Faculty of Mathematics
and Information Science

WARSAW UNIVERSITY OF TECHNOLOGY

Predicting Cryptocurrencies Exchange Rates Based on Cryptocurrencies News Sentiment Analysis

Big Data Project

Data Acquisition

Authors:
Maciej Pawilkowski
Hubert Ruczyński
Bartosz Siński
Adrian Stańdo

Warsaw University of Technology
Faculty of Mathematics and Information Science
20 November, 2023

1 Business goal and potential benefits

The goal of this project is to create a system which enables its users to investigate the influence of the latest news articles on exchange rates of cryptocurrencies. Our tool will scrap current exchange rates, and recent news regarding cryptocurrencies in order to perform a sentiment analysis of those messages. Extracted features will be provided into the time-series predictive model that will present estimated exchange rates of selected cryptocurrencies, based on archival data, the most recent trends, and sentiment.

The end users will be able to track current exchange rates, our prediction, and the latest news concerning selected cryptocurrency. This information might help them make the best decision about when to exchange their money. As a result, they may save a lot of money or even find a way to influence main news sources to publish articles that would have positive impact on selected exchange rates.

2 Description of data sources

2.1 Coincap

Description: Coincap provides API with the real time cryptocurrencies data without API key.
Category: Real-time cryptocurrencies values.
Number of records and inflow frequency: Inflow frequency: 1 per 10 seconds.
Record description: id, **symbol**, **currencySymbol** (Symbol of cryptocurrency, ex. BTC - bitcoin), type, **rateUsd** (the exchange rate in USD), and **timestamp**.
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: Very high, no missing values.
Source format: JSON.
ML preprocessing: -
Requests Limits: 200 requests per minute, exchange values update each 10 seconds.
Link: https://coincap.io/

2.2 Alpha Vantage

Description: Alpha Vantage website provides API with financial market data such as time series stock data, digital & crypto currency exchange rates and most importantly market news data which we will use in this project. Market news are both real-time and historical and come from global news outlets. They cover stocks, cryptocurrencies, forex and others. News data consists of title, url to the article, summary and the source. Additionally API returns sentiment analysis information. However we want to perform this analysis by ourselves and therefore we will not use this feature.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: Inflow frequency: all changes from last hour (set with request parameters).
Record description: title, url, authors (a list), summary, banner_image (url), source

(name of magazine, ex. Forbes), category_within_source, source_domain (web-page of source, ex. www.forbes.com), topics (a list of topics describing the news), overall_sentiment_score, overall_sentiment_label, ticker_sentiment, **time_published**.
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: High, only columns not needed by us contain missing values.
Source format: JSON, does not contain article content.
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: 5 API requests per minute, 100 API requests per day. Possible academic access to premium plan with up to 1200 API requests per minute.
Link: https://www.alphavantage.co/

2.3 Crypto Compare

Description This API provides various crypto market data such as current price and trading information, block chain data and social stats for given coins. On top of that we can get information about latest news articles related to crypto currencies. We can choose categories and source feeds of requested articles. Similarly to the previous data source returned news data consists of title, url to the article, summary and the source.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: API is called once an hour, it returns the last 100 articles. **Record description**: id, guid, imageurl, title, url, **body** (The first part of the main text), tags, lang (language), upvotes (number of users upvotes), downvotes, categories, source_info, source, **published_on**.
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: Very high, no missing values, does not contain article content.
Source format: JSON.
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: Lifetime - 250 000 calls, Day - 50 000 calls, Minute - 2 500 calls.
Link: https://min-api.cryptocompare.com/

2.4 CryptoPanic

Description JSON API that gives access to the most recent posts, indicators and sentiment data. It offers news articles and updates from various sources related to cryptocurrencies and block-chain technology. It also returns readers feedback in shape of number of likes, dislikes, comments, etc.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: Inflow frequency: called once an hour, returns last 20 articles.
Record description: title, **timestamp**, id, url, **crypto** (a code for cryptocurrency which is talked about), **content** (small part of an article), **datetime**.

