

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**  
**VIỆN TRÍ TUỆ NHÂN TẠO**

\*\*\*



**BÁO CÁO MÔN HỌC**  
**KHAI PHÁ VÀ PHÂN TÍCH DỮ LIỆU**

**BÀI TẬP LẬP TRÌNH 2**  
**XÂY DỰNG HỆ THỐNG GIAO DỊCH ĐỊNH LƯỢNG**  
**SỬ DỤNG TRANSFORMER**

**Nhóm thực hiện:**

1. Trần Xuân Bảo - 23020332
2. Hà Xuân Huy - 23020375
3. Phan Hoàng Dũng - 23020346

**HÀ NỘI, 04/2025**

# 1. MỞ ĐẦU

Trong kỷ nguyên số hiện nay, thị trường tài chính toàn cầu ngày càng vận hành với tốc độ nhanh chóng và phức tạp, thúc đẩy sự phát triển mạnh mẽ của các hệ thống giao dịch tự động. Trong lĩnh vực này, học sâu – đặc biệt là các mô hình xử lý chuỗi thời gian như Transformer – đang ngày càng chứng minh được tính ưu việt trong việc nhận diện tín hiệu giao dịch tiềm năng từ dữ liệu lịch sử.

Bài báo cáo này tập trung vào việc xây dựng một hệ thống giao dịch định lượng thông minh có khả năng phân loại thời điểm giao dịch theo ba trạng thái: BUY, SELL, HOLD. Nhóm ứng dụng mô hình Transformer – nổi bật với cơ chế attention – kết hợp cùng kỹ thuật mã hóa thời gian Time2Vec và các chỉ báo kỹ thuật kinh điển nhằm tăng cường khả năng biểu diễn đặc trưng của mô hình. Toàn bộ hệ thống được huấn luyện trên tập dữ liệu được thu thập trong 10 năm liên tiếp từ 2011 đến 2020 và kiểm thử trên tập dữ liệu được thu thập trong 4 năm gần đây, và hiệu quả được kiểm chứng qua hai tiêu chí chính: độ chính xác phân loại và hiệu suất sinh lời khi backtest chiến lược giao dịch dựa trên tín hiệu mô hình.

## 2. HIỂU BÀI TOÁN (BUSINESS UNDERSTANDING)

Mục tiêu chính là xây dựng một mô hình phân loại theo chuỗi thời gian, có khả năng dự đoán hành động BUY, SELL hoặc HOLD tại mỗi thời điểm giao dịch, dựa trên dữ liệu thị trường đã quan sát trước đó. Kết quả phân loại sau đó sẽ được sử dụng để thiết kế chiến lược giao dịch và kiểm tra hiệu quả thông qua mô phỏng quá khứ (backtesting).

Hai tiêu chí đánh giá quan trọng của hệ thống bao gồm:

- **Hiệu quả phân loại (Classification Accuracy):** Đo lường mức độ chính xác của mô hình trong việc gán đúng nhãn hành động.
- **Hiệu suất giao dịch (Backtest Performance):** Đánh giá khả năng sinh lời và quản trị rủi ro khi triển khai chiến lược dựa trên tín hiệu từ mô hình.

Ngoài ra, hệ thống còn được khuyến khích tích hợp các kỹ thuật nâng cao như Time2Vec để mã hóa thời gian và các chỉ báo kỹ thuật để mở rộng không gian

đặc trưng đầu vào. Những yếu tố này đóng vai trò then chốt trong việc cải thiện hiệu suất phân loại và tăng độ nhạy với tín hiệu thị trường.

### 3. HIỂU DỮ LIỆU (DATA UNDERSTANDING)

Dữ liệu huấn luyện được sử dụng trong bài tập là dữ liệu lịch sử giá cổ phiếu theo khung thời gian 1 phút (M1), được thu thập từ năm 2011 đến 2020. Dữ liệu kiểm thử được thu thập trong 4 năm gần đây. Tập dev với thời lượng 1 năm, tập test thời lượng 3 năm.

Mỗi bản ghi trong dữ liệu bao gồm các trường sau:

- **Date:** Ngày
- **Time:** Giờ, phút
- **Open:** Giá ở thời điểm đầu khung thời gian (tính bằng USD)
- **High:** Giá cao nhất trong khung thời gian
- **Low:** Giá thấp nhất trong khung thời gian
- **Close:** Giá ở thời điểm cuối khung thời gian
- **Volume:** Khối lượng giao dịch

Dữ liệu có tính chất chuỗi thời gian tần suất cao, chứa nhiều yếu tố nhiễu và biến động ngắn hạn, đồng thời có thể tiềm ẩn các xu hướng, chu kỳ. Đây là đặc điểm rất phù hợp để áp dụng các mô hình học sâu như Transformer, có khả năng học các phụ thuộc dài hạn, phi tuần tự và nhận diện các mẫu hình phức tạp.

