

# Ứng dụng Spark và HDFS phân tích và dự đoán chứng khoán Việt Nam

23020342 – Bùi Thanh Dân

Báo cáo Assignment BigData

## Tóm tắt nội dung

Báo cáo này trình bày hệ thống **phân tích và dự đoán giá chứng khoán Việt Nam** sử dụng 500 tệp CSV được **crawl tự động từ thư viện vnstock**. Hệ thống tận dụng hai công nghệ trọng yếu của hệ sinh thái Big Data là **Apache Spark** và **Hadoop HDFS** để xử lý dữ liệu quy mô lớn, đồng thời kết hợp mô hình **LSTM (Long Short-Term Memory)** triển khai bằng **PyTorch** để dự đoán xu hướng giá cổ phiếu. Kết quả chứng minh khả năng xử lý song song hiệu quả, tốc độ huấn luyện nhanh và độ chính xác cao trong dự báo ngắn hạn.

## 1 Giới thiệu

Thị trường chứng khoán Việt Nam sản sinh hàng triệu bản ghi giao dịch mỗi ngày. Việc phân tích xu hướng giá yêu cầu năng lực xử lý dữ liệu lớn, khả năng mở rộng linh hoạt và độ chính xác cao. Do đó, hệ thống được thiết kế dựa trên:

- **Hadoop HDFS**: lưu trữ 500 tệp CSV dữ liệu cổ phiếu dạng phân tán, đảm bảo an toàn và chịu lỗi cao.
- **Apache Spark**: thực hiện xử lý dữ liệu, tổng hợp thống kê và chuẩn bị đầu vào cho mô hình học sâu.
- **PyTorch LSTM**: mô hình học sâu nhiều tầng dùng cho dự đoán chuỗi thời gian (Time Series Forecasting).

Các mô-đun chính:

- `stock_price_demo.ipynb`: đọc, trực quan hóa và phân tích biến động của một số cổ phiếu.
- `prediction.ipynb`: huấn luyện mô hình LSTM và dự đoán xu hướng giá tương lai.

## 2 Kiến trúc hệ thống

Hệ thống được triển khai bằng **Docker Compose** mô phỏng cụm Big Data hoàn chỉnh gồm:

- 1 **NameNode** và 4 **DataNode** (lưu trữ 500 file CSV trong HDFS tại /dataack)
- 1 **Spark Master** và 4 **Spark Worker** (mỗi worker có 16 core, 12.6 GiB RAM)
- 1 **Jupyter Container** cho việc lập trình, huấn luyện và kiểm thử mô hình

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	54.76 KB	Oct 25 10:37	3	128 MB	A32.csv
-rw-r--r--	root	supergroup	59.17 KB	Oct 25 10:37	3	128 MB	AAA.csv
-rw-r--r--	root	supergroup	16.88 KB	Oct 25 10:37	3	128 MB	AAH.csv
-rw-r--r--	root	supergroup	54.48 KB	Oct 25 10:37	3	128 MB	AAM.csv
-rw-r--r--	root	supergroup	49.76 KB	Oct 25 10:37	3	128 MB	AAS.csv
-rw-r--r--	root	supergroup	44.86 KB	Oct 25 10:37	3	128 MB	AAT.csv
-rw-r--r--	root	supergroup	53.6 KB	Oct 25 10:37	3	128 MB	AAV.csv
-rw-r--r--	root	supergroup	46.21 KB	Oct 25 10:37	3	128 MB	ABB.csv
-rw-r--r--	root	supergroup	55.7 KB	Oct 25 10:37	3	128 MB	ABC.csv
-rw-r--r--	root	supergroup	58.64 KB	Oct 25 10:37	3	128 MB	ABI.csv
-rw-r--r--	root	supergroup	54.6 KB	Oct 25 10:37	3	128 MB	ABR.csv
-rw-r--r--	root	supergroup	55.37 KB	Oct 25 10:37	3	128 MB	ABS.csv

Hình 1: Danh sách các tệp CSV được lưu trong HDFS tại thư mục /dataack.

