113] Thanh Le, Huy Tran, and Bac Le. Knowledge graph embedding with the special orthogonal group in quaternion space for link prediction. *Knowledge-Based Systems*, 266:110400, 2023.

[114] Sang-Woong Lee, Jawad Tanveer, Amir Masoud Rahmani, Hamid Alinejad-Rokny, Parisa Khoshvaght, Gholamreza Zare, Pegah Malekpour Alamdari, and Mehdi Hosseinzadeh. Sfgcn: Synergetic fusion-based graph convolutional networks approach for link prediction in social networks. *Information Fusion*, 114:102684, 2025.

[115] Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 388–396. IEEE, 2019.

[116] Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

[117] Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25, 2012.

[118] Feiyu Li, Zhibo Wan, Thomas Koch, Guokuan Zan, Mengjiao Li, Zhonghai Zheng, and Bo Liang. Improving the accuracy of multi-step prediction of building energy consumption based on eemd-pso-informer and long-time series. *Computers and Electrical Engineering*, 110:108845, 2023.

[119] Longjie Li, Shiyu Fang, Shenshen Bai, Shijin Xu, Jianjun Cheng, and Xiaoyun Chen. Effective link prediction based on community relationship strength. *IEEE Access*, 7:43233–43248, 2019.

[120] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[121] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382, 2023.

[122] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, 2016.

[123] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, 2003.

[124] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[125] Marcus Lim, Azween Abdullah, and NZ Jhanjhi. Data fusion-link prediction for evolutionary network with deep reinforcement learning. *International Journal of Advanced Computer Science and Applications*, 11(6), 2020.

[126] Marcus Lim, Azween Abdullah, NZ Jhanjhi, Muhammad Khurram Khan, and Mahadevan Supramaniam. Link prediction in time-evolving criminal network with deep reinforcement learning technique. *IEEE Access*, 7:184797–184807, 2019.

[127] Marcus Lim, Azween Abdullah, NZ Jhanjhi, and Mahadevan Supramaniam. Hidden link prediction in criminal networks using the deep reinforcement learning technique. *Computers*, 8(1):8, 2019.

[128] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI open*, 3:111–132, 2022.

[129] Chunjiang Liu, Yikun Han, Haiyun Xu, Shihan Yang, Kaidi Wang, and Yongye Su. A community detection and graph-neural-network-based link prediction approach for scientific literature. *Mathematics*, 12(3):369, 2024.

[130] H Liu, M Li, C Yang, and L Jia. Application of time-series analysis to urban climate change assessment. *International Journal of Environmental Science and Technology*, 22(12):11037–11044, 2025.

[131] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 4125–4129, 2023.

[132] Shangke Liu, Ke Liu, Zheng Wang, Yuanyuan Liu, Bin Bai, and Rui Zhao. Investigation of a transformer-based hybrid artificial neural networks for climate data prediction and analysis. *Frontiers in Environmental Science*, 12:1464241, 2025.

[133] Si-Yuan Liu, Jing Xiao, and Xiao-Ke Xu. Sign prediction by motif naive bayes model in social networks. *Information Sciences*, 541:316–331, 2020.

[134] Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *Europhysics Letters*, 89(5):58007, 2010.

[135] Yingjie Liu, Shihu Liu, Fusheng Yu, and Xiyang Yang. Link prediction algorithm based on the initial information contribution of nodes. *Information Sciences*, 608:1591–1616, 2022.

[136] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[137] Leo S Lo. The clear path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship*, 49(4):102720, 2023.

[138] Haohui Lu and Shahadat Uddin. A parameterised model for link prediction using node centrality and similarity measure based on graph embedding. *Neurocomputing*, 593:127820, 2024.

[139] Minrong Lu and Xuerong Xu. Trnn: An efficient time-series recurrent neural network for stock price prediction. *Information Sciences*, 657:119951, 2024.

[140] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[141] Hongsheng Luo, Longjie Li, Yakun Zhang, Shiyu Fang, and Xiaoyun Chen. Link prediction in multiplex networks using a novel multiple-attribute decision-making approach. *Knowledge-Based Systems*, 219:106904, 2021.