Dữ liệu cũng đã được gán nhãn theo công thức:

```
if price_{t+Δ} > price_t * (1 + threshold):  
    label = "BUY"  
elif price_{t+Δ} < price_t * (1 - threshold):  
    label = "SELL"  
else:  
    label = "HOLD"
```

(với  $\Delta = 30$  phút,  $\text{threshold} = 0.0505\% = 0.000505$ )

Nhãn này phản ánh kỳ vọng biến động giá trong ngắn hạn và đóng vai trò là mục tiêu huấn luyện cho mô hình học sâu.

## 4. CHUẨN BỊ DỮ LIỆU (DATA PREPARATION)

### 4.1. Các vấn đề trong tập dữ liệu

Dữ liệu ban đầu bao gồm hai tập:

- *labeled\_train.csv*: dữ liệu dùng để huấn luyện mô hình
- *labeled\_dev.csv*: dữ liệu dùng để kiểm tra mô hình

Mỗi dòng trong tập dữ liệu tương ứng với một phút giao dịch của một mã cổ phiếu, gồm các trường cơ bản như: *Date*, *Time*, *Open*, *High*, *Low*, *Close*, *Volume*, và nhãn giao dịch (Label: *BUY*, *SELL* hoặc *HOLD*).

Tuy nhiên, trong quá trình phân tích sơ bộ, nhóm phát hiện một số vấn đề ảnh hưởng lớn đến việc huấn luyện mô hình, cụ thể như sau:

- **Dữ liệu không liên tục theo phút**: Trong một số khoảng thời gian, dữ liệu chỉ có mỗi giờ một dòng thay vì mỗi phút một dòng, khiến chuỗi thời gian bị đứt gãy nghiêm trọng.
- **Mất mát dữ liệu**: Tổng số dòng thực tế khoảng gần 3.5 triệu, trong khi theo tính toán, phải có ít nhất hơn 5 triệu dòng nếu ghi đầy đủ mỗi phút trong khung giờ giao dịch.
- **Không đồng đều giữa các phiên**: Có phiên dữ liệu đầy đủ, nhưng cũng có phiên chỉ ghi nhận vài chục dòng nên không thể dùng để huấn luyện chuỗi dài.

Những vấn đề trên đặt ra yêu cầu phải có một quy trình xử lý dữ liệu kỹ lưỡng, nhằm lọc ra các đoạn dữ liệu chất lượng cao và tạo ra chuỗi đầu vào hợp lệ cho mô hình Transformer.

### 4.2. Các cách xử lý nhóm đã thực hiện

Để đảm bảo mô hình học sâu có thể khai thác được các đặc trưng hữu ích từ dữ liệu thô, nhóm đã thiết kế và triển khai nhiều chiến lược xử lý dữ liệu khác nhau, nhằm đánh giá mức độ ảnh hưởng của từng kỹ thuật đến hiệu quả cuối cùng của mô hình. Mục tiêu chính là kiểm nghiệm xem liệu việc làm sạch, bổ sung đặc trưng, hoặc tăng độ đầy đủ dữ liệu thông qua nội suy có cải thiện được hiệu quả của mô hình Transformer trong phân loại tín hiệu giao dịch hay không.

#### Cách 1: Chạy chạy với dữ liệu thô

- **Mô tả:** Dữ liệu gốc được đưa trực tiếp vào mô hình sau khi mã hóa nhãn (*SELL, HOLD, BUY*  $\rightarrow$  0, 1, 2).
- **Ưu điểm:** Nhanh chóng thử nghiệm mô hình đầu tiên, không mất nhiều công xử lý.
- **Vấn đề gặp phải:** Kết quả huấn luyện rất kém, do mô hình không học được thông tin chuỗi. Chuỗi ngắn khiến đầu vào không đồng nhất, gây lỗi hoặc sai lệch mạnh trong học sâu.

## Cách 2: Loại bỏ chuỗi không đủ độ dài

- **Mô tả:** Chỉ giữ lại các đoạn dữ liệu liên tục có độ dài tối thiểu bằng *seq\_len = 128*.
- **Ưu điểm:** Giảm thiểu lỗi trong quá trình huấn luyện, mô hình học được đúng định dạng chuỗi.
- **Vấn đề gặp phải:** Mất rất nhiều dữ liệu, đặc biệt là ở các thời điểm chuyển phiên giao dịch hoặc các ngày có sự kiện thị trường. Gây mất cân bằng dữ liệu và giảm khả năng khái quát hóa.

