

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CMC**

**CÔNG TRÌNH THAM DỰ**

**HỘI NGHỊ SINH VIÊN NGHIÊN CỨU KHOA HỌC CẤP KHOA**

NĂM HỌC 2024 – 2025

Tên công trình: **Ứng dụng học sâu trong bài toán dự đoán giá cổ**

**phiếu cho nhiều công ty dựa trên dữ liệu chuỗi thời gian**

Nhóm sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| 1. Lê Hải Đức, | mã SV: BCS230118, | lớp: 23IT-GM |
| 2. Trần Quỳnh Anh, | mã SV: BIT230008, | lớp: 23IT-GM |

Đơn vị: Khoa CNTT - TT

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | TS. Phạm Thị Kim Dung  TS. Ngô Hoàng Huy |
|  |  |

***Hà Nội, tháng 04 năm 2025***

# TÓM TẮT

Dự đoán giá cổ phiếu là một công nghệ tiên tiến, đang đóng vai trò ngày càng quan trọng tại Việt Nam và thế giới. Nghiên cứu này tập trung vào việc áp dụng mô hình CorrSTN vốn chỉ được sử dụng cho bài toán dự báo lưu lượng giao thông vào bài toán dự báo giá cổ phiếu cho nhiều công ty do có sự tương đồng về dữ liệu, đặc biệt chú trọng đến độ tương quan về không thời gian và khả năng trích xuất các đặc trưng quan trọng sử dụng cơ chế Multi-Head Attention. Bộ dữ liệu thực nghiệm là dữ liệu giá cổ phiếu của 10 công ty thuộc về 3 lĩnh vực là công nghệ, kinh tế và tiêu dùng được lấy từ 1/1/2012 tới 12/12/2024 và bao gồm 3.257 bản ghi theo ngày cho mỗi công ty, sau đó kết quả được đánh giá dựa trên các chỉ số hiệu suất chính như độ chính xác về giá đóng cửa của các công ty cho những ngày trong tương lai. Kết quả cho thấy rằng mô hình mang lại những cải thiện đáng kể trong đó tỉ lệ sai số là 1.42%. Nghiên cứu này đề xuất tiếp tục cải tiến mô hình và khám phá các giải pháp kỹ thuật tiên tiến hơn để tối ưu hóa hiệu suất, đáp ứng tốt hơn sự phức tạp và đa dạng của dữ liệu giá cổ phiếu trong thực tế.

**Từ khóa:** CorrSTN

Chúng tôi xin cam đoan công trình nghiên cứu là do nhóm thực hiện. Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng, minh bạch. Các chương trình thực nghiệm đều là do nhóm thực hiện mà có được, không sao chép từ bất cứ nguồn nào.

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | Sinh viên thực hiện |
| C:\Users\tranq\Documents\Zalo Received Files\chữ ký-Ngô Hoàng Huy.jpg |  |

## Dẫn nhập

Trong bối cảnh thị trường tài chính ngày càng biến động, các tổ chức tài chính đang tích cực nghiên cứu và phát triển các hệ thống dự đoán thị trường chứng khoán– một chủ đề quan trọng trong lĩnh vực kinh tế và được quan tâm rộng rãi trong lĩnh vực khoa học dữ liệu. Việc xây dựng một hệ thống dự báo giá cổ phiếu hiệu quả không chỉ giúp nhà đầu tư đưa ra quyết định sáng suốt hơn, mà còn hỗ trợ các tổ chức tài chính trong việc quản trị rủi ro, tối ưu hóa danh mục đầu tư và phát hiện sớm các xu hướng thị trường tiềm năng.

Dữ liệu giá cổ phiếu thường được mô hình hóa dưới dạng chuỗi thời gian – tức là các quan sát được ghi nhận theo trình tự thời gian cố định, trong đó mối quan hệ giữa các thời điểm kế tiếp đóng vai trò quan trọng trong việc dự đoán xu hướng tiếp theo  [[1]](#Ref1). Tuy nhiên, bản chất phi tuyến tính, biến động mạnh và chịu ảnh hưởng bởi nhiều yếu tố kinh tế – xã hội khiến việc dự báo giá cổ phiếu trở thành một trong những bài toán phức tạp và đầy thách thức [[2]](#Ref2). Trong những năm gần đây, với sự phát triển mạnh mẽ của trí tuệ nhân tạo, đặc biệt là các mô hình học sâu (Deep Learning) [[3]](#Ref3), độ chính xác và khả năng tổng quát trong dự đoán giá cổ phiếu đang dần được tăng lên.

Nghiên cứu này tập trung vào việc khai thác tiềm năng của mô hình học sâu trong việc xử lý và dự báo dữ liệu chuỗi thời gian tài chính, cụ thể là giá cổ phiếu của nhiều công ty khác nhau. Thay vì xây dựng mô hình dự báo riêng lẻ cho từng công ty, chúng tôi xây dựng một kiến trúc học sâu có tên là **CorrSTN (Correlation Information-based Spatiotemporal Network)** – một mô hình học sâu ban đầu được thiết kế cho bài toán **dự đoán lưu lượng giao thông** [[4]](#Ref4) – để ứng dụng vào lĩnh vực tài chính. CorrSTN có khả năng học đồng thời mối quan hệ không gian (giữa các công ty) và quan hệ thời gian (trong chuỗi dữ liệu) để cải thiện hiệu quả dự báo. Việc chuyển đổi CorrSTN từ bài toán giao thông sang bài toán dự báo giá cổ phiếu không chỉ thể hiện tính linh hoạt của mô hình, mà còn mở ra hướng tiếp cận mới trong việc tận dụng mối quan hệ liên công ty để cải thiện hiệu quả dự đoán. Mục tiêu của công trình nghiên cứu này là nhằm kiểm chứng độ hiệu quả của mô hình CorrSTN trên dữ liệu thực tế, từ đó làm rõ tiềm năng ứng dụng của các kiến trúc học sâu trong lĩnh vực tài chính, đặc biệt trong bài toán dự báo giá cổ phiếu đa công ty dựa trên chuỗi thời gian.