[142] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

[143] Snigdha Luthra, Rakesh Kumar, and Meenu Gupta. A novel hybrid model based on sentiment analysis using community guided link prediction for connecting people with similar interest. *IEEE Access*, 2024.

[144] Laishui Lv, Dalal Bardou, Peng Hu, Yanqiu Liu, and Gaohang Yu. Graph regularized nonnegative matrix factorization for link prediction in directed temporal networks using pagerank centrality. *Chaos, Solitons & Fractals*, 159:112107, 2022.

[145] Neda Maleki, Oxana Lundström, Arslan Musaddiq, John Jeansson, Tobias Olsson, and Fredrik Ahlgren. Future energy insights: Time-series and deep learning models for city load forecasting. *Applied Energy*, 374:124067, 2024.

[146] Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1, 2020.

[147] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer, 2023.

[148] Costas Mavromatis and George Karypis. Graph infoclust: Maximizing coarse-grain mutual information in graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 541–553. Springer, 2021.

[149] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[150] Qiao Mi, Xiaoming Wang, and Yaguang Lin. A double attention graph network for link prediction on temporal graph. *Applied Soft Computing*, 136:110059, 2023.

[151] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[152] Naoto Minakawa, Kiyoshi Izumi, and Hiroki Sakaji. Bilateral trade flow prediction by gravity-informed graph auto-encoder. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 2327–2332. IEEE, 2022.

[153] What Is Data Mining. Data mining: Concepts and techniques. *Morgan Kaufinann*, 10(559-569):4, 2006.

[154] Shivansh Mishra, Shashank Sheshar Singh, Ajay Kumar, and Bhaskar Biswas. Elp: Link prediction in social networks based on ego network perspective. *Physica A: Statistical Mechanics and its Applications*, 605:128008, 2022.

[155] Shivansh Mishra, Shashank Sheshar Singh, Ajay Kumar, and Bhaskar Biswas. Mnerlp-mul: Merged node and edge relevance based link prediction in multiplex networks. *Journal of Computational Science*, 60:101606, 2022.

[156] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[157] Hebatallah A Mohamed, Diego Pilutti, Stuart James, Alessio Del Bue, Marcello Pelillo, and Sebastiano Vascon. Locality-aware subgraphs for inductive link prediction in knowledge graphs. *Pattern Recognition Letters*, 167:90–97, 2023.

[158] Anderson Monken, Flora Haberkorn, Munisamy Gopinath, Laura Freeman, and Feras Batarseh. Graph neural networks for modeling causality in international trade. volume 34, 04 2021.

[159] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[160] João P Moutinho, André Melo, Bruno Coutinho, István A Kovács, and Yasser Omar. Quantum link prediction in complex networks. *Physical Review A*, 107(3):032605, 2023.

[161] Diyawu Mumin, Lei-Lei Shi, and Lu Liu. An efficient algorithm for link prediction based on local information: Considering the effect of node degree. *Concurrency and Computation: Practice and Experience*, 34(7):e6289, 2022.

[162] Fuseini Mumuni and Alhassan Mumuni. Explainable artificial intelligence (xai): from inherent explainability to large language models. *arXiv preprint arXiv:2501.09967*, 2025.

[163] Carlos Pedro Muniz, Ronaldo Goldschmidt, and Ricardo Choren. Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowledge-Based Systems*, 156:129–137, 2018.

[164] Shaghayegh Najari, Mostafa Salehi, Vahid Ranjbar, and Mahdi Jalili. Link prediction in multiplex networks based on interlayer similarity. *Physica A: Statistical Mechanics and its Applications*, 536:120978, 2019.

[165] Abdulqadir J Nashwan and Ahmad A AbuJaber. Harnessing the power of large language models (llms) for electronic health records (ehrs) optimization. *Cureus*, 15(7), 2023.

[166] Elahe Nasiri, Kamal Berahmand, and Yuefeng Li. A new link prediction in multiplex networks using topologically biased random walks. *Chaos, Solitons & Fractals*, 151:111230, 2021.