## Cách 3: Nội suy tuyến tính để điền giá trị thiếu

- **Mô tả:** Dữ liệu thiếu được nội suy bằng phương pháp tuyến tính. Các đặc trưng sau đó được chuẩn hóa toàn cục.
- **Ưu điểm:** Giữ lại được gần như toàn bộ dữ liệu ban đầu.
- **Vấn đề gặp phải:** Nội suy toàn cục khiến mô hình gặp hiện tượng “data leakage” khi chuẩn hóa cả các phần chưa được quan sát trong chuỗi hiện tại. Đồng thời, việc chuẩn hóa toàn bộ làm mất đi tính cục bộ theo thời gian, gây khó khăn cho mô hình trong việc học các biến động ngắn hạn.

### 4.3. Cách xử lý cuối cùng của nhóm

Sau khi thử nghiệm và đánh giá hiệu quả các cách xử lý khác nhau, nhóm đã thống nhất sử dụng một quy trình xử lý dữ liệu cụ thể bao gồm các bước sau:

#### 4.3.1. Mã hóa nhãn

Các nhãn định hướng giao dịch được chuyển từ dạng chuỗi sang số nguyên theo quy tắc: *SELL, HOLD, BUY*  $\rightarrow$  0, 1, 2

```
label_map = {'SELL': 0, 'HOLD': 1, 'BUY': 2}
df['Label'] = df['Label'].map(label_map)
```

### 4.3.2. Trích xuất đặc trưng thời gian

Từ hai trường *Date* và *Time*, nhóm tạo ra timestamp hợp nhất, sau đó trích xuất ba đặc trưng thời gian chuẩn hóa trong khoảng  $[0, 1]$ :

- **hour**: giờ trong ngày, chia cho 23.
- **day\_of\_week**: thứ trong tuần, chia cho 6.
- **index**: chỉ số thứ tự dòng trong tập dữ liệu, chia cho tổng số dòng để chuẩn hóa.

```
df['timestamp'] = pd.to_datetime(df['Date'] + ' ' + df['Time'])
df['hour'] = df['timestamp'].dt.hour / 23
df['day_of_week'] = df['timestamp'].dt.dayofweek / 6
df['index'] = np.arange(len(df)) / len(df)
df = df.drop(columns=['Date', 'Time', 'timestamp'])
```

### 4.3.4. Phân chia tập dữ liệu và khởi tạo tập huấn luyện

Các tập dữ liệu ban đầu sẽ được chuyển đổi thành chuỗi thời gian cố định với độ dài  $seq\_len = 64$ , nhả tại thời điểm kế tiếp làm đầu ra. Dữ liệu được sinh theo generator, đóng gói thành *tf.data.Dataset* để huấn luyện hiệu quả hơn với TensorFlow.

```
def sequence_generator(df, seq_len, feature_cols):
    features = df[feature_cols].values.astype('float32')
    labels = df['Label'].values.astype('int32')

    for i in range(seq_len, len(features)):
        x = features[i - seq_len:i]
        y = labels[i]
        yield x, y

def dataset_from_sequences(df, seq_len, feature_cols):
    output_signature = (
        tf.TensorSpec(shape=(seq_len, len(feature_cols)),
            dtype=tf.float32),
        tf.TensorSpec(shape=(), dtype=tf.int32)
    )

    return tf.data.Dataset.from_generator(
        lambda: sequence_generator(df, seq_len, feature_cols),
```

```

        output_signature=output_signature
    )

def create_dataset(df, seq_len, batch_size, feature_cols,
shuffle=True):
    ds = dataset_from_sequences(df, seq_len, feature_cols)
    if shuffle:
        ds = ds.shuffle(buffer_size=10000)
    return ds.batch(batch_size).prefetch(tf.data.AUTOTUNE)

train_ds = create_dataset(train_data, seq_len, batch_size,
feature_cols, shuffle=True)
val_ds = create_dataset(val_data, seq_len, batch_size,
feature_cols, shuffle=False)
test_ds = create_dataset(test_data, seq_len, batch_size,
feature_cols, shuffle=False)

```

Phương pháp xử lý cuối cùng mà nhóm áp dụng đảm bảo các đặc trưng đầu vào được chuẩn hóa đồng đều, giàu thông tin thời gian, không gây rò rỉ dữ liệu, và phù hợp với yêu cầu chuỗi cố định của các mô hình học sâu như Transformer. Tập dữ liệu sau xử lý có tính liên tục cao, ổn định và giúp mô hình học tốt hơn các quy luật ẩn trong biến động thị trường.

## 5. XÂY DỰNG MÔ HÌNH (MODELING)

Trong bài toán phân loại hành động giao dịch (BUY, SELL, HOLD) từ dữ liệu chuỗi thời gian tài chính theo phút, nhóm lựa chọn mô hình **Transformer** làm nền tảng do khả năng học tốt các mối quan hệ dài hạn, phi tuyến và không phụ thuộc trình tự thời gian cố định như LSTM. Trên cơ sở này, nhóm lần lượt xây dựng ba mô hình từ đơn giản đến phức tạp hơn, với mục tiêu tăng dần khả năng khai thác thông tin thời gian và đặc trưng thị trường.

