Báo cáo Triển khai hệ thống phân tích và xử lý dữ liệu lớn - Stock Price Bigdata

Nguyễn Thị Minh Ly
University of Engineering and Technology, VNU
MSSV: 23020399

Abstract

Báo cáo trình bày quy trình xây dựng, triển khai và phân tích dữ liệu chứng khoán Việt Nam bằng hệ thống Big Data sử dụng Hadoop và Spark trên Docker. Nguồn dữ liệu được thu thập tự động thông qua thư viện `vnstock` với 10 mã cổ phiếu niêm yết lớn. Hệ thống thực hiện lưu trữ dữ liệu trên HDFS, xử lý và thống kê bằng Spark, hiển thị kết quả trên Jupyter Notebook.

1 Giới thiệu

Thị trường chứng khoán Việt Nam có khối lượng giao dịch rất lớn mỗi ngày. Việc áp dụng các công nghệ xử lý dữ liệu lớn như Hadoop và Spark giúp tối ưu hóa việc thu thập, lưu trữ, và phân tích dữ liệu, từ đó hỗ trợ ra quyết định đầu tư hiệu quả hơn.

Dự án này mô phỏng hệ thống Big Data để:

- Thu thập dữ liệu giá cổ phiếu bằng `vnstock`.
- Lưu trữ và quản lý dữ liệu trên Hadoop Distributed File System (HDFS).
- Phân tích, tính toán thống kê bằng Apache Spark.
- Trực quan hóa kết quả phân tích.

2 Thu thập và xử lý dữ liệu

2.1 Nguồn dữ liệu

Dữ liệu được lấy từ thư viện `vnstock`, trích xuất từ nguồn `VCI` (VietCap Securities). Các mã cổ phiếu được chọn gồm: VIC, VHM, VNM, HPG, VCB, BID, CTG, MBB, TCB, FPT.

2.2 Tạo tập dữ liệu

Listing 1: Tải dữ liệu chứng khoán bằng thư viện vnstock

```
from vnstock import Vnstock
import pandas as pd, os

os.makedirs("datack", exist_ok=True)
tickers = ["VIC", "VHM", "VNM", "HPG", "VCB", "BID", "CTG", "MBB", "
    ↪ TCB", "FPT"]

for code in tickers:
    stock = Vnstock().stock(symbol=code, source='VCI')
    df = stock.quote.history(start='1990-01-01', end='2025-10-01')
```

```
    df = df[['time', 'open', 'high', 'low', 'close', 'volume']]
    df.columns = ['date', 'open', 'high', 'low', 'close', 'volume']
    df.to_csv(f"datack/{code}.csv", index=False)
```

Kết quả: 10 file CSV tương ứng 10 mã cổ phiếu, mỗi file có cột: `date, open, high, low, close, volume`.

3 Mô hình hệ thống Big Data

3.1 Kiến trúc tổng thể

- 1 **NameNode** và 4 **DataNode** (HDFS cluster)
- 1 **Spark Master** và 4 **Spark Worker**
- **Jupyter Notebook** để thực thi và hiển thị kết quả

3.2 Triển khai bằng Docker

Listing 2: Trích đoạn docker-compose.yml

```
version: "2.1"

services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    ports:
      - "9870:9870"
    environment:
      - CLUSTER_NAME=stock
    volumes:
      - hadoop_namenode:/hadoop/dfs/name

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    depends_on:
      - namenode
    deploy:
      replicas: 4
    environment:
      - SERVICE_PRECONDITION=namenode:9870
```

4 Phân tích liệu với PySpark

4.1 Kết nối Spark Cluster

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("Stock_price_analysis") \
    .master("spark://spark-master:7077") \
    .getOrCreate()
```

4.2 Đọc dữ liệu từ HDFS

