TRƯỜNG ĐẠI HỌC THỦY LỢI

PHÂN HIỆU

SINH VIÊN THỰC HIỆN: CHỬ TRẦN PHƯƠNG NAM

ĐỀ TÀI: DỰ ĐOÁN BIẾN ĐỘNG GIÁ ĐÓNG CỬA CỔ PHIẾU NGÀNH CÔNG NGHỆ VÀ NĂNG LƯỢNG TẠI VIỆT NAM GIAI ĐOẠN 2020 - 2025

GIẢNG VIÊN HƯỚNG DẪN: ThS Vũ Thị Hạnh

TP. Hồ Chí Minh - 09/2025

MỤC LỤC

DANH MỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Trong bối cảnh nền kinh tế Việt Nam và thế giới nói chung đang chuyển dịch mạnh mẽ theo hướng hiện đại hóa và bền vững, hai ngành công nghệ và năng lượng nổi lên như những trụ cột then chốt, vừa hỗ trợ lẫn nhau, vừa tạo động lực phát triển chung cho xã hội. Ngành công nghệ cung cấp nền tảng để thu thập, xử lý và phân tích dữ liệu khổng lồ, trong khi ngành năng lượng đóng vai trò bảo đảm nguồn cung cho quá trình công nghiệp hóa và chuyển đổi số quốc gia.

Đặc biệt, trong lĩnh vực năng lượng, Việt Nam đang định hướng giảm dần sự phụ thuộc vào các nguồn truyền thống như thủy điện và nhiệt điện**,** đồng thời nghiên cứu và triển khai năng lượng điện hạt nhân là nguồn điện xanh và bền vững như một giải pháp thay thế dài hạn. Sự thay đổi này không chỉ tác động sâu rộng đến cấu trúc ngành năng lượng, mà còn ảnh hưởng trực tiếp tới các doanh nghiệp công nghệ. Do đó, việc phân tích sự tương tác giữa hai nhóm ngành này trên thị trường chứng khoán không chỉ giúp nhà đầu tư nhận diện đúng hướng mà còn cho thấy được tính cấp thiết về việc phát triển ngành năng lượng song song với việc phát triển đất nước.

Trong bối cảnh đó, đề tài "Thiết kế hệ thống data pipeline kết hợp dự đoán và phân tích giá cổ phiếu (giới hạn top 5 mã cổ phiếu thuộc nhóm ngành công nghệ và năng lượng tại Việt Nam từ năm 2020 đến 2025)" được xây dựng với mục tiêu không chỉ dự báo giá cổ phiếu riêng lẻ, mà còn xem xét mối quan hệ ảnh hưởng qua lại giữa hai ngành. Thông qua việc khai thác dữ liệu lịch sử giao dịch cùng các nguồn tin tức báo chí, đề tài sẽ đánh giá mức độ tác động của biến động ngành năng lượng nói chung đối với ngành công nghệ.

Với sự hỗ trợ của các công cụ hiện đại như dbt**,** Apache Airflow và PostgreSQL, hệ thống được thiết kế nhằm tự động hóa quá trình thu thập – xử lý – lưu trữ dữ liệu, đồng thời kết hợp các kỹ thuật dự báo chuỗi thời gian và xử lý ngôn ngữ tự nhiên (NLP) để gán nhãn và phân tích tin tức. Qua đó, đề tài hướng đến việc xây dựng một mô hình vừa có giá trị học thuật trong việc minh chứng khả năng tích hợp dữ liệu đa nguồn, vừa mang giá trị ứng dụng trong phân tích tài chính.

Chương 1 Giới thiệu đề tài

1.1 Tổng quan đề tài

Trong những năm gần đây, thị trường chứng khoán Việt Nam nói riêng phát triển nhanh chóng và trở thành một trong những kênh đầu tư quan trọng của nền kinh tế. Giá cổ phiếu thường xuyên biến động và chịu ảnh hưởng mạnh mẽ từ nhiều yếu tố khác nhau như kết quả kinh doanh của doanh nghiệp, xu hướng ngành, tình hình vĩ mô và đặc biệt là các thông tin từ báo chí và truyền thông. Do đó, việc phân tích và dự báo xu hướng giá cổ phiếu là một nhu cầu cấp thiết đối với riêng các nhà đầu tư chứng khoán nhằm đưa ra quyết định chính xác, nhanh.

Nhìn chung, sự giao động của giá cổ phiếu chịu ảnh hưởng bởi rất nhiều các yếu tố khác nhau. Chính ví thế, hiện nay bài toán dự báo giá cổ phiếu vẫn gặp nhiều khó khăn. Đầu tiên, dữ liệu thường phân tán và không đồng bộ: dữ liệu lịch sử giao dịch được lưu trữ tại các sàn, trong khi tin tức và thông tin thị trường tồn tại dưới dạng văn bản phi cấu trúc từ nhiều nguồn khác nhau. Tiếp đến, các phương pháp dự báo truyền thống chủ yếu dựa vào phân tích kỹ thuật hoặc kinh nghiệm của nhà đầu tư, dẫn đến độ chính xác còn hạn chế. Sau cùng, việc thiếu các hệ thống pipeline tự động hóa xử lý dữ liệu khiến quá trình phân tích trở nên rời rạc, tốn nhiều thời gian và công sức.

Trong bối cảnh đó, sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI)**,** đặc biệt là các lĩnh vực Machine Learning (ML) và Deep Learning (DL)**,** đang mở ra nhiều cơ hội mới. Sự đóng góp của AI đã giúp các nhà đầu tư có một bức tranh tổng quan về dự liệu, từ dữ liệu giao dịch đến dữ liệu phi cấu trúc như tin tức, bài báo, báo cáo tài chính, thông tin trên mạng xã hội, đã tạo ra nguồn dữ liệu phong phú và đa dạng cho các mô hình dự báo. Mặt khác, các thuật toán ML/DL liên tục được cải tiến về cả độ chính xác lẫn khả năng tối ưu, cho phép khai phá dữ liệu hiệu quả hơn, xử lý đồng thời nhiều loại dữ liệu khác nhau và đưa ra kết quả dự báo có độ tin cậy cao hơn.

1.2 Cở sở hình thành đề tài

- Thực tiễn

Thị trường chứng khoán Việt Nam ngày càng đóng vai trò quan trọng trong huy động vốn và phản ánh sức khỏe của nền kinh tế. Đặc biệt, trong bối cảnh Việt Nam đang đẩy mạnh chuyển đổi năng lượng xanh, các nguồn năng lượng với xu hướng phát triển bền vững, nghiên cứu phát triển điện hạt nhân song song với giảm dần sự phụ thuộc vào thủy điện và nhiệt điện, sự biến động của ngành năng lượng có tác động trực tiếp đến các ngành

công nghiệp khác, trong đó ảnh hưởng tới ngành công nghệ là rất lớn. Do đó, việc dự báo và phân tích giá cổ phiếu trong nhóm ngành công nghệ và năng lượng không chỉ giúp nhà đầu tư nắm bắt xu hướng, mà còn hỗ trợ đưa ra các chính sách và chiến lược phát triển doanh nghiệp phù hợp với tình hình hiện tại.

– Công nghệ

Sự phát triển của các công cụ hiện đại đã tạo điều kiện thuận lợi để xây dựng hệ thống xử lý dữ liệu lớn phục vụ mô hình dự báo. Các nền tảng ETL/ELT như dbt giúp chuẩn hóa và biến đổi dữ liệu linh hoạt, kết hợp với Apache Airflow đảm nhiệm việc lập lịch và điều phối pipeline một cách tự động quy trình lấy dữ liệu. Hệ quản trị cơ sở dữ liệu PostgreSQL cung cấp khả năng lưu trữ và truy vấn dữ liệu hiệu quả, hỗ trợ tốt cho việc huấn luyện mô hình học sâu. Khi kết hợp với các mô hình tiên tiến như LSTM**,** GRU và FCNN, VAE,… hệ thống không chỉ giải quyết được bài toán dự báo chuỗi thời gian mà còn khai thác hiệu quả dữ liệu văn bản từ tin tức tài chính.