[167] Elahe Nasiri, Kamal Berahmand, Mehrdad Rostami, and Mohammad Dabiri. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, 137:104772, 2021.

[168] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[169] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

[170] Amril Nazir, Abdul Khalique Shaikh, Abdul Salam Shah, and Ashraf Khalil. Forecasting energy consumption demand of customers in smart grid using temporal fusion transformer (tft). *Results in Engineering*, 17:100888, 2023.

[171] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[172] Ezekiel NN Nortey, Edmund F Agyemang, Enoch Sakyi-Yeboah, Obu-Amoah Ampomah, and Louis Agyekum. Ai meets economics: Can deep learning surpass machine learning and traditional statistical models in inflation time series forecasting? *Data Science in Finance and Economics*, 2025.

[173] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied soft computing*, 93:106384, 2020.

[174] Kobby Panford-Quainoo, Avishek Joey Bose, and Michaël Defferrard. Bilateral trade modelling with graph neural networks. In *ICLR workshop on practical ML for developing countries*, 2020.

[175] Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, 2009.

[176] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[177] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[178] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[179] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[180] Samira Rafiee, Chiman Salavati, and Alireza Abdollahpouri. Cndp: Link prediction based on common neighbors degree penalization. *Physica A: Statistical Mechanics and its Applications*, 539:122950, 2020.

[181] Nasir Rahim, Shaker El-Sappagh, Sajid Ali, Khan Muhammad, Javier Del Ser, and Tamer Abuhmed. Prediction of alzheimer's progression based on multimodal deep-learning-based fusion and visual explainability of time-series data. *Information Fusion*, 92:363–388, 2023.

[182] Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE access*, 12:26839–26874, 2024.

[183] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

[184] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[185] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.

[186] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs.

[187] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[188] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

[189] Emrullah ŞAHiN, Naciye Nur Arslan, and Durmuş Özdemir. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, 37(2):859–965, 2025.

[190] Gerard Salton. Introduction to modern information retrieval. *McGraw-Hill*, 1983.

[191] Gerard Salton and Chung-Shu Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 1973.

[192] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 519–527, 2020.

[193] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[194] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36, 2024.

[195] Paul Schmiedmayer, Adrit Rao, Philipp Zagar, Vishnu Ravi, Aydin Zahedivash, Arash Fereydooni, and Oliver Aalami. Llm on fhir–demystifying health records. *arXiv preprint arXiv:2402.01711*, 2024.

[196] Bassem Sellami, Chahinez Ounoughi, Tarmo Kalvet, Marek Tiits, and Diego Rincon-Yanez. Harnessing graph neural networks to predict international trade flows. *Big Data and Cognitive Computing*, 8(6), 2024.

[197] Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 99–111. Springer, 2020.

[198] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

[199] Na Shan, Longjie Li, Yakun Zhang, Shenshen Bai, and Xiaoyun Chen. Supervised link prediction in multiplex networks. *Knowledge-Based Systems*, 203:106168, 2020.

[200] Preeti Sharma. Extrapolative modeling of international trade using arima model: A performance comparison between india and china during covid-19. *SAMVAD: SIBM Pune Research Journal*, XXI:71–91, 2020.

[201] Mei-Li Shen, Cheng-Feng Lee, Hsiou-Hsiang Liu, Po-Yin Chang, and Cheng-Hong Yang. Effective multinational trade forecasting using lstm recurrent neural network. *Expert Systems with Applications*, 182:115199, 2021.

[202] Cole E Short and Jeremy C Short. The artificially intelligent entrepreneur: Chatgpt, prompt engineering, and entrepreneurial rhetoric creation. *Journal of Business Venturing Insights*, 19:e00388, 2023.

[203] Anil Shukla. LLaMA 4: Meta's Multimodal Leap in AI - Architecture, Capabilities, and Comparative Analysis. `https://doi.org/10.13140/RG.2.2.30302.75844`, April 2025. ResearchGate.