Những đóng góp chính của nghiên cứu này bao gồm:

* Mô hình CorrSTN được điều chỉnh từ bài toán dự báo lưu lượng giao thông sang dự báo giá cổ phiếu, thể hiện tiềm năng ứng dụng mô hình trong các lĩnh vực khác nhau.
* Nghiên cứu thực hiện so sánh thực nghiệm toàn diện giữa CorrSTN và các phương pháp hiện có, bao gồm cả mô hình thống kê truyền thống và các phương pháp học sâu được sử dụng trong dự báo thị trường chứng khoán.
* Các tối ưu hóa ở cấp độ kiến trúc và huấn luyện được áp dụng cho mô hình CorrSTN nhằm nâng cao độ chính xác và hiệu quả khi áp dụng vào dữ liệu chuỗi thời gian tài chính.

## Mô hình CorrSTN

Mô hình CorrSTN (Correlation Information-based Spatiotemporal Network) là một mạng nơ-ron học sâu được thiết kế nhằm cải thiện hiệu quả dự đoán lưu lượng giao thông. Khác với các phương pháp truyền thống vốn chỉ sử dụng thông tin về thời gian riêng lẻ, CorrSTN tận dụng triệt để thông tin tương quan giữa các chuỗi dữ liệu không gian – thời gian để xây dựng đặc trưng đầu vào chính xác và hiệu quả hơn.

CorrSTN bao gồm năm thành phần chính: SCorr (thông tin tương quan không gian), TCorr (thông tin tương quan thời gian), CIGNN (mạng nơ-ron đồ thị sử dụng thông tin tương quan), CIATT (attention với thông tin tương quan), và cuối cùng là kiến trúc Transformer với cơ chế Encoder-Decoder để trích xuất các thông tin quan trọng [[4]](#Ref4).

### SCorr – Thông tin tương quan không gian

SCorr là thành phần dùng để đo lường mức độ tương quan giữa các cảm biến giao thông trên nhiều thuộc tính, thông qua hệ số MIC (Maximal Information Coefficient). MIC có khả năng phát hiện cả các mối liên hệ tuyến tính lẫn phi tuyến giữa hai chuỗi dữ liệu. Nếu hai cảm biến có xu hướng lưu lượng thay đổi giống nhau theo thời gian, MIC giữa chúng sẽ cao, và SCorr cũng sẽ có giá trị lớn.

Khác với ma trận lân cận truyền thống chỉ cho biết hai cảm biến “có kết nối hay không” (giá trị 0 hoặc 1), SCorr biểu diễn mức độ tương quan bằng các giá trị liên tục từ 0 đến 1. Điều này cho phép mô hình hiểu rõ hơn về mức độ giống nhau về hành vi giữa các cảm biến, kể cả khi chúng không gần nhau về mặt địa lý.

### TCorr – Thông tin tương quan thời gian

TCorr được sử dụng để xác định mối liên hệ giữa dữ liệu hiện tại và dữ liệu trong các chu kỳ thời gian khác nhau như giờ, ngày hoặc tuần. Thay vì thử tất cả các tổ hợp dữ liệu theo kiểu truyền thống (exhaustive search), TCorr giúp mô hình tự động đánh giá dữ liệu nào (giờ, ngày, tuần) có ảnh hưởng nhiều nhất đến kết quả dự báo.

Nhờ vậy, CorrSTN có thể lựa chọn tập dữ liệu đầu vào hiệu quả hơn, giảm đáng kể chi phí tính toán và thời gian huấn luyện. TCorr không chỉ tăng tốc quá trình huấn luyện mà còn giúp mô hình tránh đưa vào những dữ liệu kém liên quan, từ đó cải thiện độ chính xác.

### CIGNN – Mạng nơ-ron đồ thị với thông tin tương quan

CIGNN là phiên bản cải tiến của mạng nơ-ron đồ thị (GNN), nơi việc lan truyền và tổng hợp thông tin giữa các nút (cảm biến) không chỉ dựa vào cấu trúc kết nối vật lý, mà còn được hướng dẫn bởi thông tin tương quan SCorr. Ngoài ra, CIGNN còn sử dụng ma trận trọng số động để thích ứng với sự thay đổi liên tục của dữ liệu trong quá trình huấn luyện.

Nhờ sự kết hợp giữa thông tin tương quan và cấu trúc đồ thị định sẵn, CIGNN có thể học được các đặc trưng chính xác và sâu sắc hơn, giúp tăng hiệu quả trích xuất đặc trưng không gian.

### CIATT – Attention với thông tin tương quan

CIATT là cơ chế attention được cải tiến từ attention truyền thống của Transformer. Thay vì tính toán trọng số chú ý giữa tất cả các vị trí trong chuỗi (có thể gây nhiễu), CIATT sử dụng SCorr để ưu tiên các chuỗi có mức độ tương đồng cao nhất với chuỗi hiện tại. Điều này giúp tập trung vào các phần tử có thông tin liên quan thực sự, đồng thời giảm ảnh hưởng của các phần tử không liên quan.