1.3 Mục tiêu đề tài

1.3.1 Mục tiêu tổng quát

Đề tài hướng tới việc xây dựng model dự đoán xu hướng biến động giá cổ phiếu trong ngắn hạn khi kết hợp các mô hình học sâu VAE (Variational AutoEncoder) với kiến trúc Seq2Seq (Sequence to Sequence) để dự báo và phân tích giá cổ phiếu. Hệ thống tập trung vào việc dự báo xu hướng biến động giá giúp nhà đầu tư ngắn hạn có thể xác định được đỉnh và đáy của từng mã cổ phiếu. Từ đó có một cái nhìn tổng quan về xu hướng phát triển, ảnh hưởng qua lại của 2 nhóm ngành công nghệ và năng lượng.

1.3.2 Mục tiêu cụ thể

Xây dựng mô hình dự báo:

– Baseline**:** Sử dụng 2 mô hình là TCN và MLP để làm mô hình baseline
– Mô hình chính**:** Triển khai mô hình VAE với việc kết hợp đồng thời kiến trúc Seq2Seq để nâng cao hiệu quả huấn luyện mô hình.

- Đánh giá hiệu quả mô hình**:** Thực hiện đánh giá bằng các chỉ số như RMSE**,** MAPE, MAE, DA@k, TA@k để đánh giá độ chính xác.
- So sánh kết quả giữa mô hình baseline và mô hình nâng cao để xác định mô hình tối ưu cho dữ liệu chứng khoán Việt Nam.

1.3.3 Đối tượng và phạm vi nghiên cứu

Phạm vi cổ phiếu: Nghiên cứu giới hạn trong top 5 mã cổ phiếu tiêu biểu thuộc nhóm ngành công nghệ và năng lượng trên thị trường chứng khoán Việt Nam (HOSE/HNX).

Nguồn dữ liệu

Dữ liệu định lượng: lịch sử giá cổ phiếu (ngày giao dich, giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất, khối lượng giao dịch).

Dữ liệu phi cấu trúc: tin tức, bài báo, thông tin do công ty đăng tải và trên các trang thông tin phổ biến ở Việt Nam.

Giới hạn nghiên cứu

Không bao quát toàn bộ thị trường, chỉ tập trung vào hai nhóm ngành chính.

Không phân tích các yếu tố kinh tế vĩ mô phức tạp (lãi suất, lạm phát…) ngoài dữ liệu giá và tin tức.

1.3.4 Phương pháp nghiên cứu

- **Thu thập dữ liệu**

Dữ liệu được thu thập từ các nguồn công khai và đáng tin cậy như trang thông tin điện tử của Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE), Sở Giao dịch Chứng khoán Hà Nội (HNX). Quá trình thu thập được thực hiện bằng thư viện Vnstock, một thư viện cho phép lấy dữ liệu trên các sàn chứng khoán của Việt Nam thông qua API sử dụng trên nên ngôn ngữ Python, sau đó dữ liệu được lưu trữ dưới định dạng CSV.

- **Xử lý dữ liệu**

Sau khi thu thập, dữ liệu được tiến hành làm sạch nhằm loại bỏ các giá trị thiếu, dữ liệu nhiễu hoặc bất thường. Tiếp đó, dữ liệu được chuẩn hoá về định dạng thời gian và giá trị để bảo đảm tính nhất quán. Đồng thời thực hiện tính các chỉ số kỹ thuật phục vụ cho việc huấn luyện mô hình. Các chỉ số sử dụng trong bài toán này gồm có: MA_5, MA_10, MA_20, EMA_5, EMA_10, EMA_20, MACD, ATR(14), RSI(14).

- **Phân tích và dự báo**

Trong đề tài này, với mục tiêu là dự đoán xu hướng biến động của các mã cổ phiếu trong phạm vị ngắn hạn từ 3 – 7 ngày. Mô hình sử dụng trong bài toán này là VAE Seq2Seq (Variational Autoencoder). Mô hình khử nhiễu hiệu quả phù hợp với dữ liệu có nhiều giao động ngẫu nhiên.

- **Phân tích kết quả**

Để đánh giá hiệu xuất của mô hình, trong bài này sử dụng các chỉ số định lượng phổ biến như RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error) để đo lường độ chính xác về giá trị dự báo. Các chỉ số đánh giá này phù hợp với dữ mục tiêu dự đoán giá trị liên tục.Trong bài toán này sử dụng các chỉ số nhằm đánh giá mức độ chênh lệch giữa giá trị thực và giá trị dự đoán. Từ đó đánh giá xem mô hình nhận diện được xu hướng hiệu quả không.

1.3.5 Tính ứng dụng của đề tài

Kết quả nghiên cứu có thể được ứng dụng vào hoạt động phân tích và dự báo thị trường chứng khoán, hỗ trợ nhà đầu tư, doanh nghiệp và tổ chức tài chính trong việc đưa ra quyết định đầu tư chính xác hơn. Đặc biệt, đề tài tập trung vào hai nhóm ngành trọng yếu là công nghệ và năng lượng, trong đó năng lượng điện hạt nhân đang được Việt Nam nghiên cứu và triển khai trong thời gian tới với mục tiêu ngắn hạn, công nghệ là ngành tiêu thụ năng lượng lớn. Vì vậy, đề tài này không chỉ có giá trị tham khảo trong đầu tư mà còn góp phần gợi mở cho việc đánh giá sự tác động qua lại giữa hai ngành chiến lược của nền kinh tế.

1.4 Bài toán cần giải quyết

Dựa trên tình hình thực tiễn của thị trường chứng khoán Việt Nam nói riêng chịu ảnh hưởng của rất nhiều các yếu tố. Mục tiêu của nhà đầu tư là xác định được xu hướng của thị trường để đầu tư sinh lời. Những biến động về giá không theo nguyên tắc nên ta khi có càng nhiều dữ liệu thì chênh lệch về dự báo và thực tế càng nhỏ. Trong phạm vi của bài toán này đang dừng ở mức sử dụng dữ liệu lịch sử giao dịch trong phạm vi 5 năm từ năm 2020 đến 2025 đồng thời kết hợp với dữ liệu tin tức trên các website chính thức của từng công ty và cả những website tin tức chính thống lớn của Việt Nam. Để giải quyết bài toán này, mô hình sử dụng là VAE Seq2Seq nhằm phát huy tốt việc huấn luyện dữ liệu dạng Timeseries. Kết quả đầu ra của bài toán không phải là dự đoán chính xác xu hướng, giá tăng, giảm trong ngắn hạn. Mục tiêu của bài toán là giúp nhà đầu tư trung, dài hạn nhận diện được xu hướng biến động về giá của từng mã cổ phiếu trong 2 nhóm ngành là Năng lượng và Công nghệ. Kết quả mong muốn sau cùng là đường dự đoán sát với đường thực tế nhất, không quan trọng là dự đoán đúng hay sai hướng tăng, giảm tức là mô hình sẽ hoạt động tốt khi chỉ số đánh giá RMSE và MAPE nhỏ nhất.

Chương 2 Xử lý dữ liệu

2.1 Tiền xử lý dữ liệu tin tức

2.1.1 Tổng quan về dữ liệu

Dữ liệu tin tức được lấy từ các bài đăng trên website chính thức của các công ty. Phạm vi thời gian lấy bao gồm dữ liệu từ năm 2020 đến 2025

Số lượng bài viết trong từng mã như sau:

2.1.2 Tiền xử lý dữ liệu

- Hàm vi_sent_tokenize(text): Tách một chuỗi tiếng Việt thành danh sách câu, sử dụng thư viện Underthesea.

```python
def vi_sent_tokenize(text: str):
    text = (text or "").strip()
    if not text:
        return []
    # Ưu tiên: Underthesea
    try:
        from underthesea import sent_tokenize
        sents = sent_tokenize(text)
        return [re.sub(r"\s+", " ", s).strip() for s in sents if s and s.strip()]
    except Exception:
        pass
    # Dự phòng: regex dựa trên dấu câu + chữ in hoa/ngoặc/nháy
    txt = re.sub(r"\s+", " ", text)
    txt = re.sub(r'([\.!?…])(\s+)(?=[“"\']?[\(\[\{]*[A-ZÀÁÂÃÈÉÊÌÍÒÓÔÕÙÚĂĐĨŨƠÝ])',
                 r"\1\n", txt)
    return [s.strip() for s in txt.split("\n") if s.strip()]
```

Hình 1 Hàm vi_sent_tokenize

– Hàm explode_content: Chuyển DataFrame có cột văn bản thô (content) thành DataFrame với mỗi hàng là 1 câu, kèm định danh bài (article_id) và chỉ số câu (sent_idx).

```
def explode_content(df, text_col="content"):
    rows = []
    for idx, row in df.iterrows():
        article_id = f"art_{idx}"
        sents = vi_sent_tokenize(str(row.get(text_col, "") or ""))
        for i, s in enumerate(sents):
            rows.append({"article_id": article_id, "cau": s, "sent_idx": i})
    return pd.DataFrame(rows)
```