[204] Akshat Singh, Seyed-Mohammad Eslami, Yilun Wu, Sameer Singh, Sumedh Sontakke, Zachary C. Lipton, and Marco Tulio Ribeiro. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

[205] Ashwini Kumar Singh and Kailasam Lakshmanan. Pilhnb: Popularity, interests, location used hidden naive bayesian-based model for link prediction in dynamic social networks. *Neurocomputing*, 461:562–576, 2021.

[206] Shashank Sheshar Singh, Shivansh Mishra, Ajay Kumar, and Bhaskar Biswas. Clp-id: Community-based link prediction using information diffusion. *Information Sciences*, 514:402–433, 2020.

[207] Shashank Sheshar Singh, Divya Srivastva, Ajay Kumar, and Vishal Srivastava. Flp-id: Fuzzy-based link prediction in multiplex social networks using information diffusion perspective. *Knowledge-Based Systems*, 248:108821, 2022.

[208] Xizhuoran Song, Yan Zhang, Rui Pan, and Hansheng Wang. Link prediction for statistical collaboration networks incorporating institutes and research interests. *IEEE Access*, 10:104954–104965, 2022.

[209] Payal Soni, Yogya Tewari, and Deepa Krishnan. Machine learning approaches in stock price prediction: A systematic review. In *Journal of Physics: Conference Series*, volume 2161, page 012065. IOP Publishing, 2022.

[210] Thorvald A Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

[211] Liyilei Su, Xumin Zuo, Rui Li, Xin Wang, Heng Zhao, and Bingding Huang. A systematic review for transformer-based long-term series forecasting. *Artificial Intelligence Review*, 58(3):80, 2025.

[212] Mengdi Sun and Minghu Tang. A review of link prediction algorithms in dynamic networks. *Mathematics*, 13(5):807, 2025.

[213] Mark Sundberg and Ken Tucker. *International trade in services.* Taylor & Francis, 2024.

[214] Minghu Tang and Wenjun Wang. Cold-start link prediction integrating community information via multi-nonnegative matrix factorization. *Chaos, Solitons & Fractals*, 162:112421, 2022.

[215] Ye Tao, Ying Li, and Zhonghai Wu. Temporal link prediction via reinforcement learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3470–3474. IEEE, 2021.

[216] Sean J Taylor and Benjamin Letham. Forecasting at scale. *PeerJ Preprints*, 5:e3190v2, 2017.

[217] Marek Tiits and Tarmo Kalvet. Nordic Small Countries in the Global High-tech Value Chains: The Case of Telecommunications Systems Production in Estonia. Technical report, Working Papers in Technology Governance and Economic Dynamics, 2012.

[218] Marek Tiits, Tarmo Kalvet, Chahinez Ounoughi, and Sadok Ben Yahia. Relatedness and product complexity meet gravity models of international trade. *Journal of Open Innovation: Technology, Market, and Complexity*, 10(2):100288, 2024.

[219] Jan Tinbergen. *Suggestions for an international economic policy.* Periodicals Service Company, 1962.

[220] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[221] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[222] Marta Tuninetti, Alberto Aleta, Daniela Paolotti, Yamir Moreno, and Michele Starnini. Prediction of new scientific collaborations through multiplex networks. *EPJ data science*, 10(1):25, 2021.

[223] United Nations. UN Comtrade Database. `https://comtradeplus.un.org/`, 2023. Accessed: 2025.

[224] Mohammadsadegh Vahidi Farashah, Akbar Etebarian, Reza Azmi, and Reza Ebrahimzadeh Dastjerdi. A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data*, 8:1–24, 2021.

[225] Corne Van Zyl, Xianming Ye, and Raj Naidoo. Harnessing explainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of grad-cam and shap. *Applied Energy*, 353:122079, 2024.

[226] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 1999.

[227] Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[228] Sergiy Verstyuk and Michael R Douglas. Machine learning the gravity equation for international trade. *Available at SSRN 4053795*, 2022.

[229] M Vickers and S Chan. Representing classroom social structure. *Victoria Institute of Secondary Education, Melbourne*, 1981.

[230] Christian Von Mering, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, 417(6887):399–403, 2002.