```
df = spark.read.csv(
    "hdfs://namenode:9000/user/root/datack/*.csv",
    header=True,
    inferSchema=True
)
df.show(5)
```

4.3 Trực quan và các phân tích dữ liệu

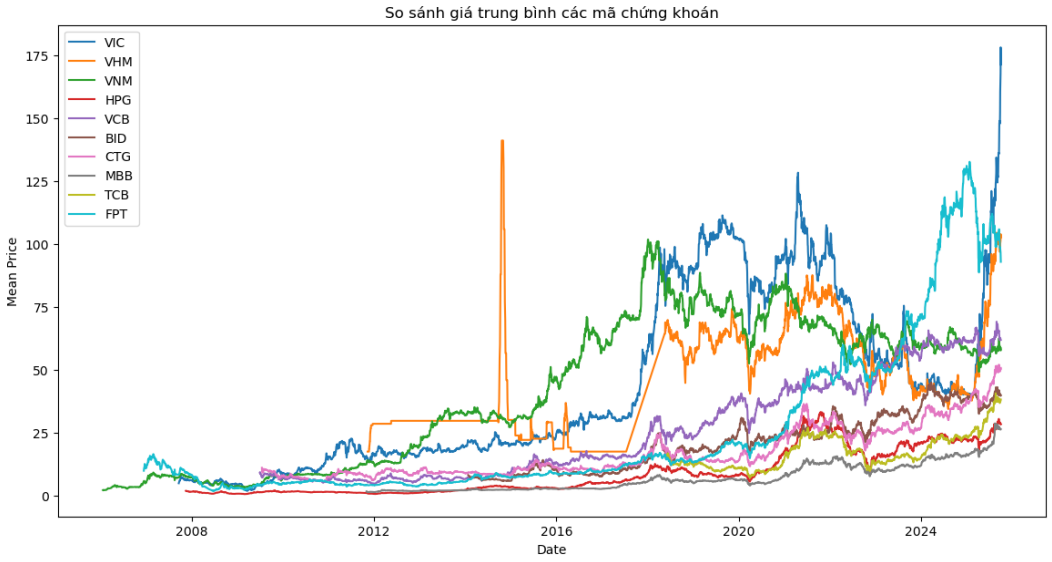
4.3.1 So sánh giá trung bình các mã cổ phiếu

Kết quả: Biểu đồ ở Hình 1 thể hiện xu hướng biến động của giá trung bình (*Mean Price*) của 10 mã cổ phiếu: VIC, VHM, VNM, HPG, VCB, BID, CTG, MBB, TCB và FPT trong giai đoạn từ năm 2006 đến 2025. Dữ liệu được thu thập trực tiếp từ thư viện `vnstock` và được xử lý trên cụm **Spark** để đảm bảo khả năng mở rộng và tốc độ tính toán.

Có thể thấy rõ sự khác biệt về quy mô giá trị giữa các nhóm cổ phiếu:

- Nhóm cổ phiếu vốn hóa lớn như **VIC**, **VHM** và **VCB** có giá trung bình cao và biến động mạnh.
- Các cổ phiếu ngân hàng như **BID**, **CTG**, **MBB**, **TCB** có xu hướng dao động ổn định hơn.
- **FPT** thể hiện mức tăng trưởng ổn định và bền vững trong dài hạn.

Phân tích này cho phép đánh giá sơ bộ xu hướng thị trường và so sánh mức độ biến động giữa các ngành.



Hình 1: So sánh giá trung bình các mã chứng khoán

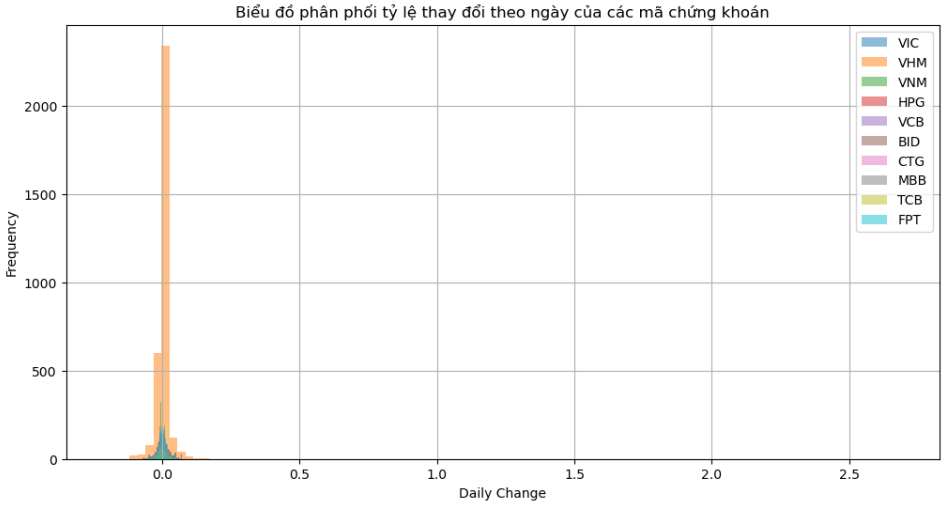
4.3.2 Phân phối tỷ lệ thay đổi giá cổ phiếu theo ngày

Để hiểu rõ hơn mức độ biến động của từng mã cổ phiếu, dữ liệu được tiếp tục xử lý để tính **tỷ lệ thay đổi giá đóng cửa hàng ngày** (*Daily Return*). Việc phân tích phân phối của đại lượng này cho phép đánh giá độ rủi ro (volatility) và mức độ ổn định của từng mã.

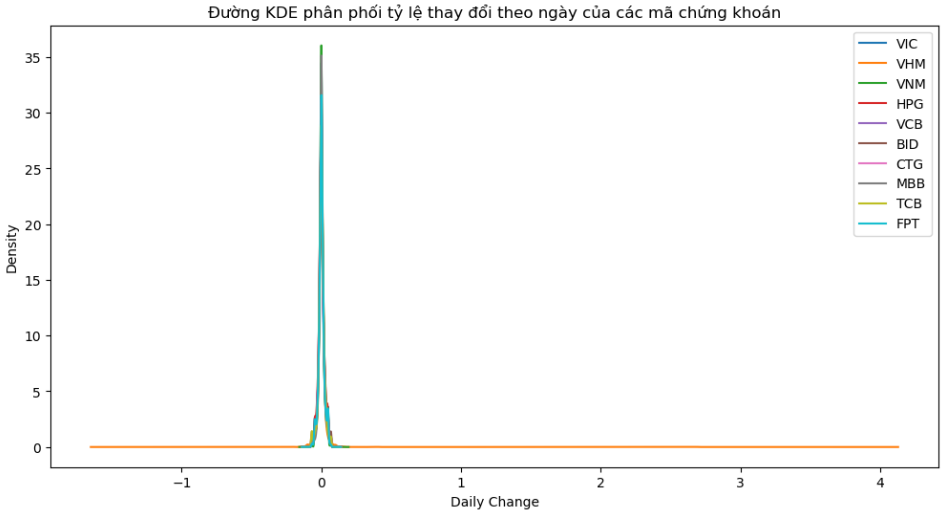
Phân tích kết quả: Hình 2 mô tả phân phối phần trăm thay đổi giá đóng cửa hàng ngày của 10 mã cổ phiếu. Kết quả cho thấy phần lớn các mã có phân phối tập trung quanh 0, thể hiện xu hướng biến động nhỏ trong ngắn hạn, phù hợp với đặc trưng của thị trường Việt Nam.

Một số nhận định cụ thể:

- Các mã **HPG**, **MBB**, **CTG** có phân phối hẹp hơn, thể hiện độ ổn định cao hơn.