Hình 2 Hàm explode_content

2.1.3 Xây dựng mô hình PhoBERT

– Khởi tạo mô hình PhoBERT

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

MODEL_NAME = "wonrax/phobert-base-vietnamese-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)
model.eval()
```

Hình 3 Khởi tạo mô hình PhoBERT

– Hàm score_sentences_vi: Biến danh sách câu thành phân phối xác suất cảm xúc p_neg, p_neu, p_pos

Trong đó:

p_neg: chỉ số đánh giá điểm tiêu cục (> = – 1 và < 0)

p_neu: chỉ số đánh giá điểm trung tính (= 0)

p_pos: chỉ số đánh giá điểm tích cực (> 0 và <= 1)

```
def score_sentences_vi(texts, batch_size=32, max_length=256, device=None):
    device = device or ("cuda" if torch.cuda.is_available() else "cpu")
    model.to(device)
    if not texts:
        return np.zeros((0, 3), dtype=np.float32)

    probs_all = []
    with torch.no_grad():
        for i in range(0, len(texts), batch_size):
            batch = [str(t) if isinstance(t, str) and t.strip() else ""
                     for t in texts[i:i+batch_size]]
            enc = tokenizer(batch, return_tensors="pt",
                            padding=True, truncation=True,
                            max_length=max_length).to(device)
            logits = model(**enc).logits
            probs = torch.softmax(logits, dim=1).cpu().numpy()  # [neg, neu, pos]
            probs_all.append(probs)
    return np.vstack(probs_all)
```

Hình 4 Hàm đánh giá chỉ số cảm xúc trong câu

– Hàm compute_article_sentiment_from_df: tính trung bình chỉ số đánh giá trên cùng 1 bài báo.

```
def compute_article_sentiment_from_df(df_sent, article_col="article_id",
                                      text_col="cau"):
    rows = []
    for aid, g in df_sent.groupby(article_col, dropna=False):
        texts = g[text_col].astype(str).tolist()
        P = score_sentences_vi(texts)
        agg = np.array([1.0, 0.0, 0.0], dtype=np.float32) if P.shape[0] == 0 else
              P.mean(axis=0)
        compound = float(agg[2] - agg[0])
        rows.append({
            "article_id": aid,
            "p_neg": float(agg[0]),
            "p_neu": float(agg[1]),
            "p_pos": float(agg[2]),
            "compound": compound,
        })
    return pd.DataFrame(rows)
```

Hình 5 Hàm tính trung bình trên cùng bài báo

– Hàm compute_daily_sentiment_from_df: tính trung bình chỉ số đánh giá theo ngày

```
def compute_daily_sentiment_from_df(df_articles_with_date, date_col="date"):
    return (
        df_articles_with_date.groupby(date_col)[["p_neg", "p_neu", "p_pos"]]
        .mean()
        .reset_index()
    )
```

Hình 6 Hàm tính trung bình chỉ số đánh giá theo ngàyTiền xử lý dữ liệu giao dịch

2.2 Tiền xử lý dữ liệu giao dịch

2.2.1 Tổng quan về dữ liệu

Nguồn dữ liệu: tập dữ liệu gồm lịch sử giao dịch các mã cổ phiếu trong phạm vi 5 năm từ 2020 đến 2025. Trong đó, các mã cổ phiếu được chia theo 2 nhóm ngành là công nghệ và năng lượng.

Các mã nhóm ngành công nghệ: CMC, ELC, FPT, SGT, VNZ

Các mã nhóm ngành năng lượng: BSR, GAS, OIL, PLX, PVG

Dữ liệu lịch sử giao dịch có các thuộc tính sau:

Time: ngày giao dich

Open: giá mở cửa

High: giá cao nhất

Low: giá thấp nhất

Volume: số lượng cổ phiếu giao dịch

2.2.2 Tiền xử lý dữ liệu

Tính toán chỉ số kỹ thuật

- RSI: Đo lường tốc độ và mức thay đổi giá, dao động từ 0 đến 100.

Mục đích: Xác định tình trạng quá mua (RSI > 70) hoặc quá bán (RSI < 30)**.**

Công thức

$$\Delta P_t = P_t - P_{t-1}$$

$$\text{Average Gain} = \frac{\sum_{i=1}^{n} \text{Gain}_i}{n}$$

$$\text{Average Loss} = \frac{\sum_{i=1}^{n} \text{Loss}_i}{n}$$

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}}$$

$$RSI = 100 - \frac{100}{1 + RS}$$

Hình 7 Công thức tính RSI

```python
def _rsi(series, period=14):
    delta = series.diff()
    gain  = delta.clip(lower=0.0)
    loss  = (-delta).clip(lower=0.0)
    avg_gain = gain.ewm(alpha=1/period, min_periods=period, adjust=False).mean()
    avg_loss = loss.ewm(alpha=1/period, min_periods=period, adjust=False).mean()
    rs  = avg_gain / avg_loss
    rsi = 100 - (100 / (1 + rs))
    return rsi
```

Hình 8 Hàm tính toán chỉ số RSI

– Nhóm chỉ số đánh giá lợi suất (Returns): chênh lệch giá đóng cửa của phiên trước và phiên hiện tại

Công thức

Returns

$$\text{ret1}_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \quad \text{logret}_t = \ln\left(\frac{P_t}{P_{t-1}}\right), \quad \text{ret}_k = \frac{P_t - P_{t-k}}{P_{t-k}}$$

Chú thích: P_t: Giá đóng cửa tại thời điểm t. k: Số phiên (ví dụ: 5, 20).

Hình 9 Công thức tính các chỉ số đánh giá lợi suất

```
# --- Returns cơ bản ---
g["ret1"]   = g["close"].pct_change()
g["logret"] = np.log(g["close"] / g["close"].shift(1))
g["ret5"]   = g["close"].pct_change(5)
g["ret20"]  = g["close"].pct_change(20)
```

Hình 10 Nhóm chỉ số đánh giá lợi suất

– Nhóm chỉ số đánh giá trung bình động (MA/EMA)

Công thức

$$\mathrm{SMA}_k(t) = \frac{1}{k}\sum_{i=0}^{k-1} P_{t-i}$$

$$\mathrm{EMA}_k(t) = \alpha P_t + (1-\alpha)\,\mathrm{EMA}_k(t-1), \quad \alpha = \frac{2}{k+1}$$

$$\mathrm{ema\_dist}(t) = \frac{P_t - \mathrm{EMA}_{20}(t)}{\mathrm{EMA}_{20}(t)}, \quad \mathrm{ema60\_slope}(t) = \mathrm{EMA}_{60}(t) - \mathrm{EMA}_{60}(t-1)$$

Chú thích: k: Số phiên (5, 10, 20, 60).

Hình 11 Công thức đánh giá trung bình động (MA/EMA)

```
# --- EMA/MA & khoảng cách, độ dốc ---
g["ema20"]       = g["close"].ewm(span=20, adjust=False).mean()
g["ema60"]       = g["close"].ewm(span=60, adjust=False).mean()
g["ema20_dist"]  = (g["close"] - g["ema20"]) / g["ema20"]
g["ema60_slope"] = g["ema60"].diff()
```

Hình 12 Nhóm chỉ số đánh giá trung bình động (MA/EMA)

– Nhóm chỉ số biến động: đo lường độ biến động của giá và biên độ giao động trong từng phiên.

Công thức

$$\text{vol20}(t) = \sqrt{\frac{1}{20}\sum_{i=0}^{19}\left(\text{logret}_{t-i} - \overline{\text{logret}}\right)^2}$$

$$\text{ATR}_{14}^{(simple)}(t) = \frac{1}{14}\sum_{i=0}^{13}\left(H_{t-i} - L_{t-i}\right)$$

Chú thích: H_t: Giá cao nhất ngày t. L_t: Giá thấp nhất ngày t.

Hình 13 Công thức nhóm chỉ số biến động

```
# --- Volatility ---
g["vol20"] = g["logret"].rolling(20).std()
g["atr14"] = (g["high"] - g["low"]).rolling(14).mean()  # ATR đơn giản