### 5.1. Mô hình 1 – Transformer cơ bản

Mô hình đầu tiên chỉ sử dụng các đặc trưng cơ bản của dữ liệu thị trường gồm giá (*Open*, *High*, *Low*, *Close*) và khối lượng (*Volume*). Dữ liệu đầu vào được chia thành các chuỗi có độ dài cố định *seq\_len*, chuẩn hóa theo từng khối nhằm đảm bảo tính ổn định về phân bố.

## Kiến trúc tổng thể

- **Input:** chuỗi thời gian đầu vào (64, 5) đặc trưng
- **4 tầng Transformer Encoder:**
  - Multi-Head Attention (6 heads, key\_dim = 64)
  - Residual connection + Layer Normalization
  - Feedforward network với hàm kích hoạt ReLU
- **Output head:**
  - Global Average Pooling → Dense(64, ReLU) → Dropout → Dense(3, Softmax)

Mô hình này đóng vai trò như baseline, giúp đánh giá hiệu quả học từ dữ liệu giá gốc mà không cần thông tin thời gian hay tín hiệu kỹ thuật nâng cao.

## 5.2. Mô hình 2: Transformer + Time2Vec

Để giúp mô hình học được các yếu tố liên quan đến thời gian, nhóm bổ sung thêm ba đặc trưng thời gian vào đầu vào:

- *hour*: giờ trong ngày (chuẩn hóa từ 0 đến 1)
- *day\_of\_week*: thứ trong tuần (0 đến 1)
- *index*: vị trí tương đối trong chuỗi dữ liệu

Ba đặc trưng này được đưa vào lớp **Time2Vec** để học biểu diễn thời gian dưới dạng tổng hợp tuyến tính và hàm sin.

## Kiến trúc tổng thể

- **Input:** (64, 8) – *OHLCV* + *hour* + *day\_of\_week* + *index*
- **Time2Vec Layer** (kernel size = 16) áp dụng cho 3 đặc trưng cuối → tạo đầu ra (64, 16)
- **Concatenate** đầu vào gốc và đầu ra Time2Vec → (64, 5+16)
- **4 tầng Transformer Encoder**
- **Output head** giống mô hình 1

Mô hình này có khả năng học tốt hơn các yếu tố tuần hoàn hoặc xu hướng phụ thuộc theo thời gian.



### 5.3. Mô hình 3: Transformer + Time2Vec + Chỉ báo kỹ thuật + Block Normalization

Trong mô hình thứ ba – và cũng là mô hình mạnh nhất – nhóm kết hợp thêm các **chỉ báo kỹ thuật (technical indicators)**, được tính toán từ các trường *Open*, *High*, *Low*, *Close*, và *Volume* thông qua thư viện *ta*.

Các nhóm chỉ báo sử dụng:

Nhóm chỉ báo	Tên chỉ báo
Momentum	RSI, ROC, Chande Momentum Oscillator (CMO), Williams %R
Volatility	ATR, Bollinger Bands (Mid, Upper, Lower, Bandwidth), Keltner Channels, Donchian Channels
Trend	SMA (20), EMA (20), MACD, MACD Histogram, DPO, Mass Index
Volume	OBV, Chaikin Money Flow (CMF), Accumulation/Distribution (AD), Force Index, Money Flow Index (MFI)

Tất cả các chỉ báo được thêm vào dataframe bằng hàm `add_all_indicators()`, sau đó được **fill giá trị thiếu** theo chiều thời gian (*bfill*, *ffill*) để đảm bảo không có giá trị NaN trong dữ liệu huấn luyện.

Đặc biệt, mô hình này nhóm áp dụng thêm kỹ thuật **Chuẩn hóa theo từng khối (Block Normalization)**

Tập dữ liệu được chia thành các khối liên tiếp có độ dài bằng `seq_len = 64`. Mỗi khối được chuẩn hóa độc lập bằng `StandardScaler`, nhằm đảm bảo tính ổn định và cục bộ của dữ liệu đầu vào.

```
def normalize_by_blocks(data, block_size) :  
    result = np.zeros_like(data)  
  
    for start_idx in range(0, len(data), block_size):  
        block = data[start_idx:start_idx + block_size]  
        scaler = StandardScaler()
```

```

        normalized_block = scaler.fit_transform(block)
        result[start_idx:start_idx + block_size] =
normalized_block

    return result

```

Hàm này chia dữ liệu thành các khối và chuẩn hóa mỗi khối một cách độc lập, giúp cải thiện tính ổn định và hiệu quả của mô hình khi xử lý chuỗi thời gian có tính biến động cao.