CIATT mang lại khả năng lọc nhiễu hiệu quả và làm tăng độ chính xác khi mô hình cố gắng tìm các mẫu tương tự trong chuỗi thời gian để dự báo tương lai.

### Kiến trúc Transformer với cơ chế Encoder-Decoder

CorrSTN được triển khai dựa trên kiến trúc Transformer dạng encoder–decoder. Đây là kiến trúc mạnh mẽ trong xử lý chuỗi, nổi bật nhờ khả năng học được các quan hệ dài hạn và phi tuyến giữa các phần tử trong chuỗi. Encoder tiếp nhận dữ liệu lịch sử (theo chu kỳ giờ, ngày, tuần) và mã hóa thành đặc trưng trừu tượng; decoder sử dụng các đặc trưng này để tạo ra chuỗi dự báo theo từng bước.

Điểm khác biệt quan trọng trong CorrSTN so với các mô hình Transformer truyền thống là việc tích hợp CIGNN và CIATT vào từng lớp encoder và decoder. Nhờ đó, mỗi bước xử lý của mô hình không chỉ học từ chuỗi thời gian mà còn đồng thời cập nhật các mối quan hệ giữa các cảm biến và giữa các chu kỳ. Sự kết hợp này giúp CorrSTN xử lý dữ liệu giao thông có cấu trúc phức tạp một cách hiệu quả và chính xác hơn, tạo ra dự báo ổn định và tin cậy hơn trong các hệ thống giao thông thông minh.

1. **Áp dụng mô hình CorrSTN trong dự đoán giá cổ phiếu cho nhiều công ty**

Mặc dù mô hình CorrSTN ban đầu được phát triển để dự đoán lưu lượng giao thông, nhưng mô hình này có thể dễ dàng áp dụng cho các bài toán dự đoán giá cổ phiếu, nhờ vào những đặc điểm chung giữa hai vấn đề này. Cả dự đoán giao thông và cổ phiếu đều liên quan đến việc xử lý chuỗi thời gian, trong đó các yếu tố không gian và thời gian đóng vai trò quan trọng trong việc dự đoán.

Mô hình CorrSTN đặc biệt hiệu quả trong việc khai thác mối quan hệ không gian và thời gian giữa các yếu tố dữ liệu. Trong trường hợp giao thông, các cảm biến giao thông trên các tuyến đường có thể có sự tương tác lẫn nhau, và trong lĩnh vực tài chính, các cổ phiếu của các công ty có thể có mối quan hệ tương tự, đặc biệt khi chúng thuộc cùng một ngành hoặc cùng chịu ảnh hưởng từ các yếu tố kinh tế vĩ mô. Mô hình sử dụng SCorr (Spatial Correlation Information) để phát hiện các mối quan hệ không gian giữa các cảm biến và TCorr (Temporal Correlation Information) để nhận diện các mẫu biến động theo chu kỳ trong dữ liệu, giúp tối ưu hóa dự báo.

Với khả năng khai thác thông tin từ dữ liệu chu kỳ (như theo giờ, ngày, tuần), CorrSTN có thể dễ dàng áp dụng cho dữ liệu tài chính, nơi các dữ liệu theo chu kỳ (ngày, tuần, tháng) cũng có ảnh hưởng lớn đến sự biến động của giá cổ phiếu. Thêm vào đó, CIGNN (Correlation Information Graph Neural Network) và CIATT (Correlation Information Multi-Head Attention) trong mô hình giúp khai thác các mối quan hệ phức tạp giữa các yếu tố không gian và thời gian, giúp dự đoán chính xác hơn các biến động của giá cổ phiếu.

Vì thế, mô hình CorrSTN không chỉ có thể áp dụng trong các hệ thống giao thông thông minh mà còn có tiềm năng lớn trong việc dự đoán giá cổ phiếu, đặc biệt là trong việc phân tích các mối quan hệ giữa các công ty và nhận diện các chu kỳ thay đổi của thị trường tài chính.

## Thực nghiệm

### Bài toán thực nghiệm

Mục tiêu của thực nghiệm là đánh giá khả năng áp dụng mô hình CorrSTN vào bài toán dự đoán giá cổ phiếu, thông qua việc sử dụng dữ liệu thực tế từ thị trường tài chính thế giới. Thực nghiệm được thiết kế nhằm kiểm tra xem mô hình có thể học được các đặc trưng không gian – thời gian giữa các dữ liệu cổ phiếu và dự đoán chính xác giá trị tương lai của chúng hay không. Việc này không chỉ giúp xác minh tính khả chuyển của CorrSTN từ lĩnh vực giao thông sang tài chính, mà còn cung cấp một góc nhìn mới trong việc ứng dụng các mô hình học sâu tiên tiến vào phân tích và dự đoán thị trường.

Dữ liệu đầu vào bao gồm chuỗi thời gian giá cổ phiếu của 10 công ty được thu thập thông qua Yahoo Finance API. Các công ty được lựa chọn thuộc nhiều ngành nghề khác nhau, nhằm đảm bảo tính đa dạng và phản ánh rõ các mối quan hệ không gian giữa các biến động cổ phiếu. Mỗi công ty bao gồm các đặc trưng như: giá mở cửa (Open), giá đóng cửa (Close), giá cao nhất (High), giá thấp nhất (Low), khối lượng giao dịch (Volume), và được thu thập theo chu kỳ ngày. Dữ liệu được chia thành tập huấn luyện, kiểm tra và kiểm định để đánh giá hiệu suất của mô hình một cách khách quan.