Collection time period: We will collect data during the winter holidays break for one week.
Data quality: High, no missing values, does not contain full article content.
Source format: JSON.
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: We are limited to 200 last news posts, which we can request 5 times per second without max request limit.
Link: https://cryptopanic.com/developers/api/about

2.5 NewsData

Description An API containing news articles regarding cryptocurrencies.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: Inflow frequency: all changes from last hour (set by query parameter).
Record description: article_id, title, link, keywords, creator, video_url, description (a one sentence description of article), **content** (full content of article), **timestamp**, image_url, source_id, source_priority, country, category, language.
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: High, only columns not needed by us contain missing values.
Source format: JSON.
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: We are limited to 200 credits per day to search and download the News data. Each credit allows download of 10 articles.
Link: https://newsdata.io

2.6 NewsAPI

Description It is a simple HTTP REST API for searching and retrieving live articles from all over the web. API doesn't return entire articles only title, description, first 200 signs, author, and source. It allows us to request articles based on language, publication date and even based on keyword or phrase.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: Inflow frequency: all changes from last hour (set by query parameter).
Record description: source, author, title, description (2 sentences), url, urlToImage, content (partial, ends with..), **publishedAt**
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: Very high, no missing values, or inconsistencies, article content is not available.

Source format: JSON.
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: We have access to articles with 24h delay, and can't pull articles older than one month. We can also make only 100 requests per day.
Link: `https://newsapi.org/`

2.7 Yahoo news

Description: Searching with Google search engine (News tab) articles with keywords "yahoo finance <cryptocurrency name>" from last hour. In <cryptocurrency name> we put names of all observed cryptocurrencies in our project. Afterwards, the content of the article from the searched yahoo websites are scrapped.
Category: Cryptocurrencies related news data.
Number of records and inflow frequency: Inflow frequency: all changes from last hour.
Record description:-
Collection time period: We will collect data during the winter holidays break for one week.
Data quality: Quite high, articles content is of high quality, but it includes various languages.
Source format: -
ML preprocessing: We will retrieve the sentiment with spaCy from provided texts, and provide only this information to the model.
Requests Limits: ?
Link: -

3 Article content scrapping

Because almost all considered news source return only links to the articles (without content), we decided to put another layer between NiFi and data source - our custom HTTP server. Its main aim is to imitate having REST API, which returns also article content. This additional layer is implemented for all news sources (except NewsData, which is the only one to return content). It is implemented in Python using FastAPI and uvicorn libraries for HTTP server, BeautifulSoup and Selenium for scrapping. An additional preprocessing step done by this layer is converting dates (from different formats and attribute names) to timestamp format with the same attribute name.

4 Used technologies

Technologies which will be used in the project:

1. **Data flow:** `Apache NiFi`
   Raw data from APIs will be downloaded and preprocessed using `Apache NiFi` . Cryptocurrency data will be sent to both `HDFS Archive`, and `Apache Kafka Buffer`, as

archival data will be used to cyclically train the model, and recent data will be used to fine-tune it. The crypto-news data will be also send to both places as as we tread the batch layer as a kind of back-up.

2. **Data storage:** `HDFS`, `Apache Hive` and `Apache HBase`
   As presented in the Figure 1, cryptocurrencies and crypto-news data will be saved in `HDFS Archive`. Additionally we will redirect the results of sentiment analysis here, whereas the stocks predictions will be accessible from the serving layer only. `Apache Hive` will be used as a serving layer for the Batch Layer data, whereas `Apache HBase` will be serving the current data from a Speed Layer.

3. **Data processing:** `Apache Spark`
   `Apache Spark` scripts will be widely used in our project at multiple steps, as presented in the Figure 1. First and foremost, it has components dedicated for NLP analysis such as calculating sentiment, thus it will be used for the ML part of the project. Moreover, we will use it to transform the data coming from `Apache Kafka Buffer`, and `HDFS Archive`, before they will be transported to the respective Serving Layer views.