**Spark Master at spark://cc34b495a768:7077**

URL: spark://cc34b495a768:7077  
 Alive Workers: 4  
 Cores in use: 64 Total, 0 Used  
 Memory in use: 50.2 GiB Total, 0.0 B Used  
 Resources in use:  
 Applications: 0 Running, 2 Completed  
 Drivers: 0 Running, 0 Completed  
 Status: ALIVE

▼ Workers (4)

Worker Id	Address	State	Cores	Memory	Resources
worker-20251025033659-172.18.0.10-34683	172.18.0.10:34683	ALIVE	16 (0 Used)	12.6 GiB (0.0 B Used)	
worker-20251025033659-172.18.0.11-40071	172.18.0.11:40071	ALIVE	16 (0 Used)	12.6 GiB (0.0 B Used)	
worker-20251025033659-172.18.0.8-35179	172.18.0.8:35179	ALIVE	16 (0 Used)	12.6 GiB (0.0 B Used)	
worker-20251025033700-172.18.0.12-39925	172.18.0.12:39925	ALIVE	16 (0 Used)	12.6 GiB (0.0 B Used)	

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

▼ Completed Applications (2)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20251025034724-0001	Stocks-price-Prediction	64	1024.0 MiB		2025/10/25 03:47:24	joyvan	FINISHED	9.5 min
app-20251025034618-0000	Stocks-price-Analysis	64	1024.0 MiB		2025/10/25 03:46:18	joyvan	FINISHED	24 s

Hình 2: Cụm Spark Master và 4 Spark Worker đang hoạt động trong Docker.

### 3 Dữ liệu

Dữ liệu được thu thập tự động qua thư viện **vnstock**, bao gồm 500 mã cổ phiếu niêm yết (ví dụ: ACB, BCM, BID, CTG, DGC, FPT, ...). Các tệp này sẽ được lưu trong trên HDFS trong quá trình xử lý.

Mỗi tệp CSV chứa lịch sử giá cổ phiếu từ năm 2020 đến 2025 có dạng như sau:

Bảng 1: Ví dụ dữ liệu cổ phiếu của FPT.

time	open	high	low	close	volume
23-10-2025	97.2	97.3	94	95	6673501
22-10-2025	95.2	99	95	97	18365820
21-10-2025	89	93	88.5	93	19571735
20-10-2025	88.3	91.8	87	87	14633215

## 4 Xử lý và Phân tích Dữ liệu

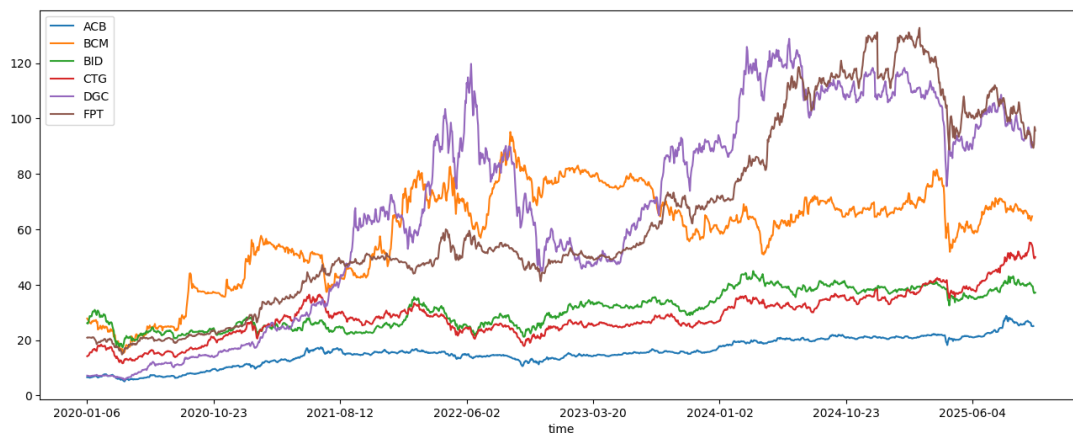
Bước đầu tiên là đọc dữ liệu từ HDFS và tính toán giá trị trung bình bằng công thức  $\text{mean} = (\text{high} + \text{low})/2$  cho từng mã.

```
spark = SparkSession.builder \
    .appName("Stocks-price-Analysis") \
    .master("spark://spark-master:7077") \
    .config("spark.hadoop.fs.defaultFS", "hdfs://namenode:9000") \
    .getOrCreate()

df_FPT = spark.read.csv("hdfs://namenode:9000/dataack/FPT.csv",
                        header=True, inferSchema=True)
df_FPTmean = df_FPT.withColumn("mean", expr('(high+low)/2'))
```

Sau đó Spark chuyển đổi sang Pandas để vẽ biểu đồ trung bình giá nhiều mã cổ phiếu:

```
ax = df_ACBmean.plot(x='time', y='mean', label='ACB', figsize=(16,6))
df_FPTmean.plot(ax=ax, x='time', y='mean', label='FPT')
plt.legend()
```



Hình 3: Trực quan hóa dữ liệu một số loại cổ phiếu.