# --- Volume Z-score (20 ngày) ---
if "volume" in g.columns:
    mean20 = g["volume"].rolling(20).mean()
    std20  = g["volume"].rolling(20).std()
    g["vol_z20"] = (g["volume"] - mean20) / std20
else:
    g["vol_z20"] = np.nan
```

Hình 14 Code tính nhóm chỉ số biến động

– Nhóm chỉ số đánh giá cảm xúc : đo tác động giữa kết hợp tâm lý và thanh khoản

sent_polarity: Chênh lệch giữa xác suất tích cực và tiêu cực.

sent_entropy: Độ bất định của phân phối sentiment.

sent_polarity_ema3: Trung bình động EMA(3) của polarity.

sent_vol_interact: Tương tác sentiment × volume.

Công thức

$$\text{sent\_polarity}(t) = p_{\text{pos}} - p_{\text{neg}}$$

$$\text{sent\_entropy}(t) = -\sum_{c \in \{neg, neu, pos\}} p_c \ln(p_c + 10^{-6})$$

$$\text{sent\_vol\_interact}(t) = \text{sent\_polarity}(t) \times \text{vol\_z20}(t)$$

Chú thích: p_c: Xác suất sentiment thuộc loại c.

Hình 15 Công thức tính các chỉ số đánh giá cảm xúc

```
# --- Sentiment features ---
g["sent_polarity"]     = g["p_pos"] - g["p_neg"]
g["sent_entropy"]      = -(g[["p_neg","p_neu","p_pos"]]
                           * np.log(g[["p_neg","p_neu","p_pos"]] + 1e-6)).sum(axis=1)
g["sent_polarity_ema3"]= g["sent_polarity"].ewm(span=3, adjust=False).mean()
g["sent_vol_interact"] = g["sent_polarity"] * g["vol_z20"]
```

Hình 16 Code tính nhóm chỉ số đánh giá cảm xúc

– Chỉ số đánh giá giao động (MACD): Là chỉ báo xu hướng dựa trên sự chênh lệch giữa hai đường EMA (ngắn hạn và dài hạn).

  Mục đích: Xác định tín hiệu mua/bán, phát hiện đảo chiều xu hướng.

  Công thức

$$\text{MACD}(t) = \text{EMA}_{12}(t) - \text{EMA}_{26}(t), \quad \text{Signal}(t) = \text{EMA}_9(\text{MACD}(t))$$

Hình 17 Công thức tính chỉ số MACD

```
# 5) MACD (12, 26, 9)
ema12 = g["close"].ewm(span=12, adjust=False).mean()
ema26 = g["close"].ewm(span=26, adjust=False).mean()
g["macd"]        = ema12 - ema26
g["macd_signal"] = g["macd"].ewm(span=9, adjust=False).mean()
```

Hình 18 Code tính chỉ số MACD

– ATR (Average True Range – Biên độ dao động trung bình)

Khái niệm: Đo lường mức biến động của giá cổ phiếu trong một giai đoạn nhất định.

Mục đích: Xác định độ rủi ro và độ mạnh của biến động giá.

Công thức

$$TR_t = \max\left(H_t - L_t,\ |H_t - C_{t-1}|,\ |L_t - C_{t-1}|\right)$$

$$ATR_t = \frac{1}{n}\sum_{i=0}^{n-1} TR_{t-i}$$

Hình 19 Công thức tính chỉ số ATR(14)