### Kiến trúc tổng thể

- **Input:** (64, ~35+) – gồm *OHLCV*, đặc trưng thời gian, và các chỉ báo kỹ thuật
- **Time2Vec** vẫn áp dụng cho *hour*, *day\_of\_week*, *index*
- **Concatenate** toàn bộ đặc trưng với đầu ra Time2Vec
- **4 tầng Transformer Encoder:**
  - **Multi Head Attention**(num\_heads=6, key\_dim=64)
  - **Feedforward:** Dense(128, ReLU)
- **Output:** GlobalAveragePooling1D → Dense(64, ReLU) → Dropout(0.1) → Dense(3, Softmax)

Mô hình này giúp kết hợp cả ba thành phần:

- Dữ liệu gốc
- Thời gian tuần hoàn
- Kiến thức từ phân tích kỹ thuật – vốn đã được sử dụng rộng rãi trong thực tiễn giao dịch.

### 5.4. Huấn luyện và tối ưu hóa

Tất cả các mô hình đều được huấn luyện với cấu hình thống nhất:

- **Batch size:** 64
- **Epochs:** 15 (có *Early Stopping* với *patience*=5)
- **Optimizer:** *Adam*, *learning rate* = 0.0001
- **Loss function:** *Sparse Categorical Cross Entropy*
- **Metrics:** Accuracy

Mô hình được đánh giá trên tập validation và test để chọn ra phiên bản tốt nhất.

## 6. ĐÁNH GIÁ MÔ HÌNH (EVALUATION)

Để đánh giá hiệu quả của mô hình phân loại tín hiệu giao dịch, nhóm sử dụng hai phương pháp chính:

- **Đánh giá chất lượng phân loại** thông qua độ chính xác trên tập kiểm tra.
- **Đánh giá thực tế hiệu suất giao dịch** bằng cách kiểm thử chiến lược trên dữ liệu thị trường lịch sử (backtesting).

### 6.1. Độ chính xác phân loại (Classification Accuracy)

Mô hình được đánh giá trên tập kiểm tra (*test\_ds*) với ba nhãn phân loại: BUY (2), SELL (0), và HOLD (1). Độ chính xác (accuracy) được tính bằng tỷ lệ số mẫu được dự đoán đúng so với tổng số mẫu.

Kết quả độ chính xác của từng mô hình như sau

Mô hình	Accuracy trên tập kiểm tra
Transformer cơ bản	47.23%
Transformer + Time2Vec	48.54%
Transformer + Time2Vec + Indicators + Block Normalization	61.94%

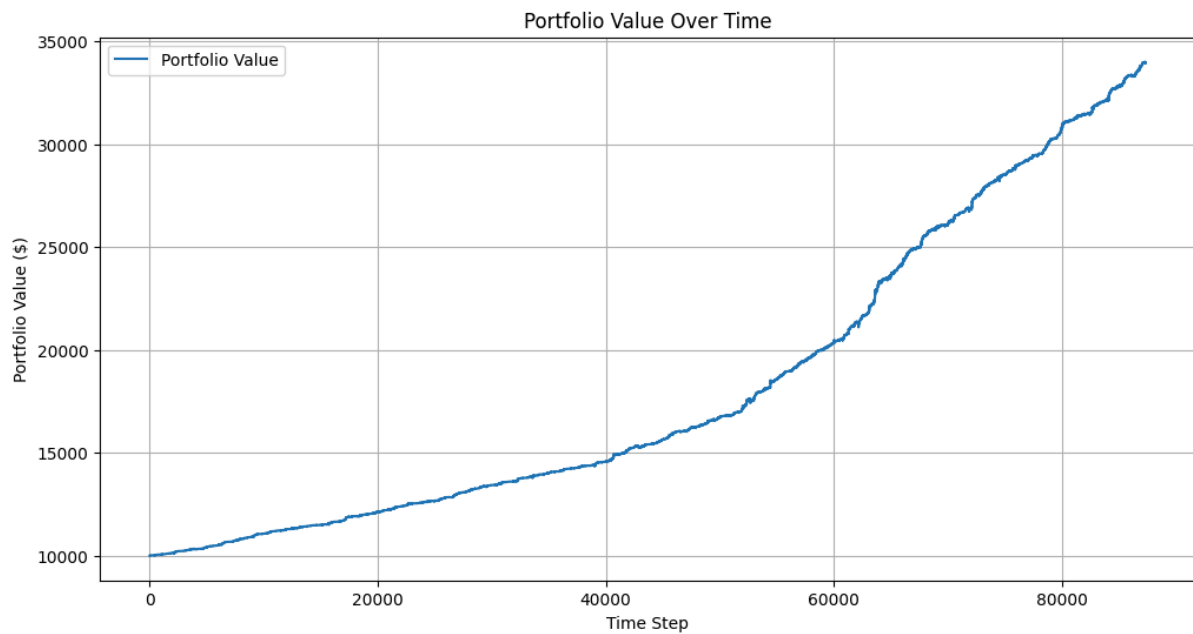
### 6.2. Backtesting

Để đánh giá khả năng áp dụng mô hình vào giao dịch thực tế, nhóm tiến hành kiểm thử chiến lược đơn giản với mô hình **Transformer + Time2Vec + Indicators + Block Normalization** trên dữ liệu lịch sử: mô phỏng hành động giao dịch dựa trên nhãn dự đoán tại mỗi thời điểm.