Dữ liệu đầu ra là giá cổ phiếu của nhiều công ty trong một hoặc nhiều bước thời gian tương lai, được dự đoán dựa trên dữ liệu lịch sử và mối tương quan về không thời gian giữa các công ty. Trong quá trình dự đoán, mô hình tận dụng thông tin tương quan về hành vi giá cổ phiếu giữa các thời điểm và không gian khác nhau cũng như xu hướng biến động theo thời gian, tương tự như cách CorrSTN xử lý mối quan hệ giữa các cảm biến trong bài toán giao thông.

Thực nghiệm được tiến hành nhằm kiểm tra độ chính xác, khả năng học đặc trưng và mức độ ổn định của mô hình CorrSTN trong bối cảnh mới – thị trường tài chính – từ đó làm nền tảng cho việc đánh giá tính ứng dụng rộng rãi của mô hình trong các bài toán chuỗi thời gian đa chiều có yếu tố tương quan cao.

### Môi trường thực nghiệm

Toàn bộ quá trình thực nghiệm được thực hiện trên nền tảng Google Colab, sử dụng ngôn ngữ lập trình Python cùng các thư viện như NumPy, Scikit-learn và PyTorch để xây dựng và huấn luyện mô hình. Google Colab được lựa chọn nhờ khả năng hỗ trợ GPU miễn phí, dễ dàng truy cập và tích hợp với Google Drive. Dữ liệu cổ phiếu được thu thập thông qua Yahoo Finance API, lưu trữ trực tiếp trên Drive và xử lý ngay trong môi trường Colab, giúp tối ưu hiệu quả thử nghiệm và thuận tiện cho việc tái sử dụng, mở rộng nghiên cứu.

### Dữ liệu thực nghiệm

Dữ liệu được sử dụng trong thực nghiệm là chuỗi bản ghi giá cổ phiếu của 10 công ty niêm yết trên thị trường chứng khoán, được thu thập thông qua Yahoo Finance API. Các công ty được lựa chọn thuộc nhiều lĩnh vực khác nhau như công nghệ, tài chính và tiêu dùng nhằm đảm bảo tính đa dạng và cho phép mô hình khai thác các mối quan hệ tương quan giữa các ngành nghề trong thị trường. Dữ liệu được lấy theo chu kỳ ngày (daily) trong khoảng thời gian từ 1/1/2012 đến 12/12/2024, bao gồm các đặc trưng phổ biến như: giá mở cửa (Open), giá cao nhất (High), giá thấp nhất (Low), giá đóng cửa (Close), và khối lượng giao dịch (Volume).

### Thu thập và tiền xử lý dữ liệu

### Loại dữ liệu

Dữ liệu thử nghiệm trong nghiên cứu này là chuỗi thời gian giá cổ phiếu của 10 công ty thuộc các ngành công nghệ, tài chính và tiêu dùng. Mỗi mẫu dữ liệu bao gồm các thông tin về giá mở cửa (Open), giá cao nhất (High), giá thấp nhất (Low), giá đóng cửa (Close), và khối lượng giao dịch (Volume).

* + 1. **Nguồn dữ liệu**

Dữ liệu được thu thập từ Yahoo Finance API trong khoảng thời gian từ 01/01/2012 đến 12/12/2024. Đây là nguồn dữ liệu công khai, được đảm bảo tính minh bạch và dễ dàng truy xuất.

* + 1. **Số lượng mẫu**

Tổng cộng có 3.257 bản ghi theo ngày cho mỗi công ty, tạo thành một bộ dữ liệu đa chiều với hơn 30.000 bản ghi (10 công ty x 3.257 ngày).

* + 1. **Đặc điểm của mẫu**
* Đơn vị: 10 công ty đại diện cho các ngành công nghệ, tài chính và tiêu dùng.
* Các đặc trưng bao gồm: giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, và khối lượng giao dịch.
* Chu kỳ lấy mẫu: theo ngày.
  + 1. **Đảm bảo tính đại diện**

Dữ liệu được chọn từ các công ty thuộc nhiều lĩnh vực khác nhau nhằm đảm bảo tính đa dạng, giúp mô hình khai thác tốt mối quan hệ không gian – thời gian giữa các công ty. Việc lựa chọn các công ty thuộc nhiều ngành khác nhau giúp phản ánh đúng sự biến động đa chiều của thị trường tài chính.

* + 1. **Ví dụ minh họa**

Dưới đây là một ví dụ về một số dòng dữ liệu mẫu được sử dụng trong nghiên cứu:

**Microsoft**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Open** | **High** | **Low** | **Close** | **Volume** |
| 2012-01-03 | 26.55 | 26.96 | 26.39 | 26.77 | 64,731,500 |
| 2012-01-04 | 26.82 | 27.47 | 26.78 | 27.40 | 80,516,100 |

**Apple**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Open** | **High** | **Low** | **Close** | **Volume** |
| 2012-01-03 | 14.62 | 14.73 | 14.61 | 14.69 | 302,220,800 |
| 2012-01-04 | 14.64 | 14.81 | 14.62 | 14.77 | 260,022,000 |

### Mã hóa dữ liệu đầu vào

Sau khi dữ liệu được chuẩn hóa và chia theo trình tự thời gian, mỗi mẫu dữ liệu được tổ chức dưới dạng tensor ba chiều với cấu trúc: (số lượng mẫu, số lượng công ty, số lượng đặc trưng). Trước khi đưa vào mô hình, dữ liệu này được biến đổi qua một lớp tuyến tính nhằm đưa các đặc trưng về không gian đặc trưng có số chiều phù hợp với kiến trúc mạng nơ-ron.