```
# ATR(14) theo True Range
high_low   = g["high"] - g["low"]
high_close = (g["high"] - g["close"].shift()).abs()
low_close  = (g["low"]  - g["close"].shift()).abs()
tr = pd.concat([high_low, high_close, low_close], axis=1).max(axis=1)
g["atr_14"] = tr.rolling(window=14).mean()
```

Hình 20 Code tính chỉ số ATR(14)

- Kiểm tra schema và sắp xếp theo thời gian

  required_cols = set(feature_cols + ["ret", TARGET_COL, "symbol", "time"])

  df["time"] = pd.to_datetime(df["time"])

  df = df.sort_values(["symbol","time"]).reset_index(drop=True)

- Xử lý giá trị thiếu

  df1 = df[df["symbol"] == symbol].copy()

  df1["time"] = pd.to_datetime(df1["time"])

  df1 = df1.sort_values("time").reset_index(drop=True)

  df1 = df1.bfill().ffill()

  df1 = df1.iloc[:-1].copy() # vì ret = shift(-1)

- Xử lý outlier và ép kiểu dữ liệu (tránh Nan, Inf)

  trong hàm clean_and_clip_train_test

  Ép numeric và thay inf→NaN.

  Lấy quantile clip_q = (0.001, 0.999) lấy ngưỡng từ train, sau đó áp dụng cho train test.

  Giới hạn tuyệt đối max_abs = 1e6

  Với test: ffill().bfill trên feature, rồi drop hàng còn NaN.

  Loại các hàng còn giá trị không hữu hạn.

- Chuẩn hóa đặc trưng

x_scaler = RobustScaler()

X_train_all = x_scaler.fit_transform(df_train[feature_cols].astype("float32"))

X_test_all = x_scaler.transform(df_test[feature_cols].astype("float32"))

- Tạo mẫu chuỗi (hàm make_multi_step)

Input: ma trận đặc trưng đã scale và đích là ret

Tạo (X, Y) với kích thước (n_sapmples, W, F) và (n_samples, H, 1)

X_train, Y_train = make_multi_step(X_train_all, y_train_all, W, H)

Chương 3 Phương pháp và mô hình học sâu

3.1 Kiến trúc model VAE Seq2Seq

- Model VAE (Variational Autoencoder)

Tổng quan về model VAE: đây một mô hình sinh (generative model)**,** thuộc họ Autoencoder nhưng mang tính xác suất. Thay vì học một latent vector duy nhất, VAE học một phân phối xác suất trong không gian tiềm ẩn.

Nguyên tắc hoạt động: Autoencoder thường học biểu diễn nén của dữ liệu (latent vector) và tái tạo lại dữ liệu gốc. Model VAE mở rộng ý tưởng bằng cách coi latent vector không phải là một điểm cố định, mà là một phân phối xác xuát (thường giả định là Gaussian)

Cấu trúc model VAE (Gồm 3 phần chính)

Phần 1: Encoder (Mã hóa)

Ý tưởng: trong phần này, model sẽ nhận đầu vào là một biến x. Sau đó xuất ra 2 tham số của phân phối ẩn là trung bình và phương sai.

Phần 2: Reparameterization trick (thủ thuật tái tham số hoá)

Ý tưởng: để có thể lan truyền gradient khi huấn luyện, ta cần tìm được hàm z = trung bình + phương sai * esilon, với esilon thuộc (0, 1).

Phần 3: Decoder

Ý tưởng: nhận đầu vào là z sinh lại dữ liệu x'. Từ đó tìm được xác xuất phân phối.

Hàm mất mát

Encoder: biến đổi dữ liệu đầu vào x thành phân phối tiềm ẩn $q_\phi(z|x)$, thường là Gaussian với tham số $\mu(x), \sigma^2(x)$.

Sampling: dùng *reparameterization trick* để lấy mẫu $z = \mu + \sigma \cdot \epsilon$, trong đó $\epsilon \sim \mathcal{N}(0, I)$.

Decoder: từ z, tái tạo lại x' sao cho giống x.

Hình 21 Hàm mất mát trong model VAE

Reconstruction loss: đảm bảo tái tạo sát dữ liệu gốc.

KL divergence: ép phân phối tiềm ẩn gần với chuẩn N(0,I) → không gian latent trơn tru, dễ sinh mẫu mới.

Đặc điểm

- Có khả năng sinh dữ liệu mới gần giống phân phối gốc.
- Biểu diễn dữ liệu trong latent space có tính chất mượt, liên tục**.**
- Xử lý được uncertainty trong dự báo.

Kiến trúc Seq2Seq

- Tổng quan: Seq2Seq là kiến trúc Encoder–Decoder, dùng phổ biến trong NLP (chatbot) và chuỗi thời gian.
- Encoder đọc chuỗi đầu vào và mã hóa thành vector đặc trưng.
- Decoder dựa trên vector này để sinh chuỗi đầu ra.

Thành phần

- Encoder: LSTM, GRU
- Decoder: mô hình sinh chuỗi là LSTM và GRU.

Ưu điểm

- Học được quan hệ phụ thuộc dài hạn trong dữ liệu tuần tự.
- Linh hoạt: đầu vào và đầu ra có thể có độ dài khác nhau.
- Rất hiệu quả cho dự báo nhiều bước (multi-step forecasting)**.**

Kết hợp model VAE với kiến trúc Seq2Seq

Ý tưởng: Seq2Seq cho phép mô hình hóa dữ liệu tuần tự (thời gian). VAE đưa yếu tố xác suất vào latent space, không chỉ có một vector đặc trưng duy nhất, mà một phân phối tiềm ẩn. Có thể sinh nhiều kịch bản dự báo khác nhau.

Kết hợp lại: mô hình vừa học được cấu trúc tuần tự, vừa phản ánh được bất định**.**

Nguyên tắc hoạt động:

- Encoder (Conv1D + LSTM): rút đặc trưng từ chuỗi quá khứ → tham số μ,σ2.
- Sampling: lấy mẫu từ phân phối tiềm ẩn.
- Decoder (LSTM): sinh chuỗi dự báo trong H bước tương lai.
- Loss: vừa khớp dự báo với thực tế (Huber loss), vừa regularize latent space (KL loss).

Kết quả đạt được

- Probabilistic Forecasting: không chỉ đưa ra một dự báo duy nhất, mà có thể sinh nhiều kịch bản.
- Regularization tự nhiên: KL loss giúp tránh overfitting.
- Tổng quát hóa tốt hơn: latent space được ép gần Gaussian → dự báo ổn định hơn khi dữ liệu nhiễu.
- Ứng dụng tài chính: có thể mô hình hóa phù hợp với dữ liệu có tính bất định cao.

3.2 Ứng dụng mô hình với dữ liệu thực tế

3.2.1 Chia dữ liệu

Tập dữ liệu được chia thành 2 phần là train và test

Đối với tập dữ liệu train df_train = df1.iloc[:-H].copy() sẽ là dữ liệu từ ngày đầu tiên (theo W) cho đến dữ liệu trước ngày test (theo H)

Đôi với tập dữ liệu test df_test = df1.iloc[-H:].copy() sẽ là dữ liệu cuối cùng theo ngày train và kết thúc bằng số ngày (theo H)

Ở đây W là số ngày giao dịch trong quá khứ

H là số ngày trong tương lai cần dự báo

3.2.2 Tham số mô hình

Window size (W): số ngày quan sát trong quá khứ mà mô hình dùng làm đầu vào để dự báo tương lai.

Trong mô hình này sử dụng W = 90. Qua quá trình fine turning ta nhận thấy W = 90 cho thấy kết quả mô hình đạt hiệu quả tốt nhất. Trong thực tế nếu W nhỏ quá → không đủ thông tin, W lớn quá → mô hình nặng, dễ nhiễu. Ngoài ra, còn tùy vào mục tiêu dự đoán của bài toán để xác định được W hợp lý

Horizon (H)

Giá trị thử nghiệm: 20 đạt hiệu quả tốt nhất

Ý nghĩa: số ngày trong tương lai mà mô hình cần dự báo.

Mục tiêu của mô hình là đầu tư trung, dài hạn nên ta cần nắm bắt được xu hướng trong phạm vi từ 20 – 30 phiên tới.

Epochs : số lần train mô hình

Hidden size (HIDDEN): số chiều trong mạng LSTM giúp quyết định khả năng học của mô hình. Việc sử dụng 128 chiều giúp mô hình có đủ số chiều để học được các đặc trưng dữ liệu nhưng lại không quá phức tạp sẽ khiến mô hình train lâu.

Latent dimension (LATENT): số chiều của vector ẩn z trong model VAE

Dropout rate (DROPOUT): tỉ lệ loại bỏ ngẫu nhiên nơ-ron trong quá trình train (0.2%) giúp regularization, giảm overfitting.

Learning rate (0.001): tốc độ cập nhật trọng số. Đây là mức chuẩn với Adam optimizer giúp cân bằng giữa hội tụ nhanh và ổn định

Batch size (BATCH): trong mô hình sử dụng giá trị 128. Đây là số mẫu xử lý trong mỗi lần cập nhật trọng số.

Validation split (VAL_SPLIT): tỉ lệ dữ liệu train dùng cho validation

Beta (BETA_MAX): hệ số nhân với KL Divergence loss.

Loss funtion

Huber loss (δ=0.01): ổn định hơn MSE khi dữ liệu có ngoại lai.

KL Divergence: ép phân phối tiềm ẩn gần Gaussian chuẩn, giúp latent space mượt và dễ sinh kịch bản mới.

3.2.3 Huấn luyện mô hình

Mô hình huấn luyện sẽ gồm 3 phần: Encoder, Latent, Decoder

Encoder: Nhận đầu vào là một chuỗi thời gian, sau đó nén lại thành một vector

Lớp 1 Conv1D (Convolution 1D): hoạt động như một bộ lọc trượt trên chuỗi thời gian. Với kernel size = 5, tại mỗi thời điểm, mô hình nhìn 5 ngày liên tiếp để phát hiện "pattern cục bộ" (ví dụ: chuỗi tăng giá ngắn hạn, đảo chiều, biến động mạnh). Với nhiều filters (64 filters), mô hình học được nhiều loại pattern song song.

Trong lớp này sẽ thực hiện 2 lần

Conv1D lần 1 sẽ nhận đầu vào là số mẫu (128), W và số đặc trưng. Đầu ra sẽ là vector 64 chiều thay cho các đặc trưng ban đầu theo từng ngày

Conv1D lần 2: Đầu vào sẽ là (batch_size, W, 64) từ lớp 1, tiếp tục học các đặc trưng chi tiết hơn và đầu ra cũng là (batch_size, W, 64).

Lớp 2 LayerNormalization: Chuẩn hóa giá trị đầu ra của Conv1D theo từng mẫu. Giúp dữ liệu sau khi qua convolution không bị lệch quá nhiều về scale hoặc phân phối. Giúp giảm các hiện tượng như lan truyền ngược khi gradient được nhân dần qua nhiều lớp, nếu giá trị gradient nhỏ hơn 1 sẽ giảm dần về 0 dẫn đến các lớp sẽ không học được gì. Bên cạnh đó còn giảm được hiện tượng Exploding gradients gọi là gradient bùng nổ, ngược lại với lan truyền ngược thì khi gradient lớn hơn 1 và bị nhân qua nhiều bước thì gradient sẽ tăng lên rất lớn và hậu quả sẽ là mô hình không hội tụ, mất tính ổn định.

Lớp 3 LSTM (Long Short-Term Memory): LSTM là một loại RNN cải tiến, có cổng quên (forget gate)**,** cổng nhớ (input gate)**,** cổng xuất (output gate)**.** Nó giữ lại thông tin

quan trọng từ xa (dài hạn), đồng thời loại bỏ nhiễu ngắn hạn. Trong mô hình nhận đầu vào từ Conv1D LayerNorm. Sau đó tích hợp cả thông tin ngắn hạn (từ Conv1D) và dài hạn (nhờ bộ nhớ LSTM). Kết quả của đầu ra cuối cùng (last hidden state) chính là "summary vector" biểu diễn toàn bộ chuỗi W ngày.

Trong lớp này sẽ lấy dữ liệu đầu ra của lớp 3 (batch_size, W, 64), với mỗi bước nó sẽ cập nhập hidden state dựa vào input hiện tại và trạng thái trước đó Cuối cùng sẽ lấy hiddent state ở bước cuối. Đầu ra của lớp này sẽ là (batch_size, hidden) với hidden = 128

Latent

Trong phần này sẽ xử lý với 4 lớp

- Lớp 1 Dense (mean vector): Dữ liệu trong lớp 4 của Encoder (batch_size, hidden). Lớp này sẽ biến vector ẩn này thành một vector có kích thước bằng số chiều của latent (12 chiều). Đầu ra sẽ là ((batch_size, 12) đây sẽ đại diện cho tham số trung bình của phân phối tiềm ẩn
- Lớp 2 Dense (log-variance vector): đầu vào của lớp này là vector ẩn trong lớp LSTM ở phần Encoder. Với lớp Dense này cũng tạo ra một vector (batch_size, 12) nhưng đại diện cho tham số độ lệch chuẩn

** Cả 2 lớp Dense đều dùng cùng 1 công thức nhưng khác trọng số.

mu = layers.Dense(latent, name="z_mu")(x) (Đối với lớp 1)

logvar = layers.Dense(latent, name="z_logvar")(x) (Đối với lớp 2)

Lớp 3 Sampling

```
class Sampling(layers.Layer):
    def call(self, inputs):
        mu, logvar = inputs
        eps = tf.random.normal(shape=K.shape(mu))
        return mu + K.exp(0.5 * logvar) * eps
```

Hình 22 Lớp Sampling

Công thức

$$z = \mu + \sigma \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Hình 23 Công thức tính Z trong lớp Sampling

Sau lớp 1, 2 ta có được 2 tham số vector (μ, logσ²) có shape (batch_size, 12). Trong lớp này sẽ thực hiện tính σ=exp(0.5·logσ²). Sinh nhiễu ε từ phân phối chuẩn N(0,I). Sau đó lấy mẫu z bằng công thức trên. Kết quả của lớp này ta sẽ thu được một vector z có dạng (batch_size, 12).

- Lớp 4 KLDivergenceLayer (β = 0.5): đầu vào trong lớp này vẫn tiếp tục là 2 tham số vector (μ, logσ²) nhưng thực công thức tính khác.

Công thức

Với μ và $\log \sigma^2$ (vector chiều `latent=12`):

$$\mathcal{L}_{KL} = \frac{1}{2} \sum_{j=1}^{\text{latent}} (\mu_j^2 + \sigma_j^2 - 1 - \log \sigma_j^2) = -\frac{1}{2} \sum_{j} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2)$$

Hình 24 Công thức cho lớp KL DivergenceLayer

Trong lớp này sẽ thực hiện tính KL Divergence giữa phân phối tiềm ẩn $q(z|x)=N(\mu,\sigma 2I)$ và phân phối chuẩn N(0,I). Dữ liệy trả về (μ, $\log\sigma^2$) kèm thêm 1 tham số term KL vào hàm loss. Mục tiêu sẽ giúp latent phân phối chuẩn hơn.

Decoder

Trong lớp này sẽ thực hiện 4 tiến trình như sau:

- RepeatVector (horizon = H): nhận đầu vào là một vector latent $z \in R(batch\_size,latent\_dim)$, với latent_dim = 12.
- Hoạt động: nhân bản vector latent z thành một chuỗi có H bước (ứng với H ngày dự báo).
- Đầu ra: R(batch_size,H,latent_dim).
- Mục đích: đưa thông tin tiềm ẩn vào mỗi bước dự báo trong chuỗi thời gian tương lai.

LSTM(hidden = 128, return_sequences = True, dropout = 0.2)

- Đầu vào: chuỗi R(batch_size,H,latent_dim).
- Hoạt động: xử lý tuần tự, sinh ra trạng thái ẩn cho từng bước thời gian.
- Đầu ra: R(batch_size,H,128).
- Mục đích: mô hình hóa quan hệ phụ thuộc theo thời gian giữa các bước dự báo (giữa ngày 1, ngày 2, … đến ngày H).

TimeDistributed(Dense(64, activation = "relu"))

- Đầu vào: chuỗi R(batch_size,H,128) từ LSTM.
- Hoạt động: mỗi bước dự báo đi qua một lớp Dense 64 chiều (cùng tham số cho mọi timestep).
- Đầu ra: R(batch_size,H,64).
- Mục đích: học phi tuyến cục bộ để biến đổi đặc trưng từ không gian 128 chiều sang 64 chiều, giúp tăng tính biểu diễn.

TimeDistributed(Dense(1))

- Đầu vào: chuỗi R(batch_size,H,64).

- Hoạt động: ánh xạ đặc trưng ở mỗi timestep thành một giá trị dự báo (linear).
- Đầu ra: R(batch_size,H,1).
- Mục đích: tạo ra dự báo cuối cùng cho lợi suất (ret) từng ngày trong H ngày tương lai.

Loss Function

Công thức

$$\mathcal{L} = \mathcal{L}_{\text{Huber}} + \beta \cdot \mathcal{L}_{\text{KL}}$$

Hình 25 Công thức hàm Loss Function

- Huber Loss (δ = 0.01)Mục đích của tham số này là đo sai số dự báo (reconstruction/prediction) giữa chuỗi lợi suất dự báo và chuỗi lợi suất thực. Kết hợp ưu điểm của MSE (nhạy với sai số nhỏ, mượt gradient) và MAE (robust với ngoại lai). Với δ nhỏ (0.01), loss "giống MSE" khi sai số rất nhỏ, và "giống MAE" khi sai số vượt ngưỡng δ.
- Công thức

Gọi $e = y - \hat{y}$ là sai số phần tử (từng ngày trong H), Huber(δ) theo từng phần tử:

$$\ell_\delta(e) = \begin{cases} \frac{1}{2}e^2 & \text{nếu } |e| \le \delta, \\ \delta\left(|e| - \frac{1}{2}\delta\right) & \text{nếu } |e| > \delta \end{cases}$$

Hình 26 Công thức tính hàm Huber Loss

Đầu vào dữ liệu sẽ là y (ground truth) shape (batch,H,1) là chuỗi ret thực. y^ (prediction) shape (batch,H,1) là chuỗi ret dự báo.

```
vae.compile(
    optimizer=optimizers.Adam(learning_rate=lr),
    loss=losses.Huber(delta=0.01),
```

metrics=[losses.MeanSquaredError(name="mse")]

)

- KL Divergence Loss (β = 0.5): tham số này cho biết Regularization với xác suất cho latent space: ép phân phối hậu nghiệm xấp xỉ qϕ (z|x)=N(μ,σ2I) gần với prior p(z)=N(0,I). Giúp latent space mượt, có cấu trúc, dễ tổng quát hóa, hạn chế overfitting (không "nhớ" huấn luyện một cách cứng nhắc).
- Công thức

Gọi $e = y - \hat{y}$ là sai số phần tử (từng ngày trong H), Huber(δ) theo từng phần tử:

$$\ell_\delta(e) = \begin{cases} \frac{1}{2}e^2 & \text{nếu } |e| \le \delta, \\ \delta\left(|e| - \frac{1}{2}\delta\right) & \text{nếu } |e| > \delta \end{cases}$$

Hình 27 Công thức tính hàm KL Divergence Loss

Khi huấn luyện, loss này được lấy trung bình theo batch. Trong code, nó được thêm vào tổng loss thông qua add_loss. Đầu vào sẽ là tham số μ và logσ2. Đầu ra là một scalar LKL (mean theo batch), sau đó nhân với β=0.5 rồi cộng vào tổng loss

```
class KLDivergenceLayer(layers.Layer):
    def __init__(self, beta=0.5, **kwargs):
        super().__init__(**kwargs)
        self.beta = beta