## 5 Huấn luyện mô hình LSTM

Sau giai đoạn phân tích, dữ liệu được chia làm hai tập: **Train (2020-2023)** và **Test (2024-2025)**. Mô hình LSTM gồm 4 tầng tuần tự, dropout và lớp dense đầu ra:

---

```
class LSTMRegressor(nn.Module):
    def __init__(self, input_size=1, hidden_size=50, dropout_rate=0.2):
        super(LSTMRegressor, self).__init__()

        self.lstm1 = nn.LSTM(input_size, hidden_size, batch_first=True)
        self.dropout1 = nn.Dropout(dropout_rate)

        self.lstm2 = nn.LSTM(hidden_size, hidden_size, batch_first=True)
        self.dropout2 = nn.Dropout(dropout_rate)

        self.lstm3 = nn.LSTM(hidden_size, hidden_size, batch_first=True)
        self.dropout3 = nn.Dropout(dropout_rate)

        self.lstm4 = nn.LSTM(hidden_size, hidden_size, batch_first=True)
        self.dropout4 = nn.Dropout(dropout_rate)

        self.fc = nn.Linear(hidden_size, 1)

    def forward(self, x):
        out, _ = self.lstm1(x)
        out = self.dropout1(out)
        out, _ = self.lstm2(out)
        out = self.dropout2(out)
        out, _ = self.lstm3(out)
        out = self.dropout3(out)
        out, _ = self.lstm4(out)
        out = self.dropout4(out[:, -1, :])
        out = self.fc(out)
        return out
```

---

Sau đó mô hình được huấn luyện trong 100 epoch:

---

```
for epoch in range(epochs):
    for X_batch, y_batch in train_loader:
        outputs = model(X_batch)
        loss = criterion(outputs, y_batch)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

---

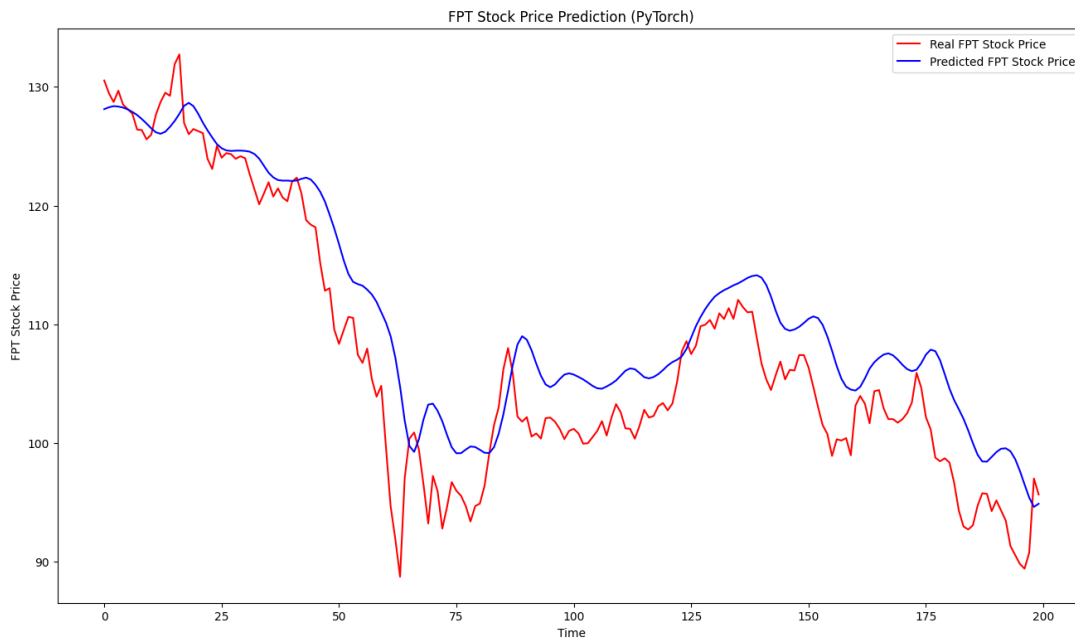
## 6 Dự đoán và Đánh giá

Sau khi huấn luyện, mô hình được dùng để dự đoán giá trung bình cho các ngày năm 2025.

Kết quả minh họa (mã FPT):

Ngày 24/10/2025: 96.01  
Ngày 25/10/2025: 97.52  
Ngày 26/10/2025: 99.05  
Ngày 27/10/2025: 100.44  
...

Biểu đồ so sánh:



Hình 4: Giá thực tế (đỏ) và giá dự đoán (xanh) của mã FPT.

Chỉ số đánh giá (FPT):

- Root Mean Squared Error (RMSE): 4.9366
- Mean Absolute Percentage Error (MAPE): 3.97 %

## 7 Hiệu năng hệ thống

- 500 tệp CSV được đọc trong 10 giây.
- Toàn bộ quá trình huấn luyện trên cụm Spark hoàn tất trong khoảng 10 phút.
- Dự đoán và đánh giá chạy song song trên 4 worker, tốc độ nhanh hơn 3-6 lần so với local.

## 8 Kết luận

Hệ thống đạt được mục tiêu: xây dựng pipeline Big Data hoàn chỉnh cho phân tích và dự đoán chứng khoán Việt Nam.

Kết quả cho thấy:

- Spark và HDFS xử lý dữ liệu crawl tự động hiệu quả.
- Mô hình LSTM đạt sai số thấp, phản ánh tốt xu hướng ngắn hạn.
- Hệ thống có thể mở rộng dễ dàng để huấn luyện thêm nhiều mã cổ phiếu khác.

## 9 Hướng phát triển

- Mở rộng huấn luyện đồng thời nhiều mã cổ phiếu với PySpark + DistributedDataParallel.
- Lưu kết quả dự đoán vào MongoDB và hiển thị bằng Streamlit dashboard.
- Kết hợp dữ liệu thời gian thực qua Kafka + Spark Streaming để dự báo trực tuyến.

## A Phụ lục: Cách chạy hệ thống

### A.1 Mã nguồn

Toàn bộ mã nguồn được cho trong link GitHub sau: <https://github.com/yamddd/Stock-Price>

### A.2 Khởi động cụm Docker với 4 Spark Worker

---