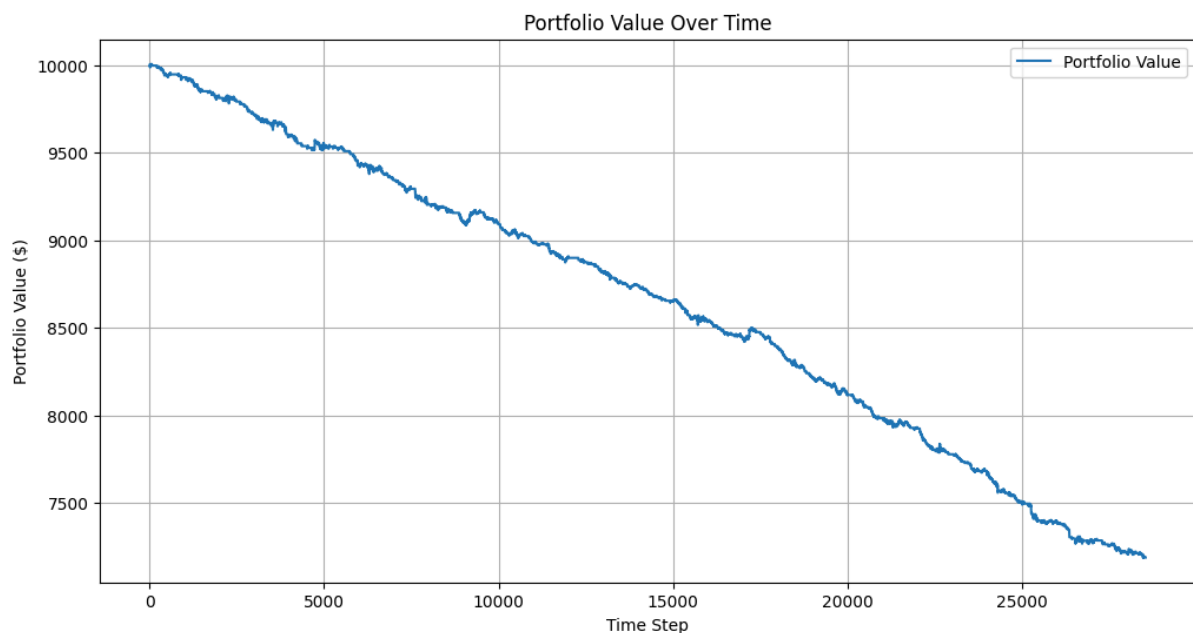
Kết quả

- Backtesting tháng 1-3 năm 2022 với cost = 0%:
- Initial Cash: \$10000.00
- Final Portfolio Value: \$33955.14
- Total Return: 239.55%
- Sharpe Ratio: 1.04

- Maximum Drawdown: -1.28%
- Number of Trades: 1822



- Backtesting tháng 1 năm 2022 với cost = 0.1%:
- Initial Cash: \$10000.00
- Final Portfolio Value: \$7189.17
- Total Return: -28.11%
- Sharpe Ratio: -0.95
- Maximum Drawdown: -28.16%
- Number of Trades: 636



### 6.3. Nhận xét

Từ kết quả của bài toán, nhóm rút ra các kết luận sau:

- Mô hình cuối cùng (**Transformer + Time2Vec + Indicators + Block Normalization**) không chỉ đạt độ chính xác cao nhất trên tập kiểm tra, mà còn mang lại hiệu quả sinh lời rõ rệt trong môi trường giả lập giao dịch, vượt trội hơn so với các mô hình đơn giản.
- Việc kết hợp các đặc trưng thời gian và chỉ báo kỹ thuật giúp mô hình hiểu được ngữ cảnh giao dịch và tín hiệu thị trường, từ đó đưa ra quyết định phù hợp hơn với hành vi thực tế.
- **Block Normalization** là một thành phần quan trọng trong pipeline xử lý dữ liệu, giúp mô hình hoạt động ổn định và chính xác hơn trên chuỗi thời gian phi tuyến, nhiều biến động như dữ liệu giá cổ phiếu theo phút.
- **Backtest** cho thấy mô hình đạt lợi nhuận cao (239.55%) và Sharpe Ratio tốt (1.04) khi không có phí giao dịch. Tuy nhiên, khi thêm phí giao dịch (0.1%), lợi nhuận giảm mạnh (-28.11%) và Sharpe Ratio trở nên âm (-0.95).
- Nguyên nhân chính là do tần suất giao dịch cao, dẫn đến chi phí giao dịch tích lũy nhanh và làm giảm lợi nhuận. Với hơn 600 giao dịch trong 1 tháng, mỗi khoản chi phí làm ảnh hưởng lớn đến tài sản. Để tối ưu hóa, việc cải thiện việc quản lý chi phí giao dịch có thể giúp tối đa hóa hiệu quả trong môi trường thực tế, từ đó nâng cao lợi nhuận bền vững.