- Ngược lại, **VIC** và **VHM** có đuôi phân phối dài, cho thấy biên độ dao động giá lớn hơn và rủi ro cao hơn.
- Phân phối của **FPT** tương đối cân đối và ổn định, phản ánh mức biến động vừa phải.



Hình 2: Phân phối tỷ lệ thay đổi giá theo ngày của các mã chứng khoán.



Hình 3: Đường KDE phân phối tỷ lệ thay đổi theo ngày của các mã chứng khoán

Kết quả này có thể được sử dụng để đánh giá rủi ro đầu tư, xác định danh mục cổ phiếu phù hợp và làm bước tiền đề cho các mô hình dự báo biến động giá trong tương lai.

4.4 Huấn luyện mô hình LSTM

Để dự đoán giá cổ phiếu, bài tập sử dụng mạng nơ-ron hồi quy dài ngắn hạn (LSTM) – một dạng mạng học sâu phù hợp với dữ liệu chuỗi thời gian. Dữ liệu được chia thành hai phần: **train_size = 4477** và **test_size = 434**. Các giá trị được chuẩn hóa trong khoảng [0, 1] bằng `MinMaxScaler` nhằm tăng tốc quá trình hội tụ của mô hình.

Listing 3: Chuẩn bị dữ liệu huấn luyện và xây dựng mô hình LSTM.

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM, Dropout, Dense
```

```
training_set = train[['Mean']].values
sc = MinMaxScaler(feature_range=(0, 1))
training_set_scaled = sc.fit_transform(training_set)

X_train = []
y_train = []
no_of_sample = len(training_set)

for i in range(60, no_of_sample):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])

X_train, y_train = np.array(X_train), np.array(y_train)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

#LSTM model
regressor = Sequential()
regressor.add(LSTM(units = 50, return_sequences = True, input_shape =
    ↪ (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1))

regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

Mô hình LSTM gồm 4 tầng LSTM liên tiếp với 50 nút mỗi tầng và các tầng `Dropout(0.2)` để tránh hiện tượng overfitting. Hàm mất mát được chọn là **Mean Squared Error (MSE)** và bộ tối ưu là **Adam**.