```
docker-compose up -d --scale spark-worker=4
```

---

### A.3 Đưa dữ liệu vào HDFS

---

```
docker cp "\your\path\Stock-Price\notebook\dataack\" namenode:/tmp/dataack
docker exec -it namenode bash
>>> hdfs dfs -mkdir -p /dataack
>>> hdfs dfs -mkdir -p /user/jovyan/output
>>> hdfs dfs -chown -R jovyan:supergroup /user/jovyan
>>> hdfs dfs -put /tmp/dataack/*.csv /dataack/
```

---

### A.4 Truy cập Jupyter

---

```
docker logs pyspark-notebook
```

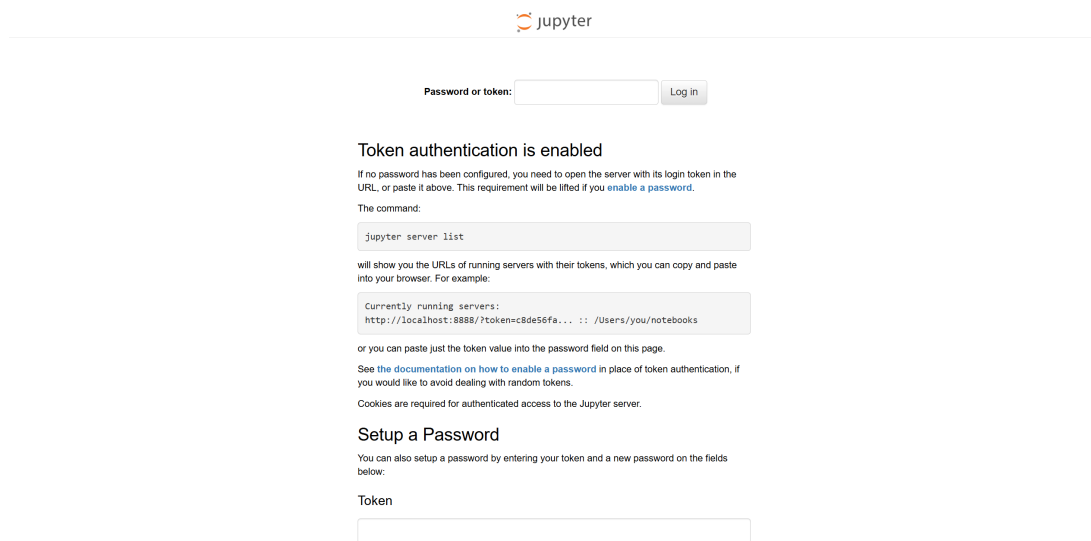
---

Sau đó lấy token Login

Ví dụ: <http://127.0.0.1:8888/lab?token=6ef3b7256830b68266bcec87f90aa335c50a9d0462cc136f>

Token: 6ef3b7256830b68266bcec87f90aa335c50a9d0462cc136f

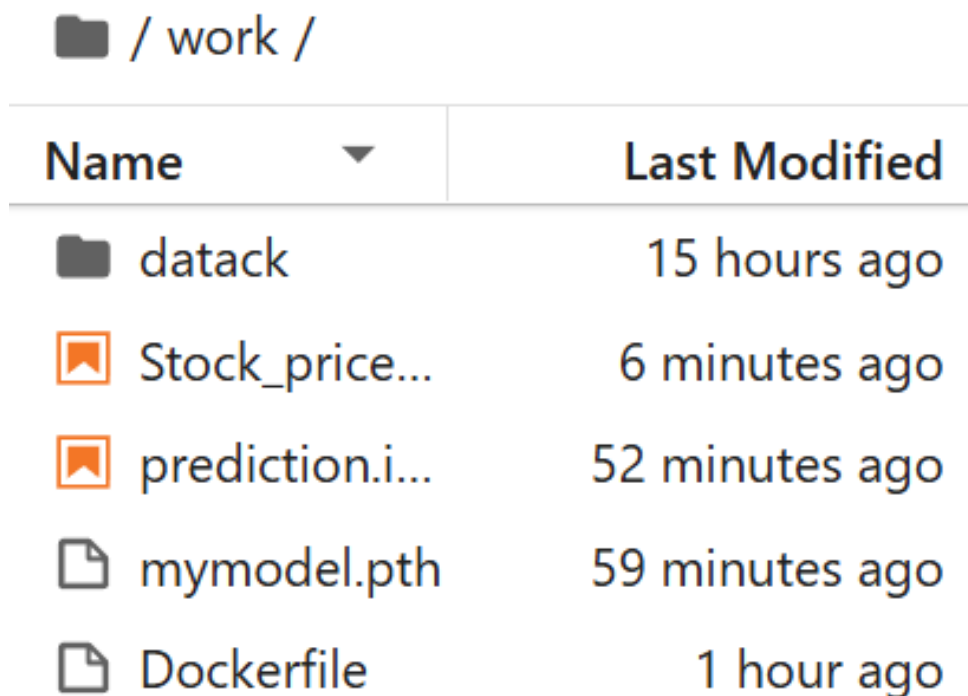
Truy cập: localhost:8888 sẽ xuất hiện giao diện:



The image shows the Jupyter Notebook login page. At the top, it says "jupyter". Below that, there is a "Password or token:" label followed by a text input field and a "Log in" button. A message states "Token authentication is enabled" and explains that if no password is configured, the user needs to open the server with its login token in the URL. It provides a command: `jupyter server list`. Below this, it says "will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:" and shows a box with "Currently running servers:" and a URL: `http://localhost:8888/?token=c8de56fa... : /Users/you/notebooks`. It