## 7. TRIỂN KHAI VÀ ỨNG DỤNG (DEPLOYMENT)

### 7.1. Kiến trúc triển khai tổng thể

Hệ thống đề xuất có thể được triển khai theo cấu trúc pipeline giao dịch tự động, bao gồm các khối chức năng chính:

- **Dòng dữ liệu thời gian thực (Real-time Data Feed)**
  - Dữ liệu giá cổ phiếu theo phút có thể được cập nhật thông qua các API như IEX Cloud, Alpaca, Polygon.io hoặc sàn giao dịch thực tế.
  - Thông tin bao gồm *Open*, *High*, *Low*, *Close*, *Volume* tại mỗi phút.
- **Tiền xử lý trực tuyến (Online Preprocessing)**

- Dữ liệu mới được nối vào sliding window 120 phút gần nhất để tạo đầu vào cho mô hình.
- Tính toán các đặc trưng như phần trăm thay đổi giá, chuẩn hóa bằng thống kê đã học trong tập huấn luyện, và cập nhật thời gian t.
- Kiểm tra liên tục xem dữ liệu có đảm bảo liên tục theo phút không. Nếu không, mô hình sẽ tạm dừng đưa ra dự đoán để tránh sai lệch.
- **Suy luận mô hình (Model Inference)**
  - Mô hình Transformer nhận đầu vào là chuỗi liên tục 120 phút và dự đoán nhãn tại thời điểm hiện tại: BUY, SELL hoặc HOLD.
  - Dự đoán có thể được kết hợp với chiến lược quản trị rủi ro trước khi sinh lệnh giao dịch thực tế.
- **Thực thi giao dịch (Order Execution)**
  - Nếu tín hiệu đủ mạnh, hệ thống sẽ gửi lệnh giao dịch qua API của nhà môi giới.
  - Hệ thống có thể kết hợp thêm logic như: xác nhận xu hướng từ nhiều tín hiệu, điều kiện stop-loss/take-profit, hoặc quy tắc xác suất đầu tư (position sizing).

## 7.2. Ứng dụng trong thực tiễn giao dịch

Hệ thống có thể được sử dụng trong nhiều tình huống thực tế:

- **Giao dịch nội nhật (Intraday Trading):** Với dữ liệu theo phút và mô hình được huấn luyện trên biến động ngắn hạn, hệ thống rất phù hợp cho chiến lược giao dịch trong ngày (mua và bán trong cùng một phiên).
- **Tư vấn đầu tư bán tự động (Signal-based Advisory):** Mô hình có thể được dùng như một engine tạo tín hiệu giao dịch, kết hợp với quyết định của con người.
- **Giao dịch tự động toàn phần (Fully Automated Trading):** Trong môi trường kiểm soát (sandbox hoặc tài khoản nhỏ), hệ thống có thể được triển khai để ra quyết định giao dịch hoàn toàn tự động theo thời gian thực.

## 7.3. Các yếu tố kỹ thuật cần cân nhắc khi triển khai

Một số vấn đề cần được xử lý kỹ lưỡng trong môi trường thực tế:

- **Độ trễ (Latency):** Việc cập nhật dữ liệu, chuẩn hóa, suy luận mô hình và gửi lệnh cần được tối ưu để duy trì phản hồi <1 giây.

- **Tính ổn định và liên tục:** Hệ thống phải phát hiện được các đoạn dữ liệu không liên tục và tránh đưa ra quyết định khi chất lượng dữ liệu không đảm bảo.
- **Quản trị rủi ro:** Kết hợp các lớp kiểm soát như giới hạn lệnh/ngày, tỉ lệ lỗ tối đa, hoặc mô hình xác suất (confidence-based decisions).
- **Khả năng kiểm thử và giám sát (monitoring):** Ghi log tất cả tín hiệu, dự đoán và quyết định để phục vụ đánh giá, kiểm thử A/B và phát hiện lỗi.