    def call(self, inputs):
        mu, logvar = inputs
        kl_per = -0.5 * K.sum(1 + logvar - K.square(mu) - K.exp(logvar), axis=1)
        kl = K.mean(kl_per)
        self.add_loss(self.beta * kl)
        return inputs
```

Hình 28 Lớp KL DivergenceLayer

Lớp KL Divergence Layer

Nguyên tắc hoạt động

- Lhuber : tập trung vào độ chính xác dự báo: giảm sai số giữa chuỗi dự báo và thực. Ổn định với ngoại lai nhờ ngưỡng δ=0.01.
- β·LKL : định hình không gian tiềm ẩn. Khi β tăng, mô hình ưu tiên "gọn" latent (gần N(0,I)) → tổng quát hóa tốt hơn nhưng có thể giảm độ khớp ngắn hạn. Khi β giảm về 0, mô hình gần giống autoencoder thường (dễ overfit, ít tính sinh). Ở đây β=0.5 là thế cân bằng giữa độ chính xác và khả năng tổng quát/tính xác suất**.**

Chương 4 Kết quả và đánh giá mô hình

4.1 Chỉ số đánh giá mô hình

- Hàm đánh giá RMSE (Root Mean Square Error): đo mức độ sai số tuyệt đối trung bình theo dữ liệu giá. Nhấn mạnh các sai số lớn vì có bình phương. Thích hợp khi cần quan tâm đến các dự báo "tệ nhất" có ảnh hưởng mạnh.
- Công thức

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Hình 29 Chỉ số đánh giá RMSE

- Hàm đánh giá MAPE (Mean Absolute Percentage Error): biểu diễn sai số dưới dạng tỷ lệ phần trăm, giúp dễ hiểu với nhà đầu tư
- Công thức

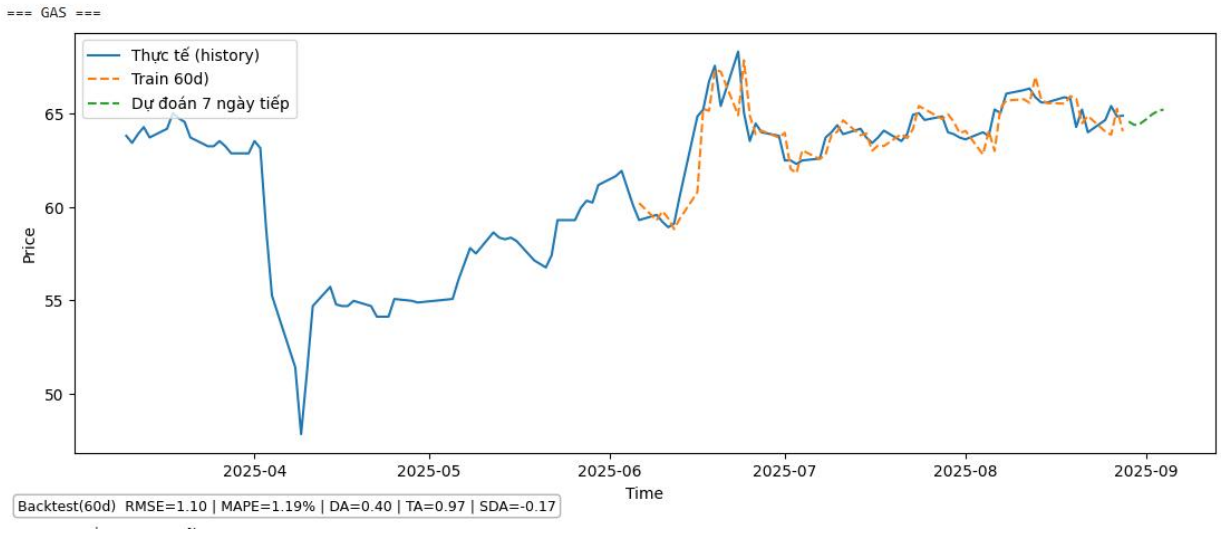
$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Hình 30 Chỉ số đánh giá MAPE

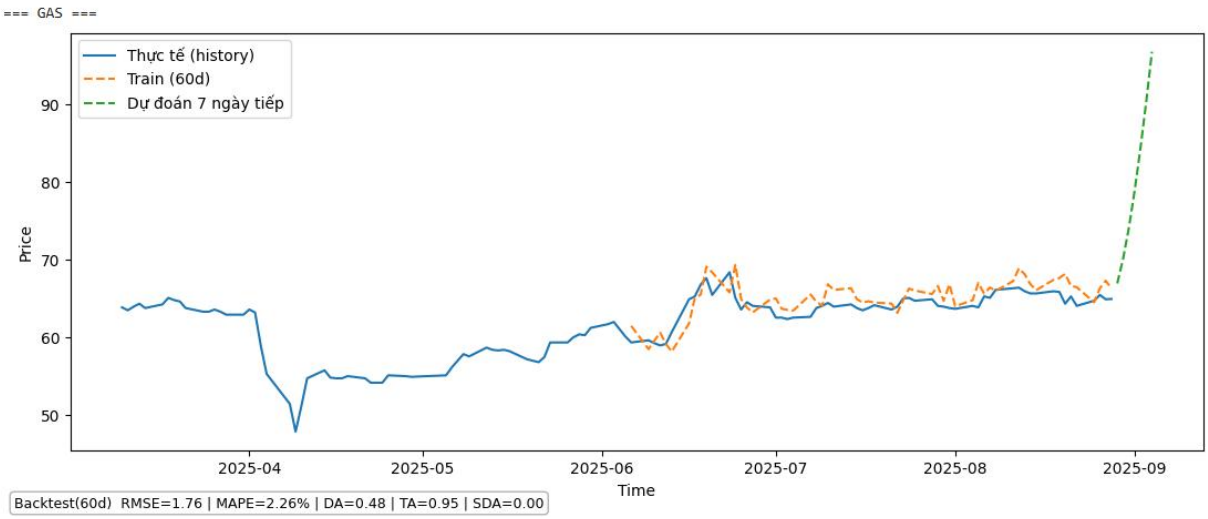
4.2 Kết quả và đánh giá

4.2.1 Đánh giá mô hình VAE Seq2Seq

Tổng quan: dữ liệu đang sử dụng đối với mã cổ phiếu GAS, các tham số truyền vào trong mô hình là như sau (khác chỉ số đánh giá cảm xúc). Trong mô hình, dữ liệu cho phần đánh giá gồm 60 ngày trước (với phần train) và 7 ngày sau (cho phần dự đoán).. Đối với model sử dụng chỉ số cảm xúc đang cho kết quả tốt hơn với RMSE giảm ~0.66 nghìn VND và MAPE giảm ~1.07% so với mô hình bình thường.

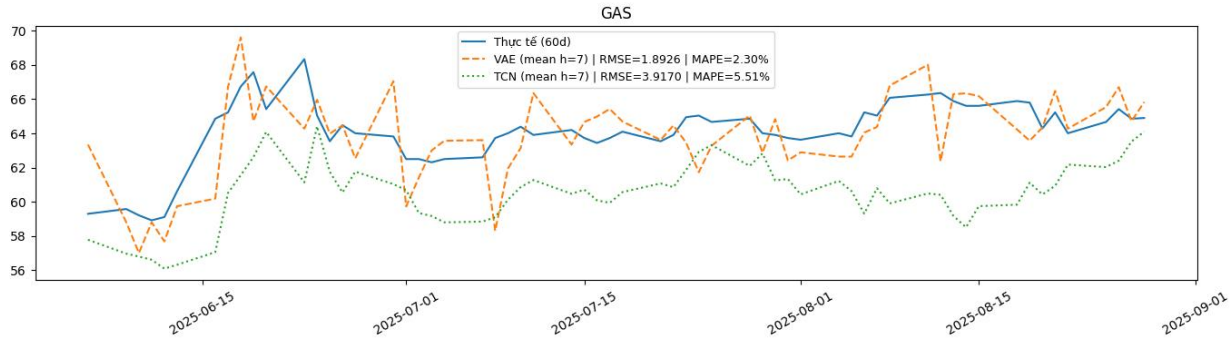


Hình 31 Mô hình có kết hợp với chỉ số cảm xúc



Hình 32 Mô hình không sử dụng chỉ số cảm xúc

4.2.2 Mô hình VAE Seq2Seq và TCN

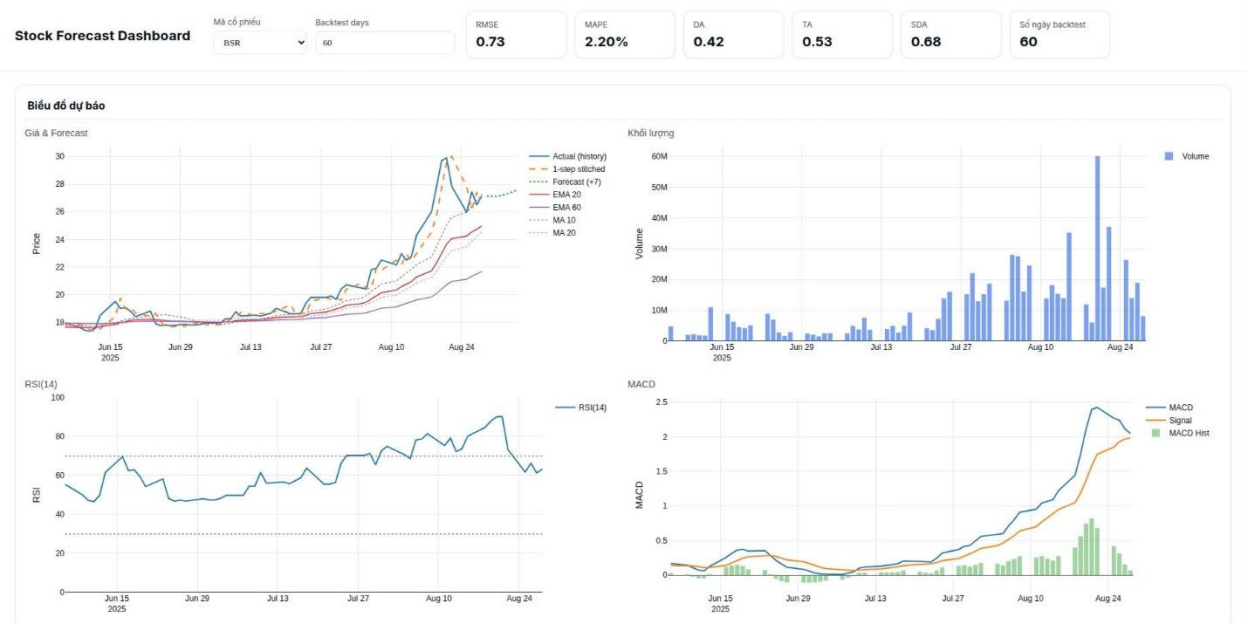


Hình 33 Biểu đồ so sánh mô hình VAE Seq2Seq và TCN

Khi so sánh hai mô hình dự báo chuỗi thời gian là VAE Seq2Seq và TCN, có thể thấy sự khác biệt rõ rệt cả về chất lượng dự báo và đặc trưng mô hình học. Kết quả cho thấy, VAE Seq2Seq đạt hiệu suất vượt trội hơn đáng kể so với TCN: sai số RMSE và MAPE đều thấp hơn, đồng thời đường dự báo bám sát biến động giá thực tế hơn. Có thể khẳng định rằng VAE Seq2Seq là lựa chọn phù hợp hơn cho các bài toán dự báo tài chính đa bước trong ngắn hạn, nơi yêu cầu mô hình vừa phải theo sát biến động vừa đảm bảo sai số dự báo thấp.

4.3 Phát triển giao diện web trực quan

Giao diện web trong hình ảnh thể hiện một bảng điều khiển dự báo cổ phiếu, trong đó tích hợp đồng thời các chỉ số định lượng và biểu đồ kỹ thuật phục vụ phân tích đầu tư. Ở phần đầu, website cung cấp các chỉ số đánh giá mô hình dự báo như RMSE, MAPE, DA, TA, SDA cùng số ngày backtest (số ngày train), giúp nhà đầu tư đánh giá mức độ chính xác và ổn định của mô hình dự báo. Biểu đồ trung tâm thể hiện so sánh giữa giá thực tế trong phạm vi 7 ngày, đồng thời kết hợp với các đường trung bình động (EMA20, EMA60, MA10, MA20) nhằm phản ánh xu hướng giá trong ngắn hạn và dài hạn. Bên phải là biểu đồ khối lượng giao dịch, cung cấp thông tin về mức độ thanh khoản và sự quan tâm của thị trường trong từng giai đoạn. Hai biểu đồ kỹ thuật ở dưới gồm RSI(14) – chỉ báo đo lường mức độ quá mua/quá bán, và MACD – chỉ báo dao động xu hướng, hỗ trợ xác định thời điểm mua/bán hợp lý.



Hình 34 Giao diện Website trực quan hóa biểu đồ

Chương 5 Kết luận và hướng phát triển

5.1 Kết luận