[231] Guanghui Wang, Yufei Wang, Jimei Li, and Kaidi Liu. A multidimensional network link prediction algorithm and its application for predicting social relationships. *Journal of Computational Science*, 53:101358, 2021.

[232] Jingwei Wang, Yunlong Ma, Min Liu, and Weiming Shen. Link prediction based on community information and its parallelization. *IEEE Access*, 7:62633–62645, 2019.

[233] Tong Wang, Xing-Sheng He, Ming-Yang Zhou, and Zhong-Qian Fu. Link prediction in evolving networks based on popularity of nodes. *Scientific reports*, 7(1):7147, 2017.

[234] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[235] Ziqian Wang, Zhihao Chen, Yang Yang, Chanjuan Liu, Xi'an Li, and Jinran Wu. A hybrid autoformer framework for electricity demand forecasting. *Energy Reports*, 9:3800–3812, 2023.

[236] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[237] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[238] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[239] Woan Foong Wong. The round trip effect: Endogenous transport costs and international trade. *American Economic Journal: Applied Economics*, 14(4):127–166, 2022.

[240] World Bank. World Bank Open Data. `https://databank.worldbank.org/`, 2024. Accessed: 2025.

[241] World Trade Organization (WTO). Global trade outlook and statistics – april 2025. `https://www.wto.org/english/res_e/publications_e/trade_outlook25_e.htm`, 2025. Accessed: 2025-12-17.

[242] Michael Wornow, Alejandro Lozano, Dev Dash, Jenelle Jindal, Kenneth W Mahaffey, and Nigam H Shah. Zero-shot clinical trial patient matching with llms. *arXiv preprint arXiv:2402.05125*, 2024.

[243] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[244] Jiayun Wu, Langzhou He, Tao Jia, and Li Tao. Temporal link prediction based on node dynamics. *Chaos, Solitons & Fractals*, 170:113402, 2023.

[245] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4840–4841, 2022.

[246] Yunpeng Xiao, Rui Li, Xingyu Lu, and Yanbing Liu. Link prediction based on feature representation and fusion. *Information Sciences*, 548:1–17, 2021.

[247] Jie Xu, Yong Lin, Mu Yang, and Lanjing Zhang. Statistics and pitfalls of trend analysis in cancer research: a review focused on statistical packages. *Journal of Cancer*, 11(10):2957, 2020.

[248] Rong-qin Xu, Ming-yang Zhou, and Hao Liao. Pnr: How to optimally combine different link prediction approaches? *Information Sciences*, 584:342–359, 2022.

[249] Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[250] Hao Xue and Flora D Salim. Utilizing language models for energy load forecasting. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 224–227, 2023.

[251] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[252] Hemant Yadav and Amit Thakkar. Noa-lstm: An efficient lstm cell architecture for time series forecasting. *Expert Systems with Applications*, 238:122333, 2024.

[253] Cheng-Hong Yang, Cheng-Feng Lee, and Po-Yin Chang. Export-and import-based economic models for predicting global trade using deep learning. *Expert Systems with Applications*, 218:119590, 2023.

[254] Liping Yang, Xin Jiang, Yiming Ji, Hua Wang, Ajith Abraham, and Hongbo Liu. Gated graph convolutional network based on spatio-temporal semi-variogram for link prediction in dynamic complex network. *Neurocomputing*, 505:289–303, 2022.

[255] Yiming Yang, Jan O Pedersen, et al. A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35. Citeseer, 1997.

[256] Yujie Yang, Long Wang, and Dong Liu. Anchor link prediction across social networks based on multiple consistency. *Knowledge-Based Systems*, 257:109939, 2022.

[257] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[258] Yabing Yao, Ruisheng Zhang, Fan Yang, Yongna Yuan, Qingshuang Sun, Yu Qiu, and Rongjing Hu. Link prediction via layer relevance of multiplex networks. *International Journal of Modern Physics C*, 28(08):1750101, 2017.

[259] Işil Yenidoğan, Aykut Çayir, Ozan Kozan, Tuğçe Dağ, and Çiğdem Arslan. Bitcoin forecasting using arima and prophet. In *2018 3rd international conference on computer science and engineering (UBMK)*, pages 621–624. IEEE, 2018.

[260] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 555–564, 2017.

[261] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

[262] Shijin Yuan, Guansong Wang, Bin Mu, and Feifan Zhou. Tianxing: A linear complexity transformer model with explicit attention decay for global weather forecasting. *Advances in Atmospheric Sciences*, 42(1):9–25, 2025.

[263] Weiwei Yuan, Kangya He, Donghai Guan, Li Zhou, and Chenliang Li. Graph kernel based link prediction for signed social networks. *Information Fusion*, 46:1–10, 2019.

[264] Herman Yuliansyah, Zulaiha Ali Othman, and Azuraliza Abu Bakar. A new link prediction method to alleviate the cold-start problem based on extending common neighbor and degree centrality. *Physica A: Statistical Mechanics and its Applications*, 616:128546, 2023.

[265] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[266] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[267] Adnan Zeb, Summaya Saif, Junde Chen, Anwar Ul Haq, Zhiguo Gong, and Defu Zhang. Complex graph convolutional network for link prediction in knowledge graphs. *Expert Systems with Applications*, 200:116796, 2022.

[268] Muhan Zhang and Yixin Chen. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 575–583, 2017.

[269] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[270] Peiliang Zhang, Jiatao Chen, Chao Che, Liang Zhang, Bo Jin, and Yongjun Zhu. Iea-gnn: Anchor-aware graph neural network fused with information entropy for node classification and link prediction. *Information Sciences*, 634:665–676, 2023.

[271] Shu Zhang, Junxin Wang, Sigang Yu, Ruoyang Wang, Junwei Han, Shijie Zhao, Tianming Liu, and Jinglei Lv. An explainable deep learning framework for characterizing and interpreting human brain states. *Medical Image Analysis*, 83:102665, 2023.

[272] Zehua Zhang, Shilin Sun, Guixiang Ma, and Caiming Zhong. Line graph contrastive learning for link prediction. *Pattern Recognition*, 140:109537, 2023.

[273] Zhongbao Zhang, Jian Wen, Li Sun, Qiaoyu Deng, Sen Su, and Pengyan Yao. Efficient incremental dynamic link prediction algorithms in social network. *Knowledge-Based Systems*, 132:226–235, 2017.

[274] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

[275] Huaqin Zhao, Zhengliang Liu, Zihao Wu, Yiwei Li, Tianze Yang, Peng Shu, Shaochen Xu, Haixing Dai, Lin Zhao, Gengchen Mai, et al. Revolutionizing finance with llms: An overview of applications and insights. *arXiv preprint arXiv:2401.11641*, 2024.

[276] Yuncong Zhao, Yiyang Sun, Yaning Huang, Longjie Li, and Hu Dong. Link prediction in heterogeneous networks based on metapath projection and aggregation. *Expert Systems with Applications*, page 120325, 2023.

[277] Jianxing Zheng, Zifeng Qin, Suge Wang, and Deyu Li. Attention-based explainable friend link prediction with heterogeneous context information. *Information Sciences*, 597:211–229, 2022.

[278] Peijun Zheng, Heng Zhou, Jiang Liu, and Yosuke Nakanishi. Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture. *Applied Energy*, 349:121607, 2023.

[279] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[280] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[281] Mingqiang Zhou, Qizhi Han, Mengjiao Li, Kunpeng Li, and Zhiyuan Qian. Nearest neighbor walk network embedding for link prediction in complex networks. *Physica A: Statistical Mechanics and its Applications*, 620:128757, 2023.

[282] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

[283] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71:623–630, 2009.

[284] Li Zou, Chao Wang, An Zeng, Ying Fan, and Zengru Di. Link prediction in growing networks with aging. *Social Networks*, 65:1–7, 2021.

[285] Unai Zulaika, Ruben Sanchez-Corcuera, Aitor Almeida, and Diego Lopez-de Ipina. Lwp-wl: Link weight prediction based on cnns and the weisfeiler–lehman algorithm. *Applied Soft Computing*, 120:108657, 2022.