Tiếp theo, dữ liệu được mã hóa thông qua các lớp encoder của mô hình CorrSTN. Mỗi lớp encoder kết hợp giữa cơ chế Attention và mạng nơ-ron đồ thị (GCN). Attention giúp mô hình học được các mối liên hệ theo thời gian, trong khi GCN khai thác mối quan hệ không gian giữa các cổ phiếu thông qua ma trận tương quan. Quá trình này cho phép mô hình trích xuất ra các đặc trưng quan trọng, thể hiện được xu hướng biến động của giá cổ phiếu và mối tương quan giữa các công ty trong danh mục đầu tư. Đây là bước quan trọng để chuẩn bị cho quá trình giải mã và dự đoán ở các bước tiếp theo.

### Xây dựng mô hình

### 

Ảnh 1: Mô hình CorrSTN

Mô hình CorrSTN được xây dựng thông qua một hàm khởi tạo mô hình bao gồm đầy đủ các thành phần từ mã hóa đến dự đoán. Quá trình xây dựng mô hình bắt đầu bằng việc tạo ra các ma trận tương quan không gian, bao gồm ma trận lân cận được chuẩn hóa từ dữ liệu cấu trúc đồ thị

, với

trong đó A là ma trận lân cận gốc giữa các cổ phiếu, I là ma trận đơn vị để thêm liên kết tự nối, và là ma trận bậc (số liên kết của mỗi nút).

Bên cạnh đó là ma trận SCorr được đo bằng độ tương quan MIC:

với là chuỗi thời gian của cổ phiếu I theo thuộc tính . SCorr phản ánh mức độ tương đồng giữa hai cổ phiếu trên từng thuộc tính. Hai ma trận này đóng vai trò quan trọng trong việc truyền thông tin giữa các cổ phiếu trong quá trình học không gian.

Tiếp theo, các lớp embedding được khởi tạo để mã hóa dữ liệu đầu vào và đầu ra. Quá trình mã hóa bao gồm hai phần: mã hóa vị trí thời gian (temporal positional encoding) để mô hình nhận biết vị trí tương đối của dữ liệu trong chuỗi thời gian:

giúp mô hình hiểu vị trí của từng bước thời gian theo cách tuần hoàn. Cùng với đó là mã hóa không gian (spatial positional encoding) để thể hiện đặc trưng của từng cổ phiếu trong mạng lưới, thường dùng dưới dạng embedding học được:

. Các mã hóa này được áp dụng lên dữ liệu đầu vào thông qua các lớp tuyến tính:

để chiếu dữ liệu vào không gian ẩn dùng trong mô hình.

CorrSTN sử dụng các cơ chế attention khác nhau tùy thuộc vào cấu hình, bao gồm attention truyền thống và attention có nhận thức về ngữ cảnh thời gian (temporal context-aware). Ba loại attention chính được sử dụng trong mô hình là: attention giữa các bước thời gian trong encoder:

giúp mô hình học quan hệ giữa các bước thời gian dựa trên mức độ liên quan giữa truy vấn Q, khóa K và giá trị V. Tiếp theo là attention giữa dữ liệu đầu ra và dữ liệu đã mã hóa, và attention trong decoder theo hướng nhân quả (causal), tức là mỗi bước thời gian chỉ được phép truy cập các thông tin từ quá khứ đến hiện tại, giúp đảm bảo tính hợp lý và khả năng ứng dụng của mô hình trong thực tế:

trong đó M là ma trận mask chứa để che đi thông tin tương lai.

Trong trường hợp sử dụng **attention có nhận thức tương quan SCorr (CIATT),** mô hình tái tạo key như sau:

và tính attention theo:

nhằm tập trung vào các vị trí có tương quan cao nhất trong SCorr, giúp tăng tính chính xác và giảm nhiễu.s

Sau khi mã hóa, các lớp encoder và decoder được xây dựng từ nhiều tầng (layers), mỗi tầng bao gồm: attention, mạng nơ-ron đồ thị (GCN), kết nối dư (residual connection) và chuẩn hóa lớp (layer normalization). Cụ thể, mạng nơ-ron đồ thị CorrSTN sử dụng là CIGNN, được định nghĩa như sau:

trong đó là ma trận trọng số động:

dùng để thích ứng với sự thay đổi dữ liệu theo thời gian, và là tham số học được cho mỗi thuộc tính. Còn kết nối dư và chuẩn hóa lớp được tính theo:

nhằm ổn định quá trình huấn luyện và giữ lại thông tin đầu vào.Các tầng này cho phép mô hình học được các mối quan hệ phức tạp về không gian và thời gian trong dữ liệu đầu vào.

Cuối cùng, đầu ra của decoder được đưa qua một lớp tuyến tính để chuyển về đúng số chiều đầu ra cần dự đoán (ví dụ: giá cổ phiếu trong tương lai). Tất cả các tham số trong mô hình được khởi tạo bằng phương pháp Xavier (Xavier Initialization) nhằm đảm bảo quá trình học ổn định và hiệu quả ngay từ đầu. Phương pháp này giúp duy trì độ phân bố của tín hiệu qua các tầng của mạng nơ-ron, giảm thiểu hiện tượng mất hoặc bão hòa gradient trong quá trình lan truyền ngược.