Với mục tiêu của bài toán là dự đoán giá cổ phiếu trong phạm vi ngắn hạn. Thông qua đề tài này đã chứng minh được việc kết hợp dữ liệu tin tức với dữ liệu lịch sử giao dịch trong model VAE Seq2Seq cho kết quả dự đoán giá sát với biến động thực tế hơn. Trong đề tài đã sử dụng model chính để so sánh với model TCN trên cùng tham số đầu vào, kết quả cho thấy model VAE Seq2Seq cho kết quả tốt hơn đáng kể khi chỉ số đánh giá RMSE và MAPE đều nhỏ hơn. Hạn chế của đề tài chỉ mới ứng dụng 2 nguồn dữ liệu là giao dịch và tin tức, chưa ứng dụng được các nguồn dữ liệu từ báo cáo tài chính, những chính sách mà chính phủ cập nhật riêng cho thị trường chứng khoán. Đồng thời dữ liệu trong bài toán chỉ dừng 2 nhóm ngành có ảnh hưởng lớn trong giai đoạn AI phát triển ở hiện tại là Công nghệ và Năng lượng.

5.2 Hướng phát triển

Trong quá trình nghiên cứu dự đoán xu hướng giá cổ phiếu, việc chỉ dựa trên dữ liệu giá và một số chỉ số tài chính cơ bản đôi khi chưa phản ánh đầy đủ sự biến động phức tạp của

thị trường. Thị trường chứng khoán chịu ảnh hưởng từ nhiều yếu tố khác nhau như tình hình kinh tế vĩ mô, kết quả hoạt động của doanh nghiệp, cũng như các sự kiện chính trị – xã hội. Do đó, hướng phát triển của nghiên cứu sẽ tập trung vào việc mở rộng nguồn dữ liệu và kết hợp nhiều phương pháp phân tích để xây dựng mô hình dự báo toàn diện và chính xác hơn, áp dụng cho các khung thời gian đầu tư ngắn hạn, trung hạn và dài hạn.

5.2.1 Mở rộng phạm vi dữ liệu

- Dữ liệu định lượng: báo cáo tài chính của doanh nghiệp (doanh thu, lợi nhuận, EPS, ROE, P/E…), các chỉ số kinh tế vĩ mô (lãi suất, tỷ giá, lạm phát, GDP, giá dầu, dòng vốn FDI).
- Dữ liệu định tính**:** tin tức kinh tế – chính trị, sự kiện quốc tế, chính sách vĩ mô, các báo cáo phân tích ngành, dữ liệu từ mạng xã hội và báo chí tài chính.

5.2.2 Phương pháp phân tích

- Mô hình thống kê và kinh tế lượng**:** sử dụng các mô hình ARIMA, GARCH, VAR để dự báo chuỗi thời gian và phân tích tác động của các yếu tố vĩ mô.
- Xử lý ngôn ngữ tự nhiên (NLP)**:** phân tích cảm xúc (Sentiment Analysis) từ tin tức, mạng xã hội để định lượng tâm lý thị trường.
- Mô hình Hybrid**:** kết hợp dữ liệu định lượng và định tính, ứng dụng học máy (Machine Learning) và học sâu (Deep Learning như LSTM, GRU, Transformer) nhằm tăng cường độ chính xác và khả năng khái quát của mô hình.

5.2.3 Phạm vi dự báo

- Ngắn hạn (1–3 tháng): tập trung vào giao dịch, tin tức nhanh, tâm lý thị trường.
- Trung hạn (6–12 tháng): dựa vào báo cáo tài chính định kỳ, xu hướng ngành và tác động của chính sách kinh tế.
- Dài hạn (trên 1 năm): phân tích triển vọng phát triển ngành, yếu tố vĩ mô bền vững, định giá cổ phiếu theo mô hình tài chính (DCF, Gordon Growth Model).

TÀI LIỆU THAM KHẢO

[1] P. Chen, Z. Boukouvalas, and R. Corizzo, “A deep fusion model for stock market prediction with news headlines and time series data,” Neural Computing and Applications, vol. 36, pp. 21229–21271, 2024, doi: 10.1007/s00521-024-10303-1.

[2] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, “Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 14, no. 1, e1519, 2024, doi: 10.1002/widm.1519.

[3] K. J. L. Koa, Y. Ma, R. Ng, and T.-S. Chua, “Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction,” arXiv preprint arXiv:2309.00073, 2023, doi: 10.48550/arXiv.2309.00073.

[4] K. Du, F. Z. Xing, R. Mao, and E. Cambria, “Financial Sentiment Analysis: Techniques and Applications,” ACM Computing Surveys, vol. 56, no. 9, 2024, doi: 10.1145/3649451.

[5] W.-J. Gu, Y.-H. Zhong, S.-Z. Li, C.-S. Wei, L.-T. Dong, Z.-Y. Wang, and C. Yan, “Predicting Stock Prices with FinBERT-LSTM: Integrating News Sentiment Analysis,” in Proc. 2024 8th Int. Conf. on Cloud and Big Data Computing (ICCBDC ’24), Oxford, U.K., 2024, pp. 67–72, doi: 10.1145/3694860.3694870.

[6] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, “Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review,” Natural Language Processing Journal, vol. 6, 100059, 2024, doi: 10.1016/j.nlp.2024.100059.

[7] A. Todd, J. Bowden, and Y. Moshfeghi, “Text-based sentiment analysis in finance: Synthesising the existing literature and exploring future directions,” Intelligent Systems in Accounting, Finance and Management, vol. 31, no. 1, e1549, 2024, doi: 10.1002/isaf.1549.