Sau khi huấn luyện, mô hình được sử dụng để dự đoán giá cổ phiếu trong giai đoạn kiểm thử (test set). Kết quả dự đoán được so sánh với giá thực tế để đánh giá độ chính xác của mô hình.

Hình 4: VNM Stock Price Prediction

5 Đánh giá

Hệ thống hoạt động ổn định:

- Dữ liệu được lưu trữ và truy xuất thành công trên HDFS.
- Spark xử lý dữ liệu song song nhanh hơn so với chạy cục bộ.
- Jupyter hiển thị kết quả trực quan hóa hiệu quả.

6 Kết luận và hướng phát triển

6.1 Kết luận

Đồ án đã triển khai thành công hệ thống phân tích và xử lý dữ liệu lớn trong lĩnh vực chứng khoán, sử dụng nền tảng Hadoop và Apache Spark. Hệ thống được thiết kế nhằm thu thập, lưu trữ và xử lý khối lượng dữ liệu cổ phiếu lớn, đảm bảo khả năng mở rộng, phân tán và tính sẵn sàng cao. Quá trình cài đặt và cấu hình các thành phần trong cụm Hadoop-Spark như NameNode, DataNode, Spark Master và Worker được thực hiện hoàn chỉnh, đảm bảo môi trường hoạt động ổn định cho quá trình xử lý dữ liệu.

Việc áp dụng Spark SQL và PySpark cho phép thực hiện các truy vấn, phân tích thống kê và trực quan hóa dữ liệu hiệu quả, góp phần làm rõ các xu hướng biến động giá cổ phiếu trên thị trường. Đồng thời, việc tích hợp mô hình học máy (Machine Learning) phục vụ dự đoán giá cổ phiếu đã minh chứng cho tiềm năng ứng dụng của hệ thống Big Data trong phân tích và ra quyết định tài chính.

Kết quả thực nghiệm cho thấy hệ thống có khả năng xử lý dữ liệu nhanh hơn đáng kể so với các phương pháp xử lý tuần tự truyền thống. Các mô hình hoạt động ổn định trên tập dữ liệu thực tế và cho thấy khả năng mở rộng linh hoạt khi áp dụng với các tập dữ liệu lớn hơn hoặc đa nguồn hơn, bao gồm dữ liệu thời gian thực và dữ liệu phi cấu trúc.

6.2 Hướng phát triển

- Tích hợp Apache Kafka để xử lý luồng dữ liệu thời gian thực.
- Áp dụng Spark MLlib để dự đoán xu hướng giá cổ phiếu.
- Tối ưu hiệu năng bằng cơ chế caching và partitioning trong Spark.

Tài liệu tham khảo

- Stock-Price: `https://github.com/thviet79/Stock-Price`