* 1. **Cấu hình mô hình và kịch bản kiểm thử**
     1. **Cấu hình mô hình**
        + - Số vòng lặp huấn luyện (epochs): 100
          - Batch size: 16
          - Optimizer: Adam
          - Learning rate: 0.001
          - Dropout: Không sử dụng (dropout = 0)
          - Số tầng (layers): 4 tầng encoder-decoder
          - Số đầu attention (Multi-head attention): 8 (nb\_head = 8)
          - Embedding không gian (SE): bật (SE = 1)
          - Embedding thời gian (TE): bật (TE = 1)
          - Cơ chế attention có trọng số tương quan (ScaledSAt): bật (ScaledSAt = 1)
          - Số ngày dùng làm đầu vào (len\_input): 5 ngày
          - Dự đoán đầu ra trong (num\_for\_predict): 5 ngày
          - Thời gian đầu vào bao gồm: 1 tuần + 3 ngày (num\_of\_weeks = 1, num\_of\_days = 3, num\_of\_hours = 0)
          - attention\_top\_k: 4 (mô hình chỉ tập trung vào 4 cổ phiếu có mức ảnh hưởng mạnh nhất tại mỗi bước)
          - Kích thước vector ẩn (d\_model): 64
          - Kích thước kernel trong GCN: 3
          - Mã hóa nhận thức theo ngữ cảnh thời gian (aware\_temporal\_context): bật
          - Số lớp smoothing: 1
     2. **Chiến lược chia dữ liệu và kiểm thử**
        + - Tỷ lệ chia: 60% tập huấn luyện, 20% kiểm tra, 20% kiểm định
          - Phương pháp chia: tuần tự theo thời gian để tránh rò rỉ thông tin từ tương lai
          - Cách tổ chức dữ liệu: sử dụng cấu trúc sliding window cho các chuỗi đầu vào – đầu ra, tức là một cửa sổ cố định được “trượt” dọc theo chuỗi thời gian để tạo ra nhiều cặp mẫu huấn luyện – dự đoán. Cách tiếp cận này giúp tận dụng tối đa dữ liệu lịch sử và tạo ra nhiều mẫu học có tính kế tiếp thực tế.
          - Không sử dụng kỹ thuật fine-tune hoặc dropout, nhằm đảm bảo thông tin được bảo toàn xuyên suốt chuỗi thời gian
          - Không áp dụng early stopping hoặc cross-validation, mô hình được đánh giá theo các cấu hình khác nhau nhằm kiểm tra độ tổng quát
     3. **Kịch bản kiểm thử**

**4.7.2.1. Kịch bản 1 – So sánh các phiên bản CorrSTN (CorrSTN 1 → CorrSTN 4)**

* + - * + Mục tiêu: đánh giá tác động của từng cải tiến kiến trúc đến hiệu suất.
        + Kết quả: CorrSTN 2 đạt MAPE thấp nhất (1.42%), cho thấy tối ưu hóa lớp spatialAttentionGCN là hiệu quả nhất.

**4.7.2.2. Kịch bản 2 – Thay đổi độ dài chuỗi đầu vào – đầu ra**

* + - * + Mục tiêu: xác định ảnh hưởng của độ dài chuỗi tới chất lượng dự đoán.
        + Cấu hình:

Input = 5 ngày → Output = 3 và 5 ngày.

Input = 10 và 15 ngày → Output = 5 ngày.

* + - * + Nhận xét: chuỗi đầu vào quá dài (15 ngày) làm tăng nhiễu, kết quả kém hơn. Cấu hình tốt nhất là Input = 5 ngày, Output = 3 ngày (MAPE = 1.15%).

**4.7.2.3. Kịch bản 3 – So sánh CorrSTN với các mô hình phổ biến khác (LSTM, ASTGCN, WGAN)**

Chúng tôi đã tự chạy các mô hình LSTM, WGAN, ASTGCN trên một tập dữ liệu độc lập để so sánh hiệu suất của từng mô hình. Các kết quả được đo bằng các chỉ số MAE (Mean Absolute Error), RMSE (Root Mean Square Error), và MAPE (Mean Absolute Percentage Error) như trong bảng dưới đây. Dựa trên các kết quả này, mô hình CorrSTN v2 cho thấy hiệu quả vượt trội hơn so với các mô hình khác về độ chính xác dự đoán, với các giá trị MAE, RMSE và MAPE thấp nhất.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mô hình** | **MAE** | **RMSE** | **MAPE** |
| LSTM | 3.42 | 7.85 | 2.12% |
| WGAN | 3.18 | 7.52 | 2.05% |
| ASTGCN | 2.95 | 7.10 | 1.88% |
| **CorrSTN v2** | **2.61** | **5.99** | **1.42%** |

## Kết quả thực nghiệm

Để đánh giá hiệu quả của mô hình CorrSTN trong dự đoán giá cổ phiếu, chúng tôi đã tiến hành thực nghiệm với tập dữ liệu gồm 10 công ty thuộc các ngành khác nhau trong khoảng thời gian từ năm 2012 đến 2024, lấy từ Yahoo Finance API. Dữ liệu được chia theo chu kỳ ngày, với các cấu hình đầu vào và đầu ra thay đổi nhằm đánh giá khả năng tổng quát và độ chính xác của mô hình trong các điều kiện khác nhau.