4. **Speed Layer Buffer:** `Apache Kafka`
   `Apache Kafka` will serve as a buffer holding the recent cryptocurrency data used for the fine tuning of our model.

5. **Presentation Layer:** `Tableau` / `Jupyter Notebook` with `Apache Spark`
   We have to strategies concerning the presentation layer. The first one focuses on the usage of Business Intelligence (BI) tool, like `Tableau` to create a clean Dashboard with proper visualizations. The approach is however not certain, as we are not sure how BI systems can interact with our data sources, and if they are enabled in free versions. The second one is a backup, where we prepare visualizations in `Jupyter Notebook`, and transform data via `Apache Spark`. Both approaches will include a curve presenting recent exchange rates, and our prediction for the near future for selected cryptocurrencies. Moreover, on this plots we will mark the timestamps of latest news occurrences and print those news / their titles.

4.1 Architecture Diagram

The figure 1 presents our solution.

5 Data Flow

We will use `Apache NiFi` to collect data from selected APIs, and orchestrate the process of putting it into HDFS, and forwarding it to `Apache Kafka`. We have designed four different NiFi flows for various sources. The first one considers the CryptoNews platform, and is presented on figure 2. As we can see in this case we use InvokeHTTP to get answer from API, later we transform the data, and convert it to parquet, and finally put it to `HDFS`.

Another flow from Figure 3 is used for the currencies exchange rate data, where we get the data, accumulate it for 60 seconds, convert to AVRO format, and later put it into `HDFS`.

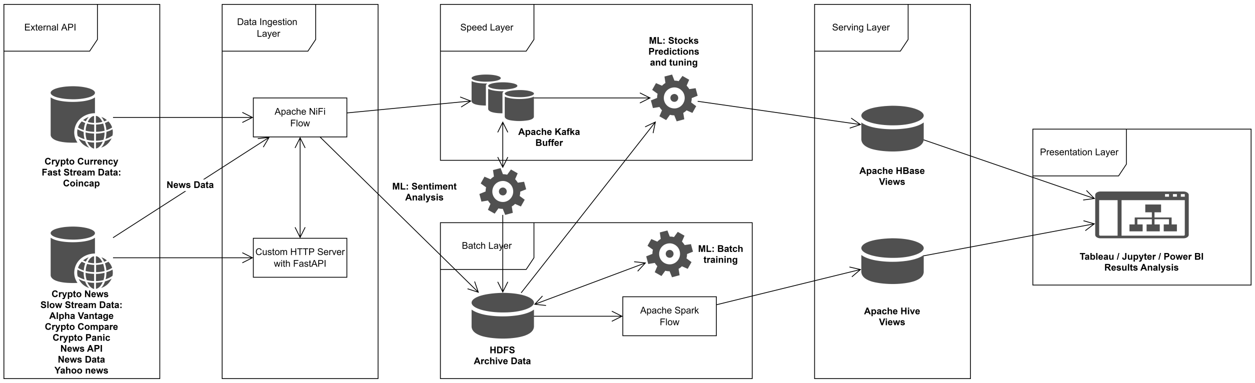
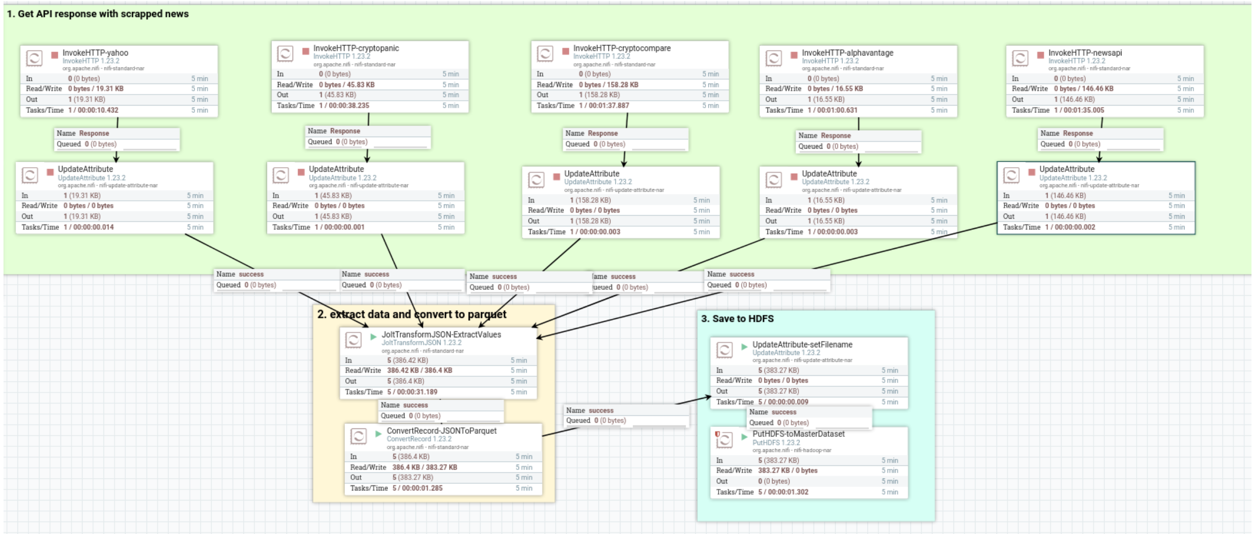