then says "or you can paste just the token value into the password field on this page." and provides a link to documentation on how to enable a password. At the bottom, there is a "Setup a Password" section with a message: "You can also setup a password by entering your token and a new password on the fields below:" and a "Token" label followed by a text input field.

Hình 5: Giao diện của Jupyter Notebook.

Thư mục work chứa các file cần thiết:



The image shows a file explorer view of the 'work' directory. The path is indicated as '/ work /'. The table below lists the files and folders in the directory.

Name	Last Modified
dataack	15 hours ago
Stock_price...	6 minutes ago
prediction.i...	52 minutes ago
mymodel.pth	59 minutes ago
Dockerfile	1 hour ago

Hình 6: Cấu trúc thư mục trong Jupyter Notebook.

## A.5 Thử nghiệm

Tùy chỉnh và thử nghiệm các file `Stock_price_demo.ipynb` và `prediction.ipynb`, xem các kết quả và dữ liệu trực quan hóa.

## B Tài liệu tham khảo

- Apache Spark Documentation. <https://spark.apache.org/docs/latest/>
- Hadoop HDFS Overview. <https://hadoop.apache.org/docs/stable/>
- PyTorch LSTM Reference. <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>
- vnstock API Documentation. <https://vnstocks.com/>
- GitHub: <https://github.com/thviet79/Stock-Price/tree/master>