Mô hình được huấn luyện trong 100 vòng lặp (epochs), với cấu hình gồm 4 lớp encoder-decoder, cơ chế attention đa đầu (multi-head attention) với 8 heads, và sử dụng embedding không gian – thời gian (SE = 1, TE = 1). Các thông số như attention\_top\_k = 5 giúp mô hình tập trung vào những mối quan hệ có ảnh hưởng lớn nhất giữa các giá cổ phiếu của nhiều công ty. Dự đoán được thực hiện cho chuỗi giá trị 5 ngày tiếp theo dựa trên 10 ngày gần nhất.

### Kết quả so sánh giữa các biến thể CorrSTN

Chúng tôi đã tiến hành cải tiến mô hình CorrSTN qua bốn phiên bản nhằm tối ưu hóa hiệu suất. Kết quả đánh giá bằng ba chỉ số: MAE, RMSE, và MAPE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phiên bản** | **Thay đổi chính** | **MAE** | **RMSE** | **MAPE** |
| CorrSTN 1 | Cải tiến lớp Spatial\_Attention\_Layer | 2.85 | 6.80 | 1.47 |
| CorrSTN 2 | CorrSTN 1 + Cải tiến thêm lớp spatialAttentionGCN | 2.61 | 5.99 | 1.42 |
| CorrSTN 3 | CorrSTN 2 + Cải tiến lớp spatialAttentionScaledGCN | 2.58 | 5.74 | 1.44 |
| CorrSTN 4 | CorrSTN 2 + Cải tiến lớp GCN | 2.86 | 6.89 | 1.48 |

Bảng 1: Các phiên bản CorrSTN

Ở phiên bản đầu tiên **(CorrSTN 1)**, mô hình được tinh chỉnh tại lớp Spatial\_Attention\_Layer bằng cách thay thế phép nhân ma trận *matmul* bằng *einsum* nhằm tối ưu hiệu suất tính toán và bảo toàn cấu trúc tensor. Bên cạnh đó, softmax được ổn định bằng kỹ thuật trừ giá trị lớn nhất ở mỗi hàng để tránh hiện tượng tràn số, và thao tác reshape() được loại bỏ nhằm giữ nguyên cấu trúc chuỗi thời gian và tránh phân mảnh bộ nhớ. Attention trong phiên bản này vẫn tuân theo công thức truyền thống:

Sang phiên bản **CorrSTN 2**, lớp SpatialAttentionGCN được tối ưu hơn với việc bổ sung BatchNorm1d sau attention và GCN để chuẩn hóa đầu ra, cải thiện độ ổn định khi huấn luyện. Hàm kích hoạt ReLU được thay bằng GELU nhằm duy trì tính liên tục của gradient và bảo toàn thông tin âm, đồng thời thêm dropout với xác suất 0.1 để giảm hiện tượng overfitting. Việc sử dụng permute() thay cho reshape() giúp bảo toàn cấu trúc tensor trong xử lý tuần tự. Công thức tính toán GCN được thay đổi từ:

thành

Phiên bản **CorrSTN 3** tiếp tục mở rộng khả năng biểu diễn của mô hình bằng cách thay thế lớp tuyến tính đơn trong GCN bằng một mạng nơ-ron nhiều lớp (MLP – Multilayer Perceptron), được định nghĩa bởi công thức:

Ngoài ra, phiên bản này áp dụng cơ chế attention chuẩn (scaled dot-product), đồng thời tích hợp chuẩn hóa lớp (LayerNorm) và kết nối dư (residual connection) theo công thức:

Việc khởi tạo trọng số trong phiên bản này cũng được cải thiện để đảm bảo quá trình huấn luyện ổn định hơn.

Cuối cùng, ở phiên bản **CorrSTN 4**, mô hình được tối ưu hóa về mặt chi phí tính toán bằng cách loại bỏ vòng lặp *for* trong tính toán GCN và thay thế bằng phép broadcast tensor:

Trong phiên bản này, hàm kích hoạt vẫn là ReLU nhưng không sử dụng dropout nhằm tránh mất mát thông tin trong quá trình truyền tín hiệu qua mạng.

CorrSTN 2 là phiên bản có hiệu quả tốt nhất trong số các biến thể thử nghiệm, đạt kết quả về sai số phần trăm tuyệt đối trung bình thấp nhất, thể hiện khả năng học biểu diễn không gian – thời gian hiệu quả hơn.

### Kết quả theo các cấu hình đầu vào – đầu ra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô hình sử dụng** | **Cấu hình đầu vào – đầu ra** | **MAE** | **RMSE** | **MAPE** |
| CorrSTN 2 | Input: 10 ngày, Output: 5 ngày | 2.61 | **5.99** | 1.42 |
| Input: 5 ngày, Output: 3 ngày | **2.5**2 | 6.83 | **1.1**5 |
| Input: 5 ngày, Output: 5 ngày | 2.82 | 6.72 | 1.45 |
| Input: 15 ngày, Output: 5 ngày | 3.26 | 7.61 | 1.61 |

Bảng 2: Kết quả theo đầu vào - đầu ra

Kết quả cho thấy rằng mô hình hoạt động hiệu quả nhất khi đầu vào là 5 ngày và đầu ra là 3 ngày, đồng thời phiên bản CorrSTN 2 cho độ chính xác tốt nhất về mặt tổng thể. Ngoài ra, việc sử dụng chuỗi đầu vào quá dài (15 ngày) có thể gây ra nhiễu, làm tăng sai số dự đoán.