#### 7.4. Khả năng mở rộng hệ thống

Hệ thống có thể được mở rộng theo nhiều chiều:

- **Đa mã cổ phiếu (multi-asset):** Áp dụng mô hình cho nhiều mã cùng lúc, với dữ liệu độc lập hoặc chung (multi-head Transformer).
- **Đa khung thời gian (multi-timeframe):** Kết hợp dữ liệu M1 với M5 hoặc M15 để đưa ra quyết định nhiều tầng (hierarchical decision).
- **Triển khai theo microservice:** Mỗi khối (data feed, model inference, order logic) là một service độc lập, dễ mở rộng và bảo trì.
- **Cập nhật mô hình định kỳ (retraining):** Triển khai cơ chế học lại mô hình theo tháng, quý hoặc bằng chiến lược online learning (nếu có nhãn mới).

## 8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN (CONCLUSIONS & NEXT STEPS)

Trong bài tập này, nhóm đã xây dựng thành công một hệ thống giao dịch định lượng sử dụng mô hình học sâu Transformer để phân loại các thời điểm giao dịch theo ba trạng thái: BUY, SELL, và HOLD. Mô hình được huấn luyện trên dữ liệu giá cổ phiếu theo phút, được xử lý cẩn thận theo cấu trúc chuỗi thời gian trượt, kết hợp với các đặc trưng kỹ thuật quan trọng và mã hóa thời gian thông qua Time2Vec.

Về mặt kỹ thuật, mô hình Transformer tỏ ra đặc biệt hiệu quả trong việc nắm bắt các mối liên hệ dài hạn giữa các thời điểm trong chuỗi giá. Việc tích hợp thêm các chỉ báo kỹ thuật cùng với các kỹ thuật tiền xử lý khác nhau giúp nâng cao đáng kể khả năng phân loại của mô hình. Kết quả đánh giá cho thấy mô

hình đạt được hiệu suất phân loại tốt, đồng thời tạo ra các tín hiệu giao dịch có khả năng sinh lời khi áp dụng vào chiến lược backtest.

Tuy nhiên, vẫn còn nhiều khía cạnh cần được cải thiện và phát triển thêm. Trong tương lai, hệ thống có thể được mở rộng theo các hướng sau:

- **Nâng cao kiến trúc mô hình:** Tối ưu hóa cấu trúc Transformer (số tầng, attention heads), thử nghiệm với các mô hình khác như Quantformer, Informer, hoặc kết hợp Transformer và LSTM để khai thác cả đặc tính tuần tự và phụ thuộc dài hạn.
- **Tích hợp thêm dữ liệu ngoài:** Kết hợp với các nguồn dữ liệu phi cấu trúc như tin tức tài chính, dữ liệu vĩ mô, hoặc tín hiệu tâm lý thị trường từ mạng xã hội để bổ sung chiều sâu ngữ nghĩa cho hệ thống.
- **Giao dịch đa tài sản và danh mục:** Mở rộng hệ thống để quản lý và tối ưu danh mục gồm nhiều mã cổ phiếu hoặc tài sản khác nhau, từ đó nâng cao khả năng ứng dụng thực tế trong quản lý quỹ.
- **Triển khai thực tế và học liên tục:** Xây dựng pipeline triển khai đầy đủ, kết hợp học online (online learning) hoặc học lại định kỳ để mô hình thích ứng với sự thay đổi liên tục của thị trường.

Tổng thể, bài tập này đã khẳng định tiềm năng ứng dụng của Transformer và học sâu trong giao dịch tài chính tần suất cao, mở ra nhiều cơ hội mới cho việc phát triển các hệ thống giao dịch thông minh, hiệu quả và tự động hóa cao trong kỷ nguyên dữ liệu lớn.

## 9. TÀI LIỆU THAM KHẢO (REFERENCES)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin: *Attention is All you Need*

Aniket Kumar; Rodrigue Rizk; KC Santosh: *Transformer-based Reinforcement Learning Model for Optimized Quantitative Trading*

Z Zhang, B Chen, S Zhu, N Langrené: *Quantformer: from attention to profit with a quantitative transformer trading strategy*

P Bilokon, Y Qiu: *Transformers versus LSTMs for electronic trading*



L Berti, G Kasneci: *TLOB: A Novel Transformer Model with Dual Attention for Stock Price Trend Prediction with Limit Order Book Data*

Adams Wei Yu, Lei Huang, Qihang Lin, Ruslan Salakhutdinov, Jaime Carbonell: *Block-Normalized Gradient Method: An Empirical Study for Training Deep Neural Network*

EH Cuthill, RS Varga: *A method of normalized block iteration*

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, Marcus Brubaker: *Time2Vec: Learning a Vector Representation of Time*

Manish Agrawal, Piyush Kumar Shukla, Rajit Nair, Anand Nayyar, and Mehedi Masud: *Stock Prediction Based on Technical Indicators Using Deep Learning Model*

Manuel R. Vargas; Carlos E. M. dos Anjos; Gustavo L. G. Bichara; Alexandre G. Evsukoff: *Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles*