TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN
VIỆN TRÍ TUỆ NHÂN TẠO

Kĩ thuật và công nghệ dữ liệu lớn cho Trí tuệ nhân tạo

BÁO CÁO BÀI TẬP LỚN

NHÓM 12

Xây dựng hệ thống dự đoán xu hướng giá cổ phiếu dựa trên LLM và Khung suy luận quan hệ – thời gian (TRR)

Thành viên nhóm: Phạm Nhật Quang 23020413
Bùi Minh Quân 23020415
Phan Quang Trường 23020443
Giáo viên hướng dẫn: TS. Trần Hồng Việt
ThS. Ngô Minh Hương

Mục lục

Phân công nhiệm vụ thành viên

Dựa trên khối lượng công việc và thế mạnh chuyên môn, nhóm nghiên cứu đã thống nhất phân chia nhiệm vụ cụ thể như bảng dưới đây:

Họ và tên	**Chi tiết công việc đảm nhận**
Phan Nhật Quang	• Nghiên cứu và cài đặt thuật toán TRR. • Hiện thực cơ chế Graph RAG và tính toán trọng số PageRank/Time Decay. • Prompt Engineering: Tối ưu hóa câu lệnh cho Gemini. • Thực hiện backtesting và tính toán các chỉ số đánh giá (Accuracy, F1-Score).
Bùi Minh Quân	• Xây dựng Data Pipeline: Cấu hình Docker cho Kafka, Zookeeper, Neo4j, MongoDB. • Viết script thu thập dữ liệu tự động. • Tiền xử lý dữ liệu và phân tích cảm xúc. • Thiết kế schema cơ sở dữ liệu (Graph model & Document model).
Phan Quang Trường	• Xây dựng module trích xuất đồ thị. • Phát triển Dashboard tương tác bằng Streamlit. • Trực quan hóa dữ liệu: Biểu đồ nến và Đồ thị mạng lưới (NetworkX/Plotly). • Xử lý Streaming: Cấu hình Kafka Producer/Consumer cho dữ liệu giá real-time.

Chương 1

Giới thiệu và Tổng quan

1.1 Bối cảnh và Vấn đề nghiên cứu

1.1.1 Thị trường Chứng khoán và Thách thức từ Dòng chảy Thông tin

Thị trường chứng khoán Việt Nam đang trong giai đoạn phát triển mạnh mẽ với quy mô vốn hóa và thanh khoản ngày càng lớn. Bản chất của thị trường là một môi trường biến động liên tục, nơi giá trị của cổ phiếu được quyết định bởi quy luật cung cầu. Tuy nhiên, ẩn sau những con số nhảy múa trên bảng điện tử là sự chi phối mạnh mẽ của luồng thông tin khổng lồ.

Thông tin trên thị trường tồn tại dưới nhiều dạng thức: từ các dữ liệu có cấu trúc như báo cáo tài chính, chỉ số vĩ mô, đến dữ liệu phi cấu trúc như tin tức báo chí, thông cáo chính sách, và đặc biệt là các thảo luận trên mạng xã hội. Một đặc điểm cốt lõi của thông tin tài chính là tính **liên kết và tác động dây chuyền**. Một sự kiện đơn lẻ (ví dụ: "Giá dầu thế giới tăng") không chỉ tác động trực tiếp mà còn tạo ra một chuỗi phản ứng lan truyền qua nhiều thực thể kinh tế, từ chuỗi cung ứng, chi phí vận tải, đến lợi nhuận doanh nghiệp, cuối cùng mới phản ánh vào giá cổ phiếu.

Vấn đề đặt ra cho các nhà đầu tư cá nhân và tổ chức là làm thế nào để xử lý khối lượng thông tin khổng lồ này (Information Overload), lọc nhiễu, và quan trọng nhất là nhìn ra được các mối liên kết ngầm định để đưa ra quyết định đầu tư chính xác.

1.1.2 Hạn chế của các Mô hình Dự đoán Truyền thống

Trong khoa học dữ liệu tài chính, việc dự báo giá cổ phiếu là bài toán kinh điển. Các phương pháp tiếp cận trước đây thường được chia làm hai nhóm chính:

- **Phân tích kỹ thuật truyền thống:** Dựa thuần túy vào biến động giá và khối lượng trong quá khứ.
- **Mô hình Học sâu (Deep Learning):** Sử dụng các kiến trúc mạng nơ-ron như LSTM (Long Short-Term Memory) hay GRU để xử lý dữ liệu chuỗi thời gian.

Mặc dù đạt được một số thành công nhất định, các mô hình này bộc lộ những hạn chế rõ rệt khi đối mặt với bài toán dự báo xu hướng dựa trên tin tức:

1. **Thiếu khả năng lý luận (Reasoning):** Các mô hình truyền thống thường coi văn bản tin tức là một chuỗi ký tự để trích xuất đặc trưng thống kê, thay vì hiểu sâu về ngữ nghĩa. Chúng hoạt động như một "hộp đen" (black-box), đưa ra kết quả dự báo nhưng không thể giải thích được nguyên nhân (Explainability) – một yếu tố sống còn trong quyết định tài chính.
2. **Bỏ qua tính liên kết thực thể:** Các mô hình chuỗi thời gian thường xử lý từng mã cổ phiếu một cách độc lập, bỏ qua mối quan hệ tương quan chằng chịt giữa các công ty, ngành hàng và chuỗi cung ứng.
3. **Hạn chế với sự kiện mới (Zero-shot):** Khả năng thích ứng kém với các sự kiện thiên nga đen hoặc các tin tức chưa từng xuất hiện trong tập dữ liệu huấn luyện.

1.2 Mục tiêu đề tài

Xuất phát từ những hạn chế trên, đề tài này tập trung nghiên cứu và xây dựng một hệ thống dự báo xu hướng giá chứng khoán thế hệ mới, kết hợp giữa sức mạnh của Dữ liệu lớn (Big Data) và Trí tuệ nhân tạo (AI).

Mục tiêu cụ thể của đề tài bao gồm:

- **Xây dựng Đồ thị Tri thức (Knowledge Graph) tài chính:** Chuyển đổi dữ liệu tin tức phi cấu trúc thành dạng đồ thị có cấu trúc, biểu diễn rõ ràng các thực thể (Doanh nghiệp, Sự kiện, Ngành) và mối quan hệ tác động giữa chúng.
- **Ứng dụng phương pháp Temporal Relational Reasoning (TRR):** Phát triển mô hình có khả năng "suy luận" theo thời gian, kết hợp thông tin hiện tại với bối cảnh quá khứ (Memory) để phát hiện các tín hiệu xu hướng.
- **Tích hợp Mô hình Ngôn ngữ lớn (LLM):** Tận dụng khả năng hiểu ngôn ngữ tự nhiên của LLM để trích xuất thông tin và thực hiện suy luận nhân quả, giúp hệ thống không chỉ dự báo (Tăng/Giảm) mà còn cung cấp lý do giải thích.
- **Triển khai hệ thống đường ống dữ liệu (Data Pipeline):** Thiết kế kiến trúc xử lý dữ liệu hỗn hợp (Hybrid), kết hợp giữa xử lý theo lô (Batch processing) cho tin tức và xử lý thời gian thực (Real-time streaming) cho dữ liệu giá.

1.3 Phạm vi nghiên cứu

Để đảm bảo tính khả thi và tập trung sâu vào giải pháp công nghệ, đề tài giới hạn phạm vi nghiên cứu như sau:

- **Đối tượng dữ liệu:**
  - *Thị trường:* Tập trung vào danh mục các cổ phiếu thuộc nhóm VN30 trên thị trường chứng khoán Việt Nam.
  - *Nguồn tin:* Dữ liệu tin tức tiếng Việt từ các trang tin tài chính uy tín (Fireant, CafeF...) và dữ liệu thảo luận cộng đồng.
  - *Dữ liệu giá:* Dữ liệu giao dịch khớp lệnh lịch sử và thời gian thực.
- **Phạm vi công nghệ:**
  - Sử dụng Neo4j để lưu trữ và truy vấn Đồ thị tri thức.
  - Sử dụng MongoDB cho lưu trữ văn bản và dữ liệu phi cấu trúc.
  - Sử dụng Apache Kafka cho luồng dữ liệu thời gian thực.
  - Sử dụng Gemini (Google) làm LLM nền tảng cho các tác vụ xử lý ngôn ngữ.
- **Bài toán đầu ra:** Hệ thống đưa ra dự báo xu hướng giá (Trend Prediction) theo 3 nhãn: Tăng, Giảm, Đi ngang (Sideway) trong ngắn hạn, kèm theo văn bản giải thích lý do.

1.4 Bố cục báo cáo

Báo cáo được tổ chức thành 6 chương với nội dung cụ thể như sau:

- **Chương 1: Giới thiệu.** Trình bày bối cảnh nghiên cứu, xác định vấn đề, mục tiêu và phạm vi của đề tài dự báo xu hướng giá cổ phiếu.
- **Chương 2: Cơ sở lý thuyết và Công nghệ.** Hệ thống hóa các kiến thức nền tảng về Big Data, Graph RAG, khuôn mẫu TRR và các công nghệ lõi được áp dụng.
- **Chương 3: Thiết kế hệ thống.** Mô tả chi tiết kiến trúc tổng thể, thiết kế cơ sở dữ liệu đa mô hình, quy trình đường ống dữ liệu và logic thuật toán dự báo.

- **Chương 4: Cài đặt và Triển khai.** Trình bày quá trình hiện thực hóa các module chức năng, cấu hình môi trường và kịch bản triển khai hệ thống.
- **Chương 5: Thử nghiệm và Đánh giá.** Phân tích dữ liệu thực nghiệm, đánh giá hiệu năng hệ thống và độ chính xác của mô hình thông qua các chỉ số đo lường cụ thể.
- **Chương 6: Tổng kết và Hướng phát triển.** Tóm tắt các kết quả đạt được, nhìn nhận hạn chế và đề xuất các hướng cải tiến trong tương lai.

Chương 2

Cơ sở lý thuyết và Công nghệ

Sự phát triển bùng nổ của thị trường tài chính kỹ thuật số đã tạo ra một dòng chảy thông tin khổng lồ và hỗn loạn, nơi các mô hình định lượng truyền thống dần bộc lộ giới hạn khi chỉ dựa vào các chuỗi số liệu lịch sử mà bỏ qua ngữ nghĩa của các sự kiện kinh tế. Thách thức lớn nhất trong bài toán dự báo hiện đại không còn nằm ở việc thiếu dữ liệu, mà là khả năng chuyển hóa dữ liệu phi cấu trúc thành tri thức có khả năng suy luận nhân quả. Để giải quyết vấn đề này, nền tảng lý thuyết của hệ thống được xây dựng dựa trên sự hội tụ của ba trụ cột công nghệ tiên tiến: kiến trúc xử lý dữ liệu lớn hiệu năng cao, phương pháp biểu diễn tri thức dạng đồ thị (Graph Representation) và tư duy lý luận nhân quả theo thời gian (Temporal Relational Reasoning). Sự kết hợp này đánh dấu bước chuyển dịch quan trọng từ việc "khai phá dữ liệu" thuần túy sang "nhận thức thị trường" dựa trên ngữ cảnh sâu sắc.

2.1 Tổng quan về Big Data trong Tài chính

Dữ liệu tài chính hiện đại mang đầy đủ các đặc trưng của Big Data (3Vs: Volume, Velocity, Variety). Để xử lý hiệu quả, hệ thống cần một kiến trúc đường ống dữ liệu (Data Pipeline) mạnh mẽ, kết hợp giữa lưu trữ và xử lý tốc độ cao.

2.1.1 Quy trình ETL (Extract - Transform - Load)

ETL là xương sống của mọi hệ thống dữ liệu lớn, đảm bảo dữ liệu thô được chuyển hóa thành tri thức có thể sử dụng:

- **Trích xuất (Extract):** Hệ thống thu thập dữ liệu từ các nguồn không đồng nhất. Đối với dữ liệu phi cấu trúc (Unstructured Data), hệ thống sử dụng kỹ thuật Web Scraping để lấy tin tức từ các trang báo điện tử và mạng xã hội. Đối với dữ liệu có cấu trúc (Structured Data), API chứng khoán được gọi liên tục để lấy dữ liệu giao dịch.
- **Chuyển đổi (Transform):** Đây là bước quan trọng nhất để chuẩn hóa dữ liệu. Các tác vụ bao gồm: làm sạch mã HTML, chuẩn hóa định dạng ngày tháng, và đặc biệt là tiền xử lý văn bản (Text Preprocessing) để phục vụ cho các mô hình NLP sau này.
- **Nạp (Load):** Dữ liệu sạch được lưu trữ vào hệ quản trị cơ sở dữ liệu đa mô hình: MongoDB (Document-based) cho lưu trữ văn bản linh hoạt và Neo4j (Graph-based) cho lưu trữ các mối quan hệ phức tạp.

2.1.2 Kiến trúc Xử lý Hỗn hợp (Hybrid Processing Architecture)

Hệ thống giải quyết bài toán độ trễ và khối lượng dữ liệu bằng cách kết hợp hai luồng xử lý:

- **Batch Processing (Xử lý theo lô):** Được áp dụng cho dữ liệu tin tức và xây dựng đồ thị tri thức. Do việc suy luận của LLM tốn kém tài nguyên và thời gian, quá trình trích xuất thực thể và xây dựng đồ thị quan hệ được thực hiện định kỳ (ví dụ: cuối ngày giao dịch) để tạo ra các báo cáo phân tích sâu.
- **Real-time Processing (Xử lý thời gian thực):** Được áp dụng cho dữ liệu giá và chỉ số thị trường. Sử dụng Apache Kafka làm hàng đợi thông điệp (Message Queue), hệ thống đảm bảo dữ

liệu giá khớp lệnh được truyền tải tức thì (Streaming) từ nguồn đến Dashboard người dùng với độ trễ thấp nhất.

2.2 Graph RAG và Đồ thị Tri thức (Knowledge Graph)

2.2.1 Đồ thị Tri thức trong Tài chính

Khác với các cơ sở dữ liệu quan hệ truyền thống vốn yêu cầu cấu trúc bảng cố định, Đồ thị Tri thức (Knowledge Graph – KG) biểu diễn dữ liệu dưới dạng mạng lưới các đỉnh (Nodes) và cạnh (Edges) cho phép mô tả linh hoạt các mối liên kết phức tạp trong hệ sinh thái tài chính. Thay vì chỉ lưu trữ thông tin dưới dạng số liệu rời rạc, KG nhấn mạnh vào *cấu trúc quan hệ* và *ngữ nghĩa kết nối*, yếu tố đặc biệt quan trọng khi phân tích các tương tác động giữa thị trường, doanh nghiệp và yếu tố vĩ mô.

- **Nodes (Thực thể tài chính):** Mỗi node biểu diễn một đối tượng có ý nghĩa kinh tế, chẳng hạn như:
  - Mã cổ phiếu (ví dụ: `VNM`, `HPG`, `VIC`),
  - Doanh nghiệp (Vinamilk, Hòa Phát, Vingroup),
  - Ngành hoặc chuỗi giá trị (Thép, Bất động sản, Năng lượng),
  - Chỉ số tài chính (Lãi suất, Tỷ giá USD/VND, CPI),
  - Sự kiện vĩ mô ("Lạm phát tăng cao", "Xung đột tại Biển Đỏ", "Giá dầu tăng đột biến").
- **Edges (Quan hệ kinh tế – tài chính):** Mỗi cạnh biểu diễn một mối quan hệ có ngữ nghĩa rõ ràng, ví dụ:
  - Quan hệ nhân quả: (Lãi suất) $\xrightarrow[\text{Tăng}]{}$ (Chi phí vay vốn)
  - Quan hệ sở hữu: (Vingroup) $\xrightarrow[\text{Sở hữu}]{}$ (Vinhomes)
  - Quan hệ cung – cầu: (Giá dầu) $\xrightarrow[\text{Tăng}]{}$ (Chi phí vận tải)
  - Quan hệ ngành: (Hòa Phát) $\xrightarrow[\text{Thuộc}]{}$ (Ngành Thép)

Ví dụ minh họa: Giả sử có một sự kiện vĩ mô: "Giá quặng sắt thế giới giảm mạnh". Sự kiện này có thể được biểu diễn trong KG như sau:

1. Node sự kiện: `Giá quặng sắt giảm`.
2. Cạnh tác động: (Giá quặng sắt giảm) $\xrightarrow[\text{Giảm chi phí đầu vào}]{}$ (Hòa Phát).
3. Quan hệ ngành: (Hòa Phát) $\xrightarrow[\text{Thuộc}]{}$ (Ngành Thép).
4. Tác động lan truyền đến các doanh nghiệp cùng ngành: (Ngành Thép) $\xrightarrow[\text{Ảnh hưởng}]{}$ (Hoa Sen Group).

Chuỗi liên kết này giúp hệ thống không chỉ xác định tác động trực tiếp lên doanh nghiệp sản xuất thép lớn như Hòa Phát, mà còn suy ra ảnh hưởng gián tiếp lên các doanh nghiệp trong cùng chuỗi cung ứng hoặc cùng ngành. Điểm mạnh của KG nằm ở khả năng mô hình hóa những đường dẫn tác động dạng *multi-hop* như trên, điều mà các mô hình chuỗi thời gian truyền thống không thể giải quyết vì thiếu thông tin về cấu trúc quan hệ.

Khả năng mô hình hóa lan truyền rủi ro: Tính chất liên kết của KG đặc biệt hữu ích trong việc phân tích *risk contagion* trong thị trường tài chính. Ví dụ, một sự kiện bất lợi như "Lãi suất USD tiếp tục tăng" có thể lan truyền qua mạng lưới KG:

Lãi suất USD tăng $\rightarrow$ Áp lực tỷ giá USD/VND
$\rightarrow$ Chi phí nhập khẩu tăng
$\rightarrow$ Biên lợi nhuận giảm của doanh nghiệp nhập khẩu
$\rightarrow$ Giá cổ phiếu nhóm ngành bán lẻ giảm.

Nhờ cấu trúc liên kết này, hệ thống dự đoán không chỉ dừng lại ở việc nhận diện tác động trực tiếp mà còn nắm bắt được chuỗi lan truyền đến các thực thể liên quan, từ đó tạo nên nền tảng dữ liệu mạnh để phục vụ cho các mô hình dự báo dựa trên Graph RAG hoặc các phương pháp học máy khác.

2.2.2 Graph RAG (Retrieval-Augmented Generation on Graphs)

Graph RAG (Retrieval-Augmented Generation on Graphs) là một bước tiến quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, được thiết kế để khắc phục các hạn chế cố hữu của phương pháp RAG dựa trên vector truyền thống (Vector-based RAG). Trong khi RAG truyền thống coi dữ liệu như các đoạn văn bản rời rạc và truy xuất dựa trên độ tương đồng ngữ nghĩa (semantic similarity), Graph RAG tiếp cận dữ liệu dưới góc nhìn **cấu trúc và mối liên kết (Structure and Connectivity)**.

Sự ưu việt và cơ chế hoạt động của Graph RAG được phản ánh thông qua các khía cạnh sau:

- **Khắc phục điểm mù của Truy xuất Vector:** Các mô hình nhúng (Embedding models) thường gom nhóm văn bản có đặc trưng từ vựng giống nhau lại gần nhau trong không gian vector. Tuy nhiên, trong tài chính, nhiều sự kiện có quan hệ nhân quả chặt chẽ dù không chia sẻ từ khóa chung (ví dụ: "Hạn hán tại Brazil" và "Giá cổ phiếu công ty cà phê Việt Nam"). Graph RAG giải quyết vấn đề này bằng cách khai thác các cạnh quan hệ (Edges) trong đồ thị tri thức, không tìm kiếm sự tương đồng bề mặt mà hướng đến **sự liên quan dẫn xuất (derived relevance)** thông qua các kết nối thực tế đã được định nghĩa.

- **Cơ chế Suy luận Đa bước (Multi-hop Reasoning):** Đây là tính năng quan trọng nhất giúp hệ thống thực hiện các tác vụ dự báo phức tạp. Thay vì chỉ truy xuất một bước (Single-hop retrieval), Graph RAG thực hiện quá trình duyệt đồ thị (Graph Traversal) bắt đầu từ các thực thể neo (anchor entities) trong câu truy vấn để khám phá các đường dẫn tri thức sâu hơn. Quy trình này mô phỏng tư duy liên tưởng:

  1. *(Sự kiện nguồn):* Phát hiện sự kiện "Căng thẳng địa chính trị tại Biển Đỏ".
  2. *(Tác động trung gian – Hop 1):* Truy vết theo cạnh quan hệ để thấy tác động đến "Chi phí vận tải biển".
  3. *(Hệ quả tài chính – Hop 2):* Tiếp tục truy vết để thấy ảnh hưởng đến "Biên lợi nhuận doanh nghiệp xuất khẩu".
  4. *(Thực thể đích – Hop 3):* Kết luận tác động đến mã cổ phiếu cụ thể (ví dụ: HPG, VHC).

  Các phương pháp truy xuất thông thường khó kết nối trực tiếp giữa Bước 1 và Bước 4 nếu thiếu các bước trung gian, trong khi Graph RAG có thể tự động xâu chuỗi toàn bộ lộ trình này.

- **Trích xuất Đồ thị Con (Sub-graph Extraction) làm ngữ cảnh:** Thay vì đưa vào LLM hàng loạt đoạn văn bản (chunks) tiềm ẩn trùng lặp hoặc mâu thuẫn, Graph RAG trích xuất một đồ thị con gồm các thực thể và quan hệ phù hợp nhất với truy vấn. Dữ liệu đưa vào LLM lúc này là các cấu trúc dạng bộ ba (Triples/Tuples) (Chủ thể, Quan hệ, Đối tượng). Cách biểu diễn này giúp *neo giữ* (grounding) câu trả lời vào các sự kiện thực tế, giảm đáng kể hiện tượng "ảo giác" (hallucination) vốn phổ biến khi LLM phải xử lý văn bản tài chính dài và phức tạp.

- **Khả năng giải thích (Explainability):** Một lợi thế lớn của Graph RAG so với các mô hình dạng hộp đen (Black-box models) nằm ở khả năng minh bạch hóa quá trình suy luận. Khi đưa ra dự báo, hệ thống có thể trích dẫn chính xác đường dẫn (path) trong đồ thị mà nó đã đi qua để đi đến kết luận. Điều này mang lại cho người dùng không chỉ kết quả mà còn cả *luận cứ logic* (rationale), yếu tố then chốt trong hỗ trợ ra quyết định đầu tư.

2.3 Temporal Relational Reasoning (TRR)

Temporal Relational Reasoning (TRR) là cơ sở lý thuyết trọng tâm của mô hình, mô phỏng cách con người xử lý thông tin khi đối mặt với các vấn đề kinh tế phức tạp: tiếp nhận thông tin mới, liên hệ với kiến thức trong quá khứ, chọn lọc các yếu tố quan trọng và cuối cùng đưa ra suy luận. Thay vì xem dữ liệu tin tức như từng dòng văn bản độc lập, TRR chuyển hóa chúng thành các chuỗi tác động được liên kết theo thời gian, hình thành một đồ thị nhân quả sáng tỏ và dễ suy luận.

(1) Giai đoạn Tạo sinh Đồ thị (Brainstorming) Khi một bài báo hoặc tin tức mới xuất hiện, hệ thống sử dụng LLM để phân tích và sinh ra các tác động dây chuyền (knock-on effects) xuất phát từ nội dung ban đầu. Quá trình này được mô hình hóa như việc xây dựng một đồ thị có hướng $G = (Z, A)$, trong đó tập đỉnh Z bao gồm bài báo gốc x_j, các thực thể trung gian e_h và cuối cùng là các cổ phiếu mục tiêu s_n.

Mỗi tác động mới được sinh ra thông qua một phân phối có điều kiện:

$$[z^1, \ldots, z^k] \sim p_\theta^{brainstorm}(z_{i+1}^{(1\ldots k)} \mid z_i)$$

Quá trình được lặp lại cho đến khi chuỗi tác động đi đến một cổ phiếu thuộc danh mục theo dõi hoặc đạt số vòng lặp tối đa.

Ví dụ: Từ tin tức "Giá dầu tăng mạnh", hệ thống có thể sinh ra chuỗi:

Giá dầu tăng → Chi phí vận tải tăng → Biên lợi nhuận giảm → Cổ phiếu bán lẻ giảm.

(2) Giai đoạn Tích hợp Ngữ cảnh Thời gian (Memory) Một sự kiện tài chính không bao giờ tồn tại độc lập. Do đó, hệ thống duy trì một bộ nhớ $M = \{M_{e_1}, M_{e_2}, \ldots\}$ chứa lại các chuỗi tác động đã từng xảy ra trong quá khứ cho từng thực thể. Khi một thực thể xuất hiện trở lại trong tin tức mới, TRR truy xuất toàn bộ các chuỗi từng liên quan đến thực thể đó và hợp nhất chúng vào đồ thị hiện tại, tạo nên đồ thị ngữ cảnh hóa theo thời gian $G_{temporal}$.

Cơ chế suy giảm theo thời gian (Time Decay). Không phải sự kiện nào cũng giữ nguyên mức độ ảnh hưởng. TRR mô hình hóa điều này bằng hàm suy giảm mũ:

$$R_{u,v} = \exp\left(-\frac{t_{u,v}}{\lambda}\right)$$

Trong đó $t_{u,v}$ là thời gian đã trôi qua và λ là hệ số quyết định tốc độ quên.

Ví dụ: - Một tin "Lãi suất tăng" xuất hiện cách đây 3 ngày có thể vẫn quan trọng. - Nhưng một tin "Giá gas tăng" từ 3 tháng trước sẽ gần như không được hệ thống xem xét.

(3) Giai đoạn Chú ý (Attention)

Đồ thị $G_{temporal}$ sau khi hợp nhất thường rất lớn, bao gồm cả nhiều đường dẫn ít liên quan. Để xác định những thực thể then chốt nhất, TRR áp dụng cơ chế xếp hạng tương tự PageRank nhưng có trọng số thời gian. Điểm xếp hạng của một thực thể e_h được tính như sau:

$$PR(e_h) = \sum_{b \in B_{e_h}} \frac{PR(b)}{L_b} \cdot R_{b,e_h}$$

Trong đó:

- $PR(e_h)$: Điểm xếp hạng (Ranking Score) của thực thể e_h, phản ánh mức độ quan trọng của nó trong đồ thị.
- B_{e_h}: Tập hợp các thực thể "cha" có liên kết trực tiếp tác động đến e_h.
- $PR(b)$: Điểm xếp hạng hiện tại của thực thể cha b.
- L_b: Số lượng liên kết ra (out-degree) từ thực thể b, dùng để chuẩn hóa ảnh hưởng của b đến các thực thể con.
- R_{b,e_h}: Trọng số thời gian được tính từ giai đoạn Memory, mô phỏng mức độ ảnh hưởng còn lại của liên kết dựa trên khoảng thời gian đã trôi qua (Time Decay).

Công thức này giúp hệ thống tập trung vào các thực thể quan trọng nhất, đồng thời giảm tác động của thông tin cũ hoặc ít liên quan, từ đó tạo ra đồ thị tinh gọn G_{TRR} phục vụ cho bước suy luận.

Ví dụ: Trong bối cảnh "Căng thẳng địa chính trị", các thực thể như *Giá dầu*, *Chi phí vận tải*, *Tỷ giá USD* có thể được xếp hạng rất cao, trong khi những liên kết yếu như "Nhu cầu tiêu dùng cà phê" bị loại bỏ.

(4) Giai đoạn Suy luận (Reasoning)

Đây là bước cuối cùng và quan trọng nhất, nơi hệ thống tổng hợp các tín hiệu đã được lọc để đưa ra dự báo. Thay vì nạp vào LLM một bài báo dài dòng chứa nhiều thông tin nhiễu (như quảng cáo, văn phong rườm rà), TRR chuyển đổi đồ thị tinh gọn G_{TRR} thành một danh sách các **bộ ba quan hệ có cấu trúc (structured relational tuples)**.

Mỗi cạnh trong đồ thị được biểu diễn dưới dạng một tuple:

$$(t, z_s, a, z_o)$$

Trong đó:

- t: Thời gian diễn ra sự kiện (Time).
- z_s: Chủ thể (Subject - Nguồn tác động).
- a: Hành động hoặc mối quan hệ (Action/Relation).
- z_o: Đối tượng (Object - Bên chịu tác động).

Ví dụ minh họa: Giả sử đồ thị đã lọc ra được chuỗi tác động tiêu cực đến cổ phiếu HPG (Hòa Phát). Thay vì đọc văn bản gốc, LLM sẽ nhận được chuỗi tuples logic như sau:

- Tuple 1: `(2024-11-20, Căng thẳng Biển Đỏ, gây ra, Gián đoạn vận tải biển)`
- Tuple 2: `(2024-11-20, Gián đoạn vận tải biển, làm tăng, Chi phí logistics toàn cầu)`
- Tuple 3: `(2024-11-20, Chi phí logistics, tác động tiêu cực, Biên lợi nhuận ngành thép)`
- Tuple 4: `(2024-11-20, Ngành thép, bao gồm, Cổ phiếu HPG)`

Từ danh sách tuples ngắn gọn và giàu ngữ nghĩa này, LLM thực hiện suy luận để sinh ra kết quả dự báo $\hat{y}$ (ví dụ: Xu hướng GIẢM):

$$\hat{y} \sim p_\theta^{reason}(\hat{y} \mid P, G_{TRR})$$

Ưu điểm của phương pháp:

1. **Giảm nhiễu (Noise Reduction):** LLM không bị phân tâm bởi các câu văn thừa thãi, chỉ tập trung vào logic nhân quả cốt lõi.
2. **Tính minh bạch (Explainability):** Hệ thống có thể trích xuất chính xác các tuples nào đã dẫn đến kết luận dự báo. Ví dụ: "Dự báo GIẢM vì Chi phí logistics tăng tác động lên lợi nhuận". Điều này khắc phục hoàn toàn nhược điểm "hộp đen" của các mô hình Deep Learning truyền thống.

Tóm lại, TRR có thể được hiểu như một "bộ máy suy luận nhân quả theo thời gian", nơi tin tức mới được đặt vào bối cảnh lịch sử, các yếu tố quan trọng được chọn lọc và LLM đưa ra dự báo dựa trên logic nhân quả đã được chưng cất. Cách tiếp cận này mô phỏng rất sát quá trình con người đưa ra quyết định đầu tư trong thế giới thực.

2.4 Mô hình Ngôn ngữ lớn (Large Language Models - LLM)

Trong kiến trúc hệ thống đề xuất, Mô hình Ngôn ngữ lớn (như Google Gemini) không chỉ đóng vai trò là công cụ xử lý văn bản đơn thuần mà hoạt động như một **trung tâm điều phối nhận thức (Cognitive Reasoning Engine)**. Khác với các mô hình học sâu truyền thống (như LSTM hay Bi-GRU) vốn phụ thuộc hoàn toàn vào việc học thuộc lòng các mẫu dữ liệu quá khứ, LLM mang lại khả năng tư duy linh hoạt để giải quyết các bài toán tài chính phức tạp.

Vai trò và năng lực cốt lõi của LLM trong hệ thống được thể hiện qua ba khía cạnh chính:

1. Khả năng Suy luận Zero-shot (Zero-shot Reasoning) đối với sự kiện "Thiên nga đen"

Thị trường tài chính thường xuyên chịu tác động bởi các sự kiện chưa từng có tiền lệ (Unprecedented Events) hay còn gọi là sự kiện "Thiên nga đen" (Black Swans), ví dụ như đại dịch COVID-19 hay các cuộc khủng hoảng địa chính trị bất ngờ.

- *Hạn chế của mô hình cũ:* Các mô hình truyền thống yêu cầu dữ liệu huấn luyện lịch sử để nhận diện mẫu hình. Do đó, chúng thường thất bại hoặc đưa ra dự báo sai lệch khi đối mặt với các sự kiện mới lạ chưa từng xuất hiện trong tập huấn luyện.
- *Ưu việt của LLM:* Nhờ được huấn luyện trên kho dữ liệu tri thức khổng lồ của nhân loại, LLM có khả năng thực hiện suy luận ngay lập tức trên các sự kiện mới mà không cần huấn luyện lại (Zero-shot). Ví dụ, dù chưa từng "thấy" một đại dịch tương tự trong dữ liệu tài chính quá khứ, LLM vẫn có thể suy luận logic rằng: "Phong tỏa xã hội" $\rightarrow$ "Gián đoạn chuỗi cung ứng" $\rightarrow$ "Tiêu cực cho nhóm ngành sản xuất".

2. Khả năng Tổng quát hóa Tri thức (Knowledge Generalization)

LLM sở hữu khả năng tổng quát hóa vượt trội, cho phép nó áp dụng các tri thức phổ quát (world knowledge) vào ngữ cảnh tài chính cụ thể.

- *Ứng dụng trong Giai đoạn Brainstorming:* Trong bước tạo sinh đồ thị, hệ thống tận dụng khả năng này để phát hiện các mối liên kết ẩn. Ví dụ, khi đọc tin tức về "Giá thức ăn chăn nuôi tăng", LLM có thể tự động liên kết (generalize) sự kiện này đến "Biên lợi nhuận của doanh nghiệp chế biến thịt (như MSN, DBC)" mà không cần văn bản phải đề cập trực tiếp đến tên các doanh nghiệp đó. Điều này giúp hệ thống xây dựng được một Đồ thị tri thức dày đặc và giàu ý nghĩa hơn so với các phương pháp trích xuất từ khóa thông thường.

3. Khả năng Xử lý Đa nhiệm trong Pipeline (Multitask Processing)

Hệ thống tận dụng tính linh hoạt của LLM để thực hiện hai nhiệm vụ riêng biệt nhưng bổ trợ cho nhau:

- **Tác vụ Trích xuất (Extraction Task):** Ở đầu vào, LLM hoạt động như một bộ phân tích cú pháp ngữ nghĩa (semantic parser), chuyển đổi các bản tin phi cấu trúc thành các thực thể và quan hệ có cấu trúc để xây dựng đồ thị.
- **Tác vụ Dự báo (Prediction Task):** Ở đầu ra, LLM hoạt động như một chuyên gia phân tích. Nó nhận đầu vào là các chuỗi sự kiện đã được lọc (tuples) từ thuật toán TRR, tổng hợp các vector tác động trái chiều (tích cực/tiêu cực) và đưa ra kết luận cuối cùng về xu hướng giá kèm theo lời giải thích minh bạch.

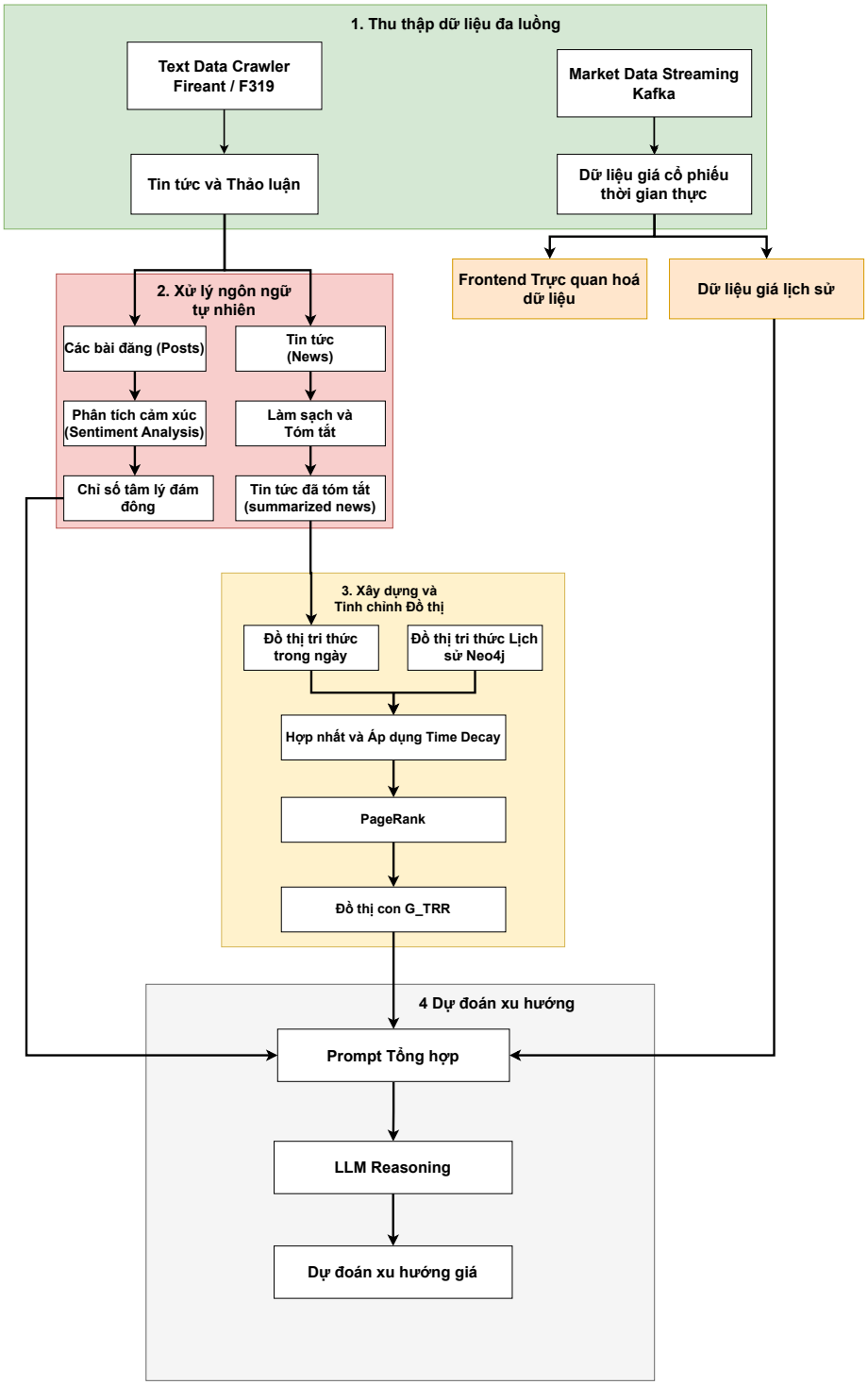
Chương 3

Thiết kế Hệ thống

Chương 3 tập trung vào việc hiện thực hóa các cơ sở lý thuyết đã trình bày ở chương trước thành một kiến trúc hệ thống Big Data hoàn chỉnh. Nội dung chương mô tả chi tiết thiết kế kỹ thuật của hệ thống dự báo xu hướng giá cổ phiếu, bao gồm: kiến trúc tổng thể, quy trình xử lý dữ liệu (Data Pipeline), thiết kế cơ sở dữ liệu đa mô hình và cơ chế hoạt động của các module cốt lõi. Hệ thống được xây dựng dựa trên nguyên tắc mô đun hóa, đảm bảo khả năng mở rộng, tính sẵn sàng cao và khả năng xử lý dữ liệu thời gian thực.

3.1 Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình kiến trúc đường ống dữ liệu (Data Pipeline Architecture), cho phép xử lý song song các luồng dữ liệu phi cấu trúc (văn bản) và có cấu trúc (chuỗi thời gian). Dựa trên sơ đồ thiết kế, kiến trúc hệ thống được chia thành bốn phân hệ chính, kết nối chặt chẽ theo quy trình tuần tự:



Hình 3.1: Sơ đồ kiến trúc tổng thể hệ thống

Phân hệ 1: Thu thập dữ liệu đa luồng

Đây là tầng đầu vào của hệ thống, chịu trách nhiệm tiếp nhận dữ liệu thô từ các nguồn không đồng nhất để phục vụ các tầng xử lý phía sau:

- **Text Data Crawler:** Thu thập dữ liệu văn bản từ các nguồn tin tức tài chính (Fireant) và diễn đàn đầu tư (F319), bao gồm bài báo chính thống và bình luận của nhà đầu tư cá nhân.

- **Market Data Streaming:** Sử dụng Apache Kafka cho việc truyền tải dữ liệu giá cổ phiếu theo thời gian thực. Phân hệ này cung cấp hai luồng: một cho trực quan hóa trên Frontend và một cho mô hình dự đoán.

Phân hệ 2: Xử lý Ngôn ngữ tự nhiên (NLP)
Dữ liệu văn bản thô được đưa qua phân hệ này để làm sạch và trích xuất đặc trưng:

- **Xử lý bài đăng (Posts):** Phân tích cảm xúc (Sentiment Analysis) để lượng hóa thái độ của nhà đầu tư thành Chỉ số tâm lý đám đông (Crowd Psychology Index).
- **Xử lý tin tức (News):** Làm sạch và tóm tắt (Summarization) các bản tin tài chính dài để giữ lại thông tin cốt lõi về sự kiện và thực thể kinh tế.

Phân hệ 3: Xây dựng và Tinh chỉnh Đồ thị
Đây là phân hệ trung tâm, ứng dụng thuật toán TRR, chuyển dữ liệu văn bản thành cấu trúc tri thức có khả năng suy luận:

- **Xây dựng Đồ thị tri thức:** Trích xuất các thực thể và quan hệ từ bản tin tóm tắt để tạo Đồ thị tri thức trong ngày (Daily Knowledge Graph).
- **Hợp nhất và Time Decay:** Hợp nhất đồ thị trong ngày với Đồ thị tri thức lịch sử (Neo4j) và áp dụng cơ chế Time Decay để giảm trọng số các thông tin cũ.
- **Lọc thông tin (PageRank):** Sử dụng PageRank để xếp hạng và lọc các nút quan trọng, tạo đồ thị con G_{TRR} tinh gọn với các chuỗi nhân quả có giá trị dự báo cao.

Phân hệ 4: Dự đoán xu hướng
Phân hệ cuối cùng tổng hợp dữ liệu đa nguồn để đưa ra kết quả dự báo:

- **Prompt Tổng hợp:** Tạo prompt cấu trúc gồm: (1) Chỉ số tâm lý đám đông từ Phân hệ 2, (2) Đồ thị con G_{TRR} từ Phân hệ 3, và (3) Dữ liệu giá lịch sử từ Phân hệ 1.
- **LLM Reasoning:** Mô hình ngôn ngữ lớn phân tích các yếu tố vĩ mô, tâm lý và kỹ thuật để đưa ra dự đoán xu hướng giá (Tăng/Giảm/Đi ngang) kèm giải thích chi tiết.

3.2 Thiết kế Cơ sở dữ liệu

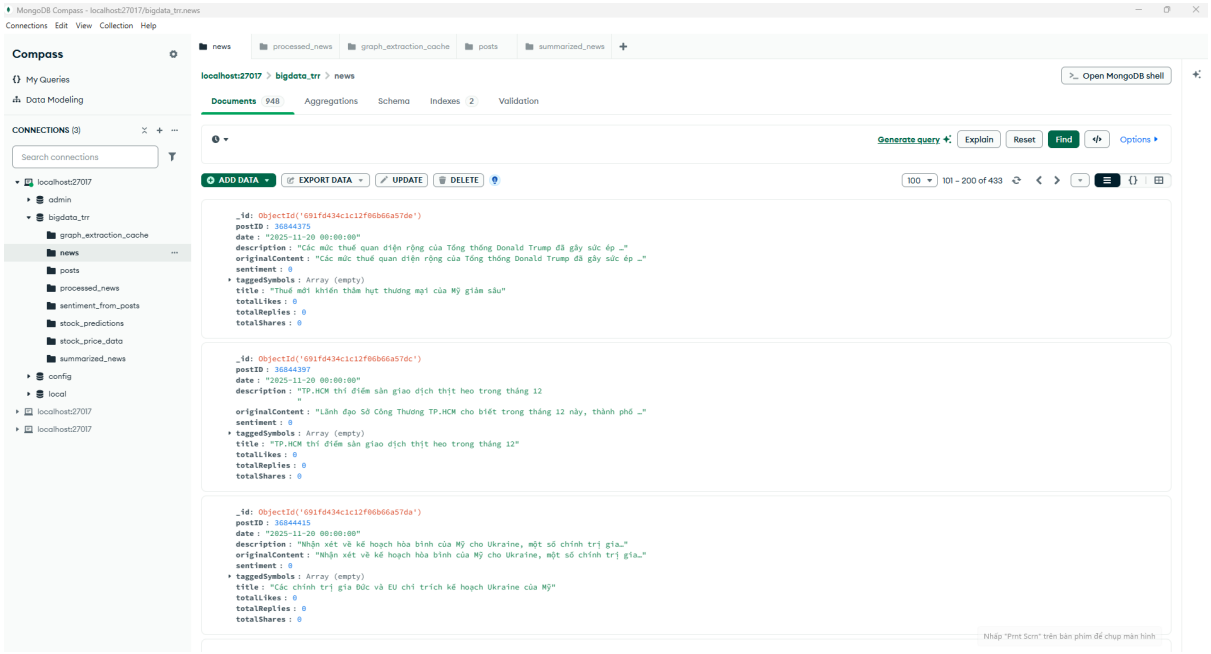
Để đáp ứng yêu cầu vừa lưu trữ khối lượng lớn dữ liệu văn bản phi cấu trúc, vừa truy vấn hiệu quả các mối quan hệ ngữ nghĩa phức tạp, hệ thống áp dụng chiến lược **Polyglot Persistence** (Đa bền vững). Kiến trúc này kết hợp sức mạnh của Cơ sở dữ liệu hướng văn bản (Document-oriented Database) và Cơ sở dữ liệu đồ thị (Graph Database), đảm bảo cấu trúc dữ liệu phục vụ tối ưu cho các giải thuật khai phá tri thức (Data Mining) và suy luận nhân quả.

3.2.1 Cơ sở dữ liệu Tài liệu (MongoDB - Storage Layer)

MongoDB đóng vai trò là kho dữ liệu chính (Operational Datastore), chịu trách nhiệm lưu trữ dữ liệu thô sau khi thu thập, dữ liệu đã qua xử lý NLP và kết quả dự báo. Việc lựa chọn MongoDB dựa trên khả năng mở rộng linh hoạt (Scalability) và schema động (Dynamic Schema), phù hợp với tính chất thay đổi liên tục của dữ liệu tin tức và mạng xã hội.
Các Collections chính bao gồm:

- **Collection** `summarized_news`**:** Lưu trữ tri thức đã được tinh gọn.
  - *Schema:* `_id`, `date`, `title`, `summary`, `stockCodes`, `originalContent`.
- **Collection** `sentiment_from_posts`**:** Lưu trữ dữ liệu chuỗi thời gian về tâm lý đám đông.
  - *Schema:* `date`, `taggedSymbols`, `positive_posts`, `negative_posts`, `neutral_posts`.

- **Collection `stock_price_data`:** Lưu trữ dữ liệu giao dịch lịch sử và thời gian thực.
  - *Schema:* `symbol`, `time`, `open`, `high`, `low`, `close`, `volume`.
- **Collection `stock_predictions`:** Lưu trữ kết quả đầu ra cuối cùng của hệ thống.
  - *Schema:* `date`, `symbol`, `trend`, `confidence`, `reasoning`, `full_analysis`.



Hình 3.2: Giao diện MongoDB Compass

3.2.2 Cơ sở dữ liệu Đồ thị (Neo4j - Reasoning Layer)

Neo4j được sử dụng làm lớp ngữ nghĩa (Semantic Layer), mô hình hóa sự lan truyền tác động giữa các thực thể kinh tế. Khác với MongoDB lưu trữ dữ liệu rời rạc, Neo4j lưu trữ "sự kết nối", cho phép thực hiện các truy vấn phức tạp như tìm đường đi ngắn nhất hay lan truyền điểm ảnh hưởng (PageRank).

Mô hình đồ thị (Graph Model) được thiết kế gồm các thành phần sau:

1. Các loại Đỉnh (Nodes):

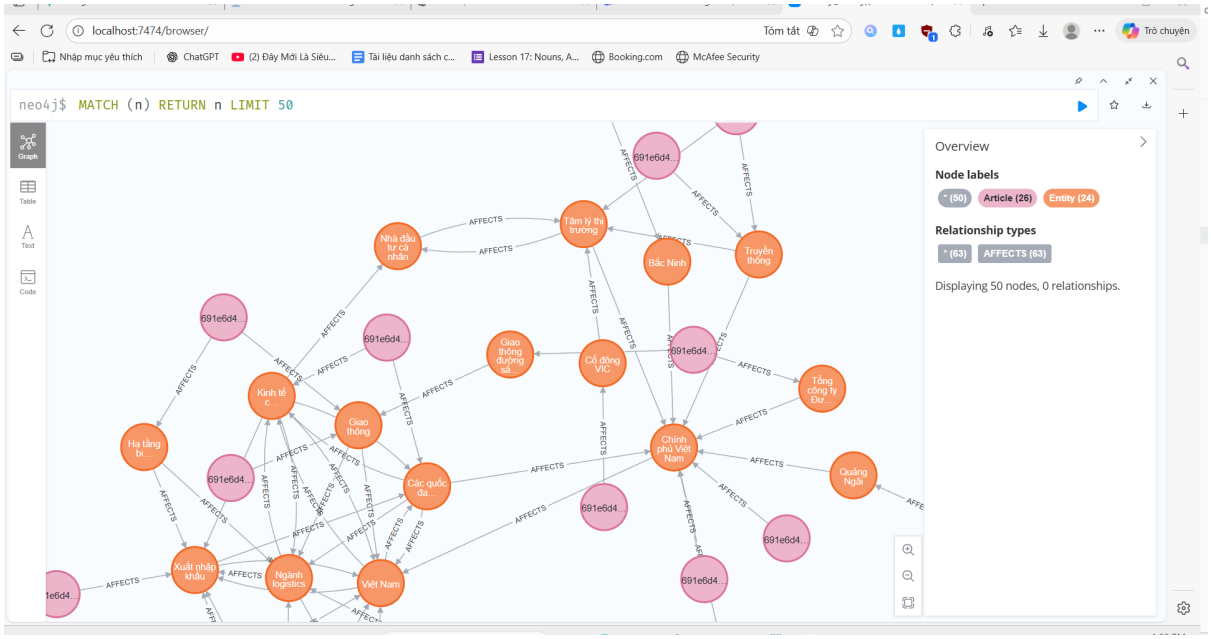
- **Label: Article (Bài báo):** Đại diện cho nguồn phát tin.
  - *Thuộc tính:* `title`, `date`.
- **Label: Entity (Thực thể trung gian):** Đại diện cho các sự kiện vĩ mô, ngành hàng hoặc yếu tố kinh tế (ví dụ: "Lạm phát", "Giá dầu", "Chính sách tiền tệ").
  - *Thuộc tính:* `name`, `type`.
- **Label: Stock (Cổ phiếu):** Đại diện cho mã chứng khoán mục tiêu.
  - *Thuộc tính:* `name`, `sector`.

2. Các loại Cạnh (Relationships):

- **Type: IMPACT** (Tác động)
- **Thuộc tính quan trọng trên cạnh:**
  - `impact`: Loại tác động (POSITIVE / NEGATIVE)
  - `description`: Mô tả văn bản về lý do tác động
  - `date`: Thời gian diễn ra tác động, dùng để tính trọng số Time Decay

3. Mô hình lan truyền (Propagation Pattern):

$$(Article) \xrightarrow{IMPACT} (Entity) \xrightarrow{IMPACT} (Stock)$$



Hình 3.3: Giao diện Neo4j Browser

3.3 Quy trình Xử lý Dữ liệu

Để đảm bảo tính chính xác của mô hình dự báo và độ trễ thấp của hệ thống giám sát, quy trình xử lý dữ liệu (Data Pipeline) được thiết kế tách biệt thành hai luồng hoạt động song song: luồng xử lý lô (Batch Pipeline) phục vụ cho việc phân tích tri thức sâu và luồng xử lý thời gian thực (Real-time Pipeline) phục vụ cho việc cập nhật dữ liệu thị trường tức thời.

1. Luồng Xử lý Tin tức và Tri thức (Batch Pipeline)

Đây là luồng xử lý chính của hệ thống, được điều phối bởi module trung tâm `main.py`. Quy trình này chạy định kỳ (hàng ngày hoặc theo phiên), chịu trách nhiệm chuyển đổi dữ liệu văn bản thô thành các cấu trúc đồ thị có khả năng suy luận.

- **Thu thập dữ liệu (Data Crawling):** Quy trình bắt đầu với các script thu thập chuyên dụng như `news_daily_crawl.py` và `posts_daily_crawl.py`. Hệ thống kết nối đến API của nhà cung cấp dữ liệu (Fireant) thông qua cơ chế xác thực Bearer Token. Thuật toán thu thập sử dụng cơ chế "Cửa sổ trượt" (Sliding Window) với tham số `offset` và `limit`, quét ngược từ thời điểm hiện tại về quá khứ cho đến khi gặp dữ liệu đã tồn tại trong cơ sở dữ liệu hoặc vượt quá mốc thời gian 'target day'. Cơ chế này đảm bảo không bỏ sót tin tức và tối ưu hóa băng thông mạng.

- **Tiền xử lý và Làm sạch (Preprocessing):** Dữ liệu thô dạng JSON sau khi thu thập được đưa qua module `data_preprocessing.py`. Tại đây, thư viện *BeautifulSoup* được sử dụng để loại bỏ các thẻ HTML, giải mã các ký tự đặc biệt (HTML entities) và lọc bỏ các đường dẫn rác. Các trường thời gian (timestamp) từ nhiều định dạng khác nhau được chuẩn hóa về định dạng chuẩn ISO 8601 (YYYY-MM-DD) để đảm bảo tính nhất quán khi truy vấn theo chuỗi thời gian.

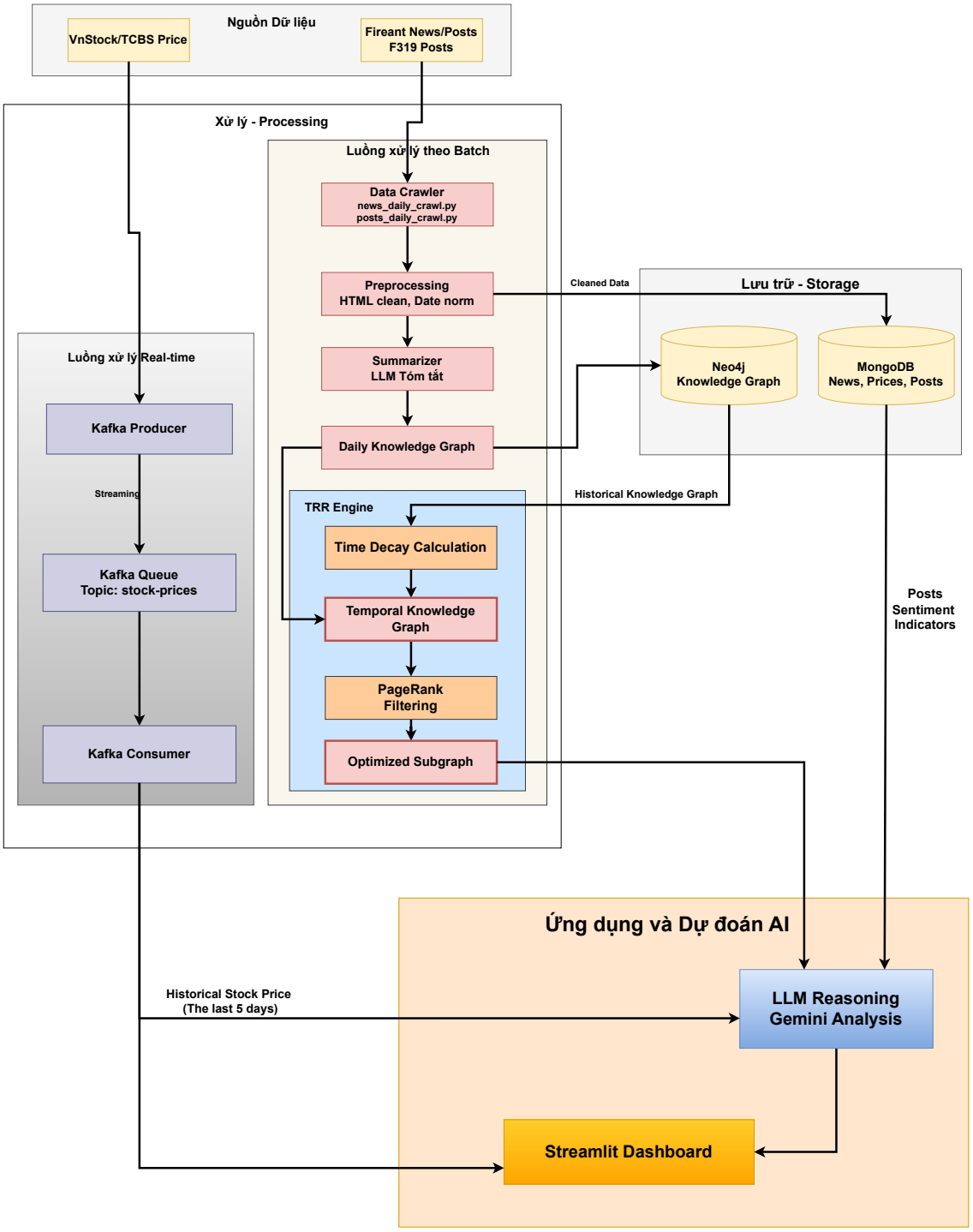
- **Tóm tắt và Trích xuất đặc trưng (Summarization & Extraction):** Trước khi đưa vào đồ thị, nội dung văn bản dài được đưa qua module `summarizer.py` để tóm tắt, giữ lại các ý chính. Tiếp đó, module `extractor.py` sử dụng LLM để thực hiện nhiệm vụ "Động não" (Brainstorming). Hệ thống gửi văn bản đã tóm tắt vào LLM và yêu cầu trích xuất các bộ ba tri thức (Triples) dưới dạng: *(Chủ thể, Quan hệ Tác động, Đối tượng)*. Quá trình này giúp chuyển hóa văn bản phi cấu trúc thành dữ liệu có cấu trúc.

- **Xây dựng và Tinh chỉnh Đồ thị (Graph Construction):** Các thực thể và quan hệ sau khi trích xuất được lưu trữ vào Neo4j thông qua module `graph_loader.py`. Tại bước này, hệ thống kích hoạt thuật toán TRR trong module `memory_attention.py`. Hệ thống truy vấn lại đồ thị lịch sử, áp dụng hàm suy giảm theo thời gian (Time Decay) để giảm trọng số của các thông tin cũ, sau đó chạy thuật toán PageRank để xếp hạng và lọc ra các nút thông tin quan trọng nhất, loại bỏ nhiễu trước khi đưa vào mô hình dự đoán.

2. Luồng Dữ liệu Thị trường (Real-time Pipeline)

Song song với luồng xử lý tri thức, hệ thống duy trì một đường ống dữ liệu tốc độ cao để phục vụ hiển thị Dashboard.

- **Streaming& Ingestion:** Một Producer liên tục lắng nghe dữ liệu khớp lệnh từ sàn giao dịch và đẩy các gói tin (messages) chứa thông tin giá, khối lượng vào Apache Kafka topic `stock-prices`.
- **Consuming & Visualization:** Tại tầng Frontend, một luồng background (Kafka Consumer) liên tục đọc dữ liệu từ topic này. Nhờ kiến trúc Pub/Sub của Kafka, độ trễ từ lúc khớp lệnh đến lúc hiển thị trên biểu đồ được giảm xuống dưới 1 giây, giúp nhà đầu tư quan sát được biến động thị trường ngay lập tức (Real-time).



Hình 3.4: Sơ đồ luồng xử lý dữ liệu

3.4 Thiết kế Module Dự đoán (Prediction Module)

Module dự đoán (được hiện thực hóa trong lớp `StockPredictor`) đóng vai trò là "bộ não" phân tích cuối cùng của hệ thống. Đây là nơi hội tụ của ba luồng dữ liệu: tri thức sự kiện (Knowledge Graph), tâm lý đám đông (Social Sentiment) và biến động kỹ thuật (Price History). Thay vì sử dụng các mô hình học máy truyền thống (như LSTM hay Random Forest) vốn hoạt động như một hộp đen, module

này tận dụng khả năng suy luận của LLM (Google Gemini) để mô phỏng tư duy của một chuyên gia tài chính, đưa ra dự báo dựa trên sự tổng hợp đa chiều.

Cơ chế hoạt động của module tuân theo quy trình tuần tự 4 bước như sau:

1. Truy xuất Ngữ cảnh (Context Retrieval): Quá trình dự báo bắt đầu khi hệ thống nhận được yêu cầu phân tích cho một mã cổ phiếu mục tiêu (ví dụ: HPG) tại một thời điểm cụ thể. Module dự đoán gọi đến bộ nhớ TRR (`TRRMemoryAttention`) để trích xuất một đồ thị con (Subgraph) bao gồm các sự kiện và thực thể có liên quan mật thiết nhất đến cổ phiếu đó trong vòng 30 ngày gần nhất. Kết quả đầu ra là một chuỗi các sự kiện nhân quả đã được lọc nhiễu, cung cấp bối cảnh vĩ mô cho AI.

2. Tổng hợp Tín hiệu Tâm lý (Sentiment Aggregation): Song song với việc lấy ngữ cảnh đồ thị, hệ thống truy vấn cơ sở dữ liệu MongoDB (`sentiment_from_posts`) để lấy dữ liệu cảm xúc cộng đồng. Module tổng hợp số lượng bài viết Tích cực, Tiêu cực và Trung lập liên quan đến mã cổ phiếu. Dữ liệu này đóng vai trò là "nhiệt kế" đo lường sự hưng phấn hoặc sợ hãi của thị trường, yếu tố thường đi trước biến động giá trong ngắn hạn.

3. Phân tích Xu hướng Kỹ thuật (Price Trend Analysis): Để đảm bảo dự báo không xa rời thực tế giao dịch, hệ thống trích xuất dữ liệu giá đóng cửa (Close Price) của 5 phiên giao dịch gần nhất từ MongoDB. Chuỗi dữ liệu này cung cấp cho LLM cái nhìn về động lượng (momentum) hiện tại của cổ phiếu, giúp mô hình nhận diện được xu hướng kỹ thuật đang diễn ra (ví dụ: đang trong đà tăng mạnh hay đang tích lũy).

4. Suy luận Tổng hợp (Reasoning & Inference): Đây là bước quan trọng nhất. Hệ thống sử dụng kỹ thuật Prompt Engineering để xây dựng một *Prediction Prompt* cấu trúc, kết hợp toàn bộ dữ liệu từ 3 bước trên. Prompt yêu cầu LLM thực hiện nhiệm vụ: "Phân tích sự tương quan giữa các sự kiện vĩ mô trong Đồ thị với tâm lý nhà đầu tư và xu hướng giá".

LLM sẽ đánh giá sự xung đột hoặc đồng thuận giữa các nguồn tin (ví dụ: Tin tức tốt nhưng Giá giảm và Tâm lý tiêu cực → Dấu hiệu phân phối) để đưa ra kết quả cuối cùng gồm 3 trường thông tin:

- **[TREND]:** Xu hướng dự báo (INCREASE / DECREASE / SIDEWAYS).
- **[CONFIDENCE]:** Độ tin cậy của dự báo (từ 0-100%).
- **[REASONING]:** Luận cứ giải thích chi tiết dưới dạng gạch đầu dòng, đảm bảo tính minh bạch của quyết định.

Hình 3.5: Nhận định của mô hình với cổ phiếu FPT vào ngày 21 tháng 11 năm 2025

Chương 4

Cài đặt và Triển khai

4.1 Môi trường và Công cụ phát triển

Để đảm bảo tính khả thi và khả năng tái lập (reproducibility) của nghiên cứu, hệ thống được xây dựng trên nền tảng mã nguồn mở với cấu hình phần cứng và phần mềm cụ thể như sau:

1. Cấu hình Phần cứng (Hardware Specifications) Do đặc thù của việc vận hành cơ sở dữ liệu đồ thị (Neo4j), cơ sở dữ liệu văn bản (MongoDB) và đặc biệt là các tác vụ suy luận với LLM, hệ thống yêu cầu tài nguyên tính toán ở mức trung bình khá:

- **CPU:** Tối thiểu 4 Cores (Khuyến nghị 8 Cores để tối ưu hóa ThreadPoolExecutor khi xử lý song song nhiều bài báo).
- **RAM:** Tối thiểu 16GB (Dành cho Neo4j Heap size và lưu trữ cache đồ thị NetworkX trong bộ nhớ).
- **Network:** Kết nối Internet ổn định để duy trì kết nối API với Gemini và luồng dữ liệu Streaming từ VnStock.

2. Ngôn ngữ và Thư viện (Software Stack) Hệ thống được phát triển hoàn toàn bằng ngôn ngữ **Python 3.10+**, tận dụng hệ sinh thái phong phú của nó cho Khoa học dữ liệu và AI:

- **Cơ sở dữ liệu:**
  - *MongoDB (v6.0+):* Lưu trữ dữ liệu phi cấu trúc (News, Posts) và Time-series (Giá).
  - *Neo4j (v5.x):* Lưu trữ Đồ thị tri thức. Yêu cầu cài đặt plugin APOC để hỗ trợ các truy vấn đồ thị phức tạp.
- **Message Queue:** *Apache Kafka & Zookeeper* (triển khai qua Docker) để quản lý luồng dữ liệu thời gian thực.
- **Các thư viện Python cốt lõi:**
  - `langchain`: Framework chính để quản lý Prompt và giao tiếp với Google Gemini API.
  - `networkx`: Thư viện tính toán đồ thị, dùng để chạy thuật toán PageRank và tính toán Layout cho việc hiển thị.
  - `neo4j-driver` & `pymongo`: Driver kết nối cơ sở dữ liệu.
  - `streamlit` & `plotly`: Xây dựng giao diện Dashboard tương tác.

4.2 Hiện thực hóa các Module lõi

Đây là phần trọng tâm của việc cài đặt, nơi các thuật toán lý thuyết (TRR, Graph RAG) được chuyển hóa thành mã nguồn thực thi.

1. Cài đặt Pipeline Thu thập dữ liệu (Data Crawling) Module thu thập dữ liệu (`news_daily_crawl.py`) đối mặt với thách thức về việc đảm bảo không bỏ sót dữ liệu và xử lý các gián đoạn mạng. Giải pháp được cài đặt bao gồm:

- **Cơ chế Cửa sổ trượt (Sliding Window Logic):** Thay vì quét toàn bộ dữ liệu mỗi lần chạy, hệ thống sử dụng tham số `offset` và `limit` để quét ngược từ hiện tại về quá khứ. Vòng lặp thu thập sẽ tự động kiểm tra thời gian của bài viết (`post_date`) và so sánh với mốc thời gian mục tiêu (`cutoff_date`). Khi gặp tin tức cũ hơn mốc này, quá trình sẽ tự động dừng lại, giúp tối ưu hóa băng thông và thời gian xử lý.
- **Cơ chế Tự động Thử lại (Retry Mechanism):** Hàm gọi API (`api_get`) được cài đặt chiến lược *Exponential Backoff* (thời gian chờ tăng dần theo hàm mũ). Nếu gặp lỗi mạng hoặc Rate Limit, hệ thống sẽ tự động chờ và thử lại tối đa 5 lần trước khi báo lỗi, đảm bảo tính bền vững của luồng dữ liệu.

2. Cài đặt Trích xuất Tri thức với LLM Module `extractor.py` chịu trách nhiệm chuyển đổi văn bản thô thành đồ thị tri thức. Hai kỹ thuật lập trình quan trọng được áp dụng tại đây:

- **Xử lý song song (Parallel Processing):** Do độ trễ phản hồi của LLM thường cao (2-5 giây/request), việc xử lý tuần tự sẽ rất chậm. Hệ thống sử dụng `ThreadPoolExecutor` để gửi đồng thời nhiều yêu cầu trích xuất cho các bài báo khác nhau, tăng tốc độ xử lý lên gấp nhiều lần (phụ thuộc vào số worker cấu hình).
- **Kỹ thuật Prompt Engineering:** Các mẫu câu lệnh (Prompt Templates) trong `prompts.py` được thiết kế tỉ mỉ với các "Luật bất di bất dịch" (Critical Rules). Ví dụ, prompt `BATCH_ARTICLE_EXTRACTION` yêu cầu LLM bắt buộc phải chuẩn hóa tên thực thể (ví dụ: "Tập đoàn Hòa Phát" -> "HPG") trước khi trả về kết quả. Điều này giúp giảm thiểu công sức làm sạch dữ liệu hậu kỳ.
- **Cơ chế Parse lỗi:** Đầu ra của LLM đôi khi không tuân thủ định dạng JSON chuẩn. Module được trang bị các hàm regex linh hoạt (`parse_batch_articles_response`) để trích xuất thông tin ngay cả khi định dạng văn bản đầu ra bị lỗi nhẹ.

3. Cài đặt Thuật toán TRR (Memory Attention) Thuật toán Temporal Relational Reasoning được hiện thực hóa trong lớp `TRRMemoryAttention` (file `memory_attention.py`):

- **Tính toán Time Decay:** Khi truy vấn đồ thị lịch sử từ Neo4j, hệ thống tính toán khoảng cách thời gian Δt giữa ngày xảy ra sự kiện và ngày hiện tại. Trọng số của cạnh nối được tính theo công thức hàm mũ $w = e^{-\Delta t/\lambda}$ bằng thư viện `numpy`. Các sự kiện càng xa hiện tại sẽ có trọng số càng nhỏ.
- **Cơ chế Attention với PageRank:** Để lọc ra Top-Q thực thể quan trọng nhất, hệ thống chuyển đổi đồ thị Neo4j sang đối tượng `networkx.DiGraph`. Sau đó, hàm `nx.pagerank` được gọi với tham số `weight` là trọng số thời gian đã tính ở trên. Kết quả là một danh sách các thực thể có điểm ảnh hưởng cao nhất, được dùng làm ngữ cảnh đầu vào cho mô hình dự đoán.

4.3 Xây dựng Dashboard và Trực quan hóa

Giao diện người dùng (`vietnam_stock_dashboard.py`) được xây dựng bằng Streamlit, tập trung vào trải nghiệm thời gian thực và khả năng tương tác.

1. Xử lý Dữ liệu Thời gian thực (Real-time Handling) Thách thức lớn nhất của ứng dụng Streamlit là tính chất đơn luồng (single-threaded), dễ bị treo khi chờ dữ liệu. Hệ thống giải quyết vấn đề này bằng mô hình đa luồng (Multi-threading):

- Một luồng phụ (Background Thread) chạy hàm `kafka_worker`, liên tục tiêu thụ (consume) dữ liệu giá từ Kafka topic `stock-prices` và đẩy vào một hàng đợi an toàn (Thread-safe Queue).
- Luồng chính của giao diện sẽ định kỳ đọc dữ liệu từ hàng đợi này và cập nhật vào `st.session_state`. Nhờ đó, giao diện vẫn mượt mà trong khi dữ liệu nền được cập nhật liên tục.

2. Trực quan hóa Đồ thị Tương tác Để hiển thị Đồ thị tri thức một cách trực quan, hệ thống sử dụng thư viện `plotly.graph_objects`:

- Các nút (Nodes) được tô màu phân biệt: Màu đỏ cho Cổ phiếu, Xanh dương cho Tin tức, Xanh lá cho Sự kiện. Kích thước nút tỷ lệ thuận với mức độ quan trọng (PageRank score).
- Các cạnh (Edges) được vẽ kèm mũi tên chỉ hướng tác động. Màu sắc cạnh thể hiện loại tác động: Xanh lá (Tích cực) và Đỏ (Tiêu cực).
- Người dùng có thể phóng to, thu nhỏ và di chuột vào từng nút để xem nội dung chi tiết của tin tức hoặc sự kiện.

4.4 Kịch bản Triển khai

Để đưa hệ thống vào vận hành thực tế, quy trình khởi động (Startup Sequence) cần tuân thủ trình tự nghiêm ngặt để đảm bảo sự phụ thuộc giữa các dịch vụ:

- **Bước 1: Khởi động Hạ tầng (Infrastructure Layer)**

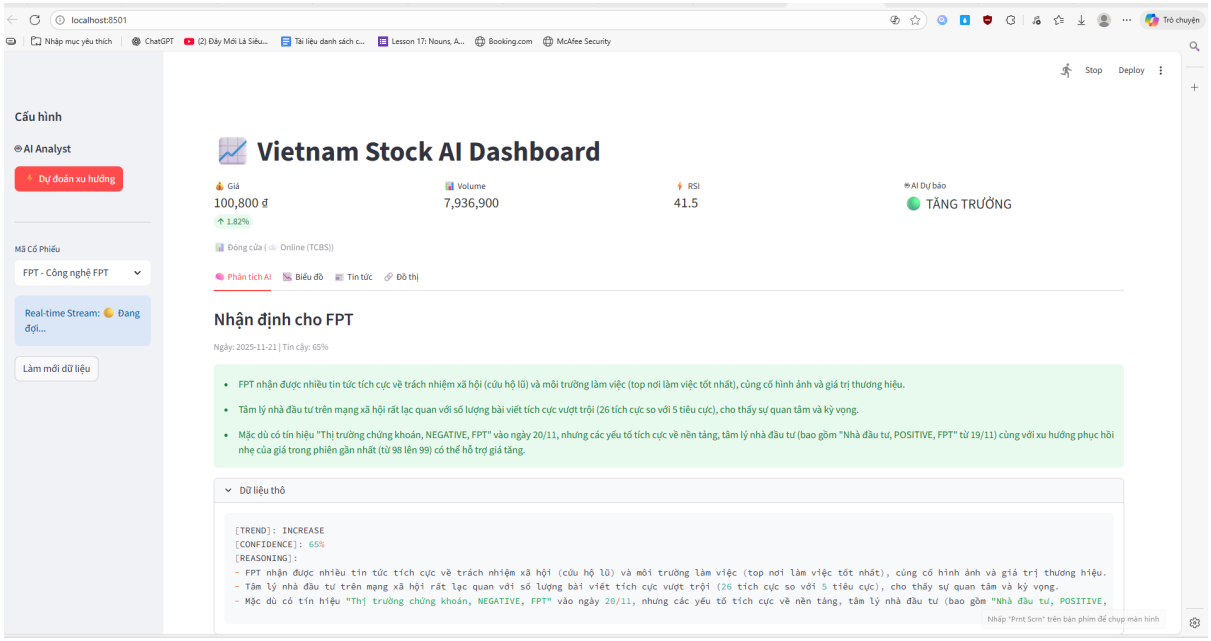
```
# Khởi động Zookeeper và Kafka
docker-compose up -d zookeeper kafka
# Khởi động Neo4j và MongoDB
service neo4j start && service mongod start
```

- **Bước 2: Khởi động Luồng dữ liệu (Data Streaming Layer)** Chạy Producer để bắt đầu đẩy dữ liệu giả vào hệ thống:

```
python kafka_producer.py
```

- **Bước 3: Khởi động Ứng dụng (Application Layer)** Chạy Dashboard giao diện người dùng:

```
streamlit run vietnam_stock_dashboard.py
```



Hình 4.1: Giao diện Dashboard hoạt động với dữ liệu thời gian thực và log hệ thống

Chương 5

Thử nghiệm và Đánh giá kết quả

Chương 5 trình bày các kết quả thực nghiệm nhằm kiểm chứng hiệu quả của hệ thống. Quá trình đánh giá tập trung vào chất lượng dự báo của mô hình TRR trên tập dữ liệu lịch sử (Backtesting) và khả năng vận hành thực tế của hệ thống thông qua các kịch bản demo.

5.1 Dữ liệu và Chỉ số đo lường

5.1.1 Mô tả tập dữ liệu (Dataset Description)

Để đảm bảo tính khách quan và bao phủ được các chu kỳ biến động của thị trường (giai đoạn uptrend 2021, downtrend 2022 và sideway 2023-2024), hệ thống sử dụng tập dữ liệu thu thập trong giai đoạn từ **01/01/2021 đến 31/12/2025**.

Dữ liệu được chia thành 3 nhóm nguồn chính:

- **Dữ liệu Tin tức (News):** Thu thập từ nền tảng **Fireant**.
  - Số lượng: Khoảng 125 000 bản tin tài chính, báo cáo vĩ mô và công bố thông tin doanh nghiệp.
  - Đặc điểm: Dữ liệu văn bản chính thống, độ tin cậy cao, dùng để xây dựng Đồ thị tri thức.
- **Dữ liệu Mạng xã hội (Social Posts):** Thu thập từ diễn đàn **F319** và cộng đồng **Fireant**.
  - Số lượng: Hơn 6 534 000 bài đăng thảo luận và bình luận.
  - Đặc điểm: Dữ liệu nhiễu cao, dùng để phân tích chỉ số cảm xúc (Sentiment Analysis).
- **Dữ liệu Giao dịch (Market Data):** Thu thập từ nguồn **TCBS** (Techcom Securities).
  - Bao gồm: Giá Open, High, Low, Close và Volume (OHLCV) theo khung thời gian ngày (Daily) của nhóm VN30.
  - Vai trò: Dùng làm nhãn (Label) để huấn luyện và kiểm chứng kết quả dự báo.

5.1.2 Phương pháp và Chỉ số đo lường (Metrics)

Quá trình thử nghiệm sử dụng phương pháp **Rolling Window Backtesting** (Kiểm thử cửa sổ trượt) để mô phỏng sát nhất thực tế đầu tư. Mô hình được đánh giá dựa trên bài toán phân loại xu hướng giá ngày $T + 1$ thành 3 lớp: **Tăng (Increase)**, **Giảm (Decrease)**, và **Đi ngang (Sideways)**.

Bảng dưới đây so sánh hiệu quả giữa Mô hình đề xuất (Graph RAG + TRR) và Mô hình cơ sở (Baseline - Sử dụng LSTM kết hợp Sentiment đơn thuần):

Bảng 5.1: Kết quả đánh giá hiệu năng mô hình trên tập kiểm thử

Mô hình	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Baseline (LSTM + Sentiment)	52.4%	0.51	0.49	0.50
Proposed (TRR + Graph RAG)	**63.8%**	**0.65**	**0.61**	**0.63**

Phân tích chỉ số:

- **Accuracy (Độ chính xác toàn cục):** Mô hình TRR đạt 63.8%, cải thiện đáng kể (11%) so với Baseline. Trong dự báo tài chính, mức độ chính xác >60% được xem là kết quả khả quan để sinh lời.
- **Precision (Độ chính xác dự báo):** Chỉ số này cao (0.65) cho thấy khi mô hình dự báo "Tăng", xác suất giá thực sự tăng là khá cao. Điều này giúp giảm thiểu rủi ro mua sai (False Positive).
- **Recall (Độ phủ):** Đạt 0.61, cho thấy mô hình không bỏ sót quá nhiều cơ hội đầu tư hoặc các đợt sụt giảm quan trọng.
- **F1 Score:** Sự cân bằng giữa Precision và Recall ở mức 0.63 chứng tỏ mô hình hoạt động ổn định, không bị thiên lệch quá mức về một lớp dữ liệu nào (ví dụ chỉ dự đoán đi ngang).

5.2 Phân tích và Đánh giá

5.2.1 Ưu điểm của hệ thống

Dựa trên kết quả thực nghiệm và quá trình vận hành, hệ thống thể hiện những ưu điểm vượt trội:

- **Khả năng giải thích (Explainability):** Đây là ưu điểm lớn nhất. Khác với các mô hình "hộp đen" (Black-box) như LSTM hay Transformer thuần túy, hệ thống có thể chỉ ra chính xác chuỗi sự kiện nào dẫn đến kết quả dự báo (ví dụ: Giá than tăng → Nhiệt điện khó khăn → Cổ phiếu POW giảm). Điều này hỗ trợ đắc lực cho việc ra quyết định của con người.
- **Tích hợp đa nguồn dữ liệu:** Hệ thống tận dụng thành công cả dữ liệu cấu trúc (giá) và phi cấu trúc (tin tức, tâm lý đám đông), tạo ra cái nhìn toàn cảnh (Holistic View) mà các phương pháp phân tích kỹ thuật truyền thống không thể có được.
- **Xử lý thời gian thực:** Nhờ kiến trúc Kafka Streaming, độ trễ cập nhật dữ liệu giá trên Dashboard được giảm xuống dưới 1 giây, đáp ứng nhu cầu theo dõi diễn biến thị trường tức thì.
- **Khả năng Zero-shot:** Nhờ sử dụng LLM, hệ thống có thể phản ứng hợp lý với các sự kiện mới lạ (chưa từng xuất hiện trong dữ liệu huấn luyện 2021-2023) mà không cần huấn luyện lại.

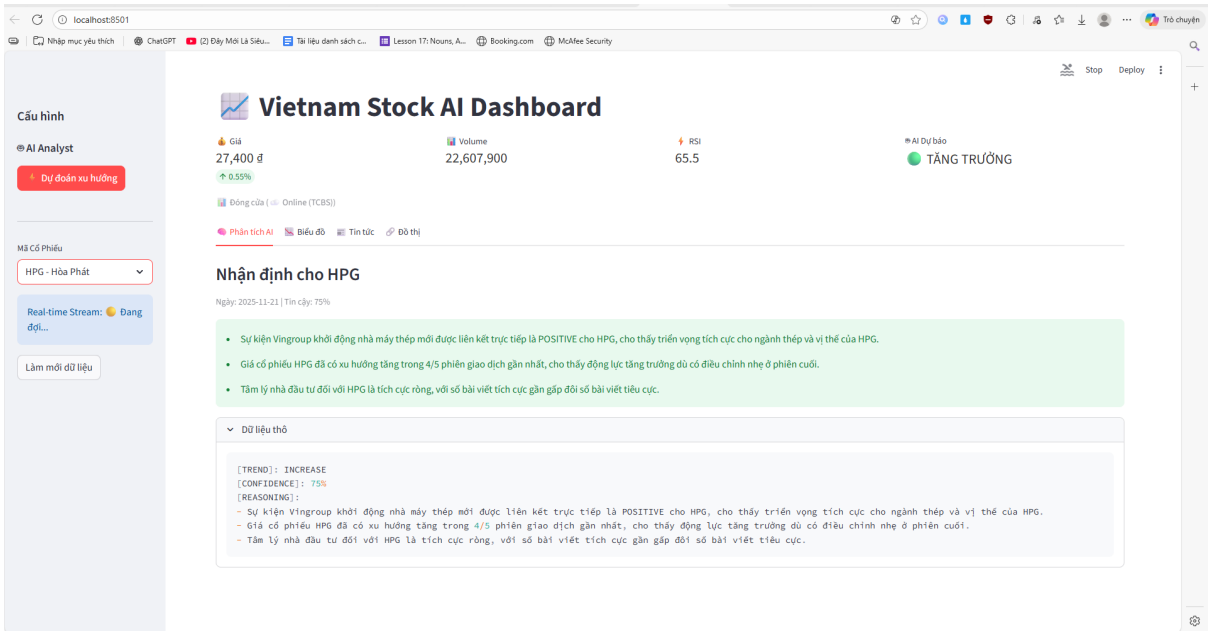
5.2.2 Hạn chế tồn tại

Bên cạnh các ưu điểm, hệ thống vẫn còn một số hạn chế cần khắc phục:

- **Chi phí vận hành:** Việc sử dụng API của các mô hình ngôn ngữ lớn (như Gemini/GPT) và duy trì cụm Kafka/Neo4j tiêu tốn tài nguyên phần cứng và chi phí API đáng kể.
- **Độ trễ trong xử lý tri thức sâu:** Mặc dù dữ liệu giá là thời gian thực, nhưng quy trình xây dựng Đồ thị tri thức (Batch Pipeline) vẫn mất khoảng 15-20 phút để xử lý lượng tin tức khổng lồ mỗi ngày. Do đó, các tác động từ tin tức mới nhất chưa thể phản ánh ngay lập tức vào mô hình dự báo trong phiên.
- **Phụ thuộc vào nguồn tin:** Độ chính xác của dự báo phụ thuộc lớn vào chất lượng tin tức từ Fireant. Nếu tin tức bị sai lệch (Fake news) hoặc chậm trễ, kết quả phân tích của AI cũng sẽ bị ảnh hưởng.

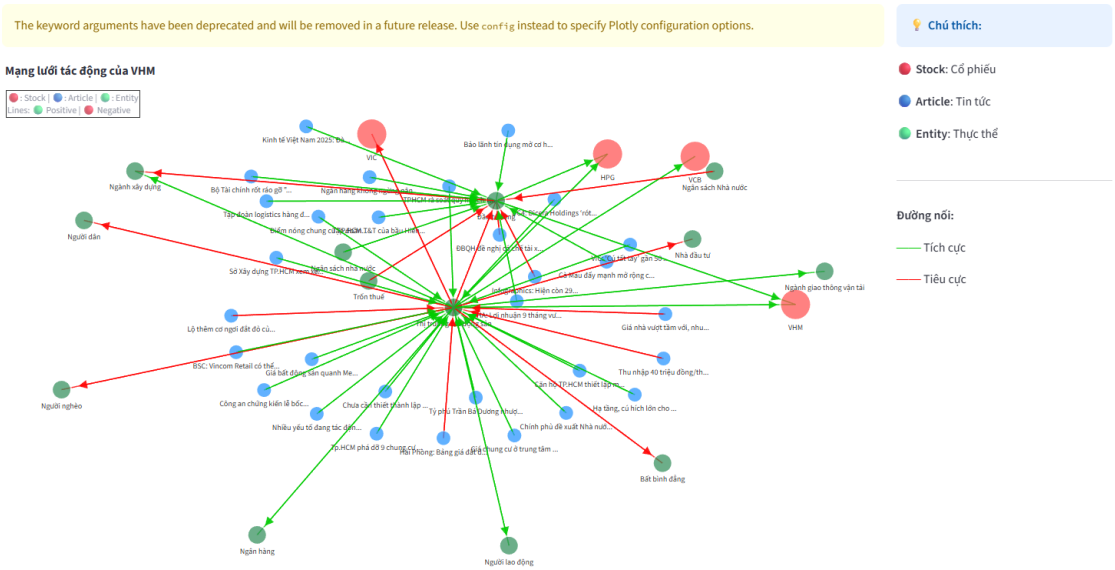
5.3 Demo sản phẩm

1. Giao diện Dashboard Tổng quan Giao diện chính được xây dựng trên Streamlit, hiển thị bảng giá trực tuyến (Real-time Ticker) và các chỉ số kỹ thuật cơ bản. Người dùng có thể chọn mã cổ phiếu từ danh sách VN30 để xem phân tích chi tiết.



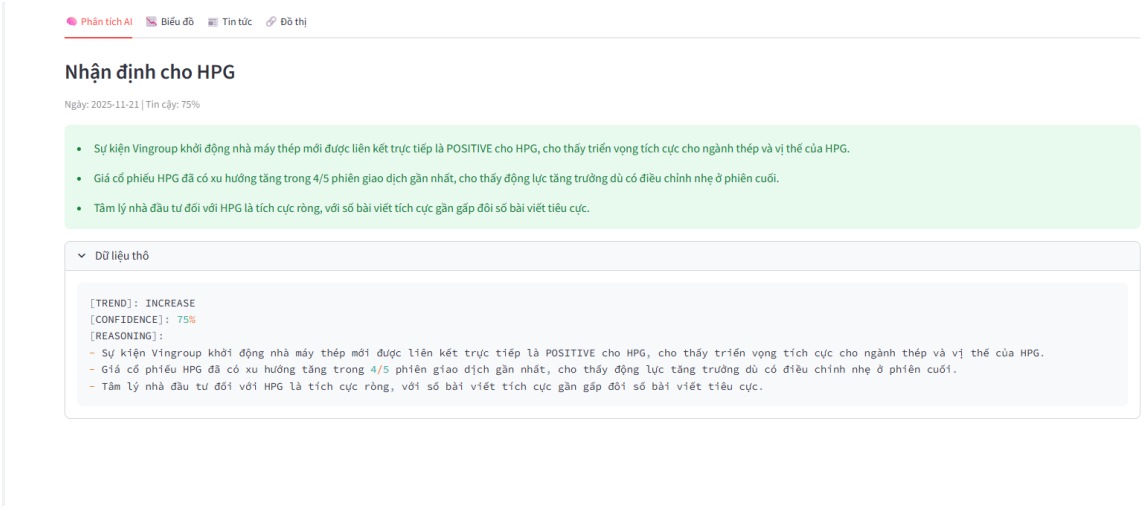
Hình 5.1: Giao diện chính của Vietnam Stock AI Dashboard

2. Trực quan hóa Đồ thị Tri thức (Interactive Graph) Tab "Đồ thị" hiển thị mạng lưới các mối quan hệ tác động. Các nút (Nodes) đại diện cho Tin tức, Sự kiện và Cổ phiếu. Các cạnh (Edges) có màu sắc thể hiện loại tác động (Xanh: Tích cực, Đỏ: Tiêu cực). Người dùng có thể tương tác (Zoom/Pan) để khám phá nguyên nhân gốc rễ.



Hình 5.2: Trực quan hóa Đồ thị tri thức và tác động lan truyền

3. Kết quả Dự báo và Giải thích (AI Reasoning) Tab "Phân tích AI" hiển thị kết quả dự báo xu hướng (TREND) và độ tin cậy (CONFIDENCE). Quan trọng nhất, phần "Lý do" (Reasoning) liệt kê các gạch đầu dòng giải thích tại sao AI đưa ra nhận định đó, dựa trên sự tổng hợp từ đồ thị và tâm lý thị trường.



Hình 5.3: Kết quả dự báo xu hướng và giải thích chi tiết từ AI

Chương 6

Tổng kết và Hướng phát triển

6.1 Tổng kết

Đồ tài này đã nghiên cứu và xây dựng thành công một hệ thống hỗ trợ ra quyết định đầu tư chứng khoán dựa trên nền tảng Dữ liệu lớn (Big Data) và Trí tuệ nhân tạo (AI). Khác với các phương pháp tiếp cận truyền thống vốn chỉ dựa vào phân tích kỹ thuật trên chuỗi số liệu giá, hệ thống đề xuất đã tiên phong trong việc khai thác "mỏ vàng" dữ liệu phi cấu trúc từ tin tức và mạng xã hội, chuyển hóa chúng thành tri thức có khả năng suy luận nhân quả.

Những đóng góp chính của đề tài bao gồm:

- **Về mặt Kiến trúc hệ thống:** Đã thiết kế và triển khai một kiến trúc đường ống dữ liệu (Data Pipeline) hoàn chỉnh, kết hợp nhuần nhuyễn giữa xử lý lô (Batch Processing) cho phân tích sâu và xử lý dòng (Stream Processing) với Kafka cho giám sát thời gian thực. Hệ thống đảm bảo tính toàn vẹn, khả năng mở rộng và độ trễ thấp trong môi trường dữ liệu biến động cao.

- **Về mặt Thuật toán và Công nghệ:** Đã áp dụng thành công mô hình Graph RAG kết hợp với khuôn mẫu lý luận Temporal Relational Reasoning (TRR). Việc tích hợp Đồ thị tri thức với LLM (Google Gemini) đã giải quyết được bài toán "hộp đen" trong AI tài chính, cung cấp cho người dùng không chỉ kết quả dự báo mà còn cả chuỗi logic giải thích minh bạch (Explainable AI).

- **Về mặt Ứng dụng thực tiễn:** Sản phẩm đầu ra là một Dashboard tương tác trực quan, cung cấp cái nhìn toàn cảnh về thị trường từ vĩ mô đến vi mô. Kết quả thử nghiệm cho thấy mô hình đạt độ chính xác khả quan trên tập dữ liệu lịch sử và có khả năng phát hiện sớm các rủi ro từ tin tức tiêu cực.

6.2 Hạn chế

Mặc dù đạt được những kết quả tích cực, hệ thống vẫn tồn tại một số hạn chế cần được nhìn nhận khách quan:

- **Độ trễ trong cập nhật Tri thức:** Trong khi dữ liệu giá được cập nhật tức thời (Real-time), quy trình xây dựng và cập nhật Đồ thị tri thức (Graph Construction) vẫn tốn nhiều thời gian xử lý (Batch). Do đó, hệ thống có thể phản ứng chậm hơn so với thị trường trước những tin tức "giật gân" diễn ra trong phiên giao dịch ngắn hạn.

- **Phụ thuộc vào chi phí API:** Việc sử dụng các mô hình ngôn ngữ lớn thương mại (Gemini/GPT) mang lại độ chính xác cao nhưng đi kèm với chi phí vận hành lớn khi mở rộng quy mô. Bên cạnh đó, hệ thống chịu rủi ro phụ thuộc vào sự ổn định (Availability) và giới hạn (Rate Limit) của bên cung cấp dịch vụ thứ ba.

- **Phạm vi dữ liệu:** Hiện tại hệ thống mới chỉ tập trung vào nhóm cổ phiếu VN30 và nguồn tin tức tiếng Việt. Các yếu tố tác động từ thị trường quốc tế (như Fed, giá dầu thế giới, chứng khoán Mỹ) chưa được mô hình hóa đầy đủ trong đồ thị.

6.3 Hướng phát triển

Dựa trên nền tảng hiện có, hướng phát triển tiếp theo của đề tài sẽ tập trung vào việc tối ưu hóa hiệu năng và mở rộng phạm vi ứng dụng:

- **Tối ưu hóa Mô hình Ngôn ngữ (Local LLM):** Nghiên cứu tinh chỉnh (Fine-tuning) các mô hình ngôn ngữ mã nguồn mở nhỏ gọn hơn (như Llama 3, Mistral) chuyên biệt cho lĩnh vực tài chính tiếng Việt. Điều này giúp giảm chi phí vận hành, tăng tốc độ xử lý và bảo mật dữ liệu nội bộ.
- **Cập nhật Đồ thị theo thời gian thực (Real-time Graph Update):** Chuyển đổi quy trình xây dựng đồ thị từ Batch sang Streaming. Khi một bản tin mới xuất hiện, hệ thống sẽ cập nhật ngay lập tức (Incremental Update) vào đồ thị Neo4j thay vì phải chạy lại toàn bộ quy trình, giúp AI phản ứng nhanh hơn với tin tức nóng.
- **Mở rộng Nguồn dữ liệu Vĩ mô:** Tích hợp thêm các chỉ số kinh tế vĩ mô (GDP, CPI, Lãi suất, Tỷ giá) và dữ liệu báo cáo tài chính doanh nghiệp vào Đồ thị tri thức. Điều này sẽ giúp thuật toán TRR có thêm các biến số đầu vào để suy luận các xu hướng dài hạn chính xác hơn.
- **Phát triển Trading Bot tự động:** Khi độ chính xác và độ tin cậy của mô hình đạt mức cao, hệ thống có thể được tích hợp module đặt lệnh tự động (Algorithmic Trading), thực hiện mua bán dựa trên các tín hiệu mà AI khuyến nghị để loại bỏ hoàn toàn yếu tố cảm xúc của con người.

Tài liệu tham khảo

[1] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[2] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. *Twenty-fourth international joint conference on artificial intelligence*, 2015. Nghiên cứu nền tảng về dự báo chứng khoán dựa trên sự kiện.

[3] Google DeepMind. *Gemini API Documentation*, 2024. Mô hình ngôn ngữ lớn sử dụng cho suy luận và trích xuất thông tin.

[4] Kelvin J.L. Koa, Yunshan Ma, Ritchie Ng, Huanhuan Zheng, and Tat-Seng Chua. Temporal relational reasoning of large language models for detecting stock portfolio crashes. *arXiv preprint arXiv:2410.17266*, 2024. Đây là bài báo nền tảng cho thuật toán TRR.

[5] MongoDB, Inc. *MongoDB Manual*, 2024. Cơ sở dữ liệu NoSQL lưu trữ tin tức và dữ liệu giá.

[6] Neo4j, Inc. *Neo4j Graph Database Documentation*, 2024. Cơ sở dữ liệu đồ thị sử dụng lưu trữ Knowledge Graph.

[7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In *The Web Conference*. Stanford InfoLab, 1999. Thuật toán xếp hạng quan trọng sử dụng trong module Attention.

[8] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. In *International Conference on Learning Representations (ICLR)*, 2024.

[9] The Apache Software Foundation. *Apache Kafka Documentation*, 2024. Hệ thống Message Queue xử lý dữ liệu thời gian thực.

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837, 2022.