## Hạn chế và hướng phát triển

## Hạn chế của nghiên cứu

* Giới hạn về loại dữ liệu: Mô hình chỉ sử dụng dữ liệu kỹ thuật, cụ thể là các đặc trưng liên quan đến giá (mở cửa, đóng cửa, cao nhất, thấp nhất) và khối lượng giao dịch. Tuy nhiên, giá cổ phiếu còn bị ảnh hưởng mạnh bởi các yếu tố phi kỹ thuật như tin tức, tâm lý nhà đầu tư, chính sách vĩ mô hoặc các sự kiện bất thường (chiến tranh, dịch bệnh...).
* Đồ thị liên kết công ty tĩnh: Ma trận quan hệ giữa các công ty được xây dựng cố định dựa trên khoảng cách tài chính, không thay đổi theo thời gian. Trong thực tế, mối quan hệ giữa các doanh nghiệp có thể biến động do sáp nhập, thay đổi ngành nghề hoặc yếu tố thị trường.
* Thiếu dữ liệu vĩ mô và cảm xúc: Nghiên cứu chưa tích hợp các yếu tố như lãi suất, tỷ giá, CPI hoặc dữ liệu cảm xúc từ tin tức và mạng xã hội, vốn có thể giúp mô hình phản ứng linh hoạt hơn với thị trường.
  1. **Hướng phát triển**

Để cải thiện độ chính xác và tính ứng dụng của mô hình trong tương lai, một số hướng phát triển có thể được xem xét:

* **Tích hợp dữ liệu phi kỹ thuật:** Kết hợp thêm dữ liệu vĩ mô (GDP, lãi suất...), tin tức tài chính, hoặc dữ liệu cảm xúc từ mạng xã hội nhằm giúp mô hình hiểu được bối cảnh thị trường tốt hơn.
* **Xây dựng đồ thị động theo thời gian:** Thay vì sử dụng một ma trận quan hệ cố định, nghiên cứu trong tương lai có thể xây dựng đồ thị động, cập nhật mối liên hệ giữa các công ty dựa trên tương quan thực tế tại từng thời điểm.
* **Tổng hợp đa mô hình:** Áp dụng học ensemble (tổng hợp mô hình) hoặc kết hợp CorrSTN với các mô hình xử lý ngôn ngữ tự nhiên (NLP) để tận dụng cả dữ liệu định lượng và định tính.
* **Mở rộng phạm vi ứng dụng:** Ngoài dự báo giá cổ phiếu, CorrSTN có thể được mở rộng sang các bài toán tài chính khác như dự báo rủi ro tín dụng, dòng tiền doanh nghiệp, hoặc cảnh báo sớm biến động thị trường.

## Kết luận

Trong nghiên cứu này, mô hình CorrSTN – vốn được phát triển ban đầu cho bài toán dự đoán lưu lượng giao thông – đã được điều chỉnh và áp dụng thành công vào lĩnh vực tài chính, cụ thể là dự đoán giá cổ phiếu của nhiều công ty. Bằng cách tận dụng thông tin tương quan thời gian và không gian giữa các giá cổ phiếu, CorrSTN đã chứng minh được khả năng học đặc trưng sâu và mô hình hóa mối quan hệ phức tạp trong dữ liệu chuỗi thời gian tài chính.

Các thực nghiệm được thực hiện trên dữ liệu của 10 công ty thuộc nhiều ngành nghề trong giai đoạn từ 2012 đến 2024 cho thấy mô hình đạt được kết quả dự đoán chính xác, đặc biệt khi sử dụng kiến trúc cải tiến kết hợp cơ chế chú ý, nhúng không gian – thời gian và mạng nơ-ron đồ thị. Trong số các phiên bản thử nghiệm, CorrSTN 2 – với sự cải tiến lớp SpatialAttentionGCN – cho kết quả tốt nhất về các chỉ số MAE, RMSE và MAPE.

Ngoài ra, việc đánh giá với nhiều cấu hình chuỗi đầu vào và đầu ra cũng cho thấy rằng lựa chọn độ dài chuỗi hợp lý đóng vai trò quan trọng trong hiệu suất dự đoán.

Nhìn chung, kết quả nghiên cứu không chỉ khẳng định tiềm năng ứng dụng của CorrSTN trong lĩnh vực tài chính, mà còn mở ra nhiều hướng phát triển tiếp theo. Những hạn chế và đề xuất mở rộng đã được trình bày cụ thể nhằm hoàn thiện và nâng cao hơn nữa khả năng ứng dụng của mô hình trong thực tiễn.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Ahmed, Sabeen and Nielsen, Ian E and Tripathi, Aakash and Siddiqui, Shamoon and Ramachandran, Ravi P and Rasool, Ghulam, "Transformers in time-series analysis: A tutorial," *Circuits, Systems, and Signal Processing,* vol. 42, pp. 7433--7466, 2023. |
| [2] | Atsalakis, George S and Valavanis, Kimon P, "Surveying stock market forecasting techniques--Part II: Soft computing methods," *Expert Systems with applications,* vol. 36, no. 3, pp. 5932--5941, 2009. |
| [3] | Li, Audeliano Wolian and Bastos, Guilherme Sousa, "Stock market forecasting using deep learning and technical analysis: a systematic review," *IEEE access,* vol. 8, pp. 185232--185242, 2020. |
| [4] | Zhu, Weiguo and Sun, Yongqi and Yi, Xintong and Wang, Yan and Liu, Zhen, "A correlation information-based spatiotemporal network for traffic flow forecasting," *Neural Computing and Applications,* vol. 35, pp. 21181--21199, 2023. |