External API
Data Ingestion Layer
Speed Layer
Serving Layer
Presentation Layer
Batch Layer
Crypto Currency Fast Stream Data: Coincap
News Data
Crypto News Slow Stream Data: Alpha Vantage Crypto Compare Crypto Panic News API News Data Yahoo news
Apache NiFi Flow
Custom HTTP Server with FastAPI
Apache Kafka Buffer
ML: Stocks Predictions and tuning
ML: Sentiment Analysis
ML: Batch training
HDFS Archive Data
Apache Spark Flow
Apache HBase Views
Apache Hive Views
Tableau / Jupyter / Power BI Results Analysis

Figure 1: Architecture plan of the proposed solution.

Finally, the rest of our news articles sources have similar data acquisition and transformation structure, presented on the Figure 4.



1. Get API response with scrapped news

```
InvokeHTTP-yahoo
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 19.31 KB       5 min
Out         1 (19.31 KB)             5 min
Tasks/Time  1 / 00:00:10.432         5 min

InvokeHTTP-cryptopanic
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 45.83 KB       5 min
Out         1 (45.83 KB)             5 min
Tasks/Time  1 / 00:00:38.235         5 min

InvokeHTTP-cryptocompare
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 158.28 KB      5 min
Out         1 (158.28 KB)            5 min
Tasks/Time  1 / 00:01:37.887         5 min

InvokeHTTP-alphavantage
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 16.55 KB       5 min
Out         1 (16.55 KB)             5 min
Tasks/Time  1 / 00:01:00.631         5 min

InvokeHTTP-newsapi
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar
In          0 (0 bytes)              5 min
Read/Write  0 bytes / 146.46 KB      5 min
Out         1 (146.46 KB)            5 min
Tasks/Time  1 / 00:01:35.005         5 min

Name  Response
Queued  0 (0 bytes)

UpdateAttribute
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          1 (19.31 KB)             5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1 (19.31 KB)             5 min
Tasks/Time  1 / 00:00:00.014         5 min

UpdateAttribute
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          1 (45.83 KB)             5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1 (45.83 KB)             5 min
Tasks/Time  1 / 00:00:00.001         5 min

UpdateAttribute
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          1 (158.28 KB)            5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1 (158.28 KB)            5 min
Tasks/Time  1 / 00:00:00.003         5 min

UpdateAttribute
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          1 (16.55 KB)             5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1 (16.55 KB)             5 min
Tasks/Time  1 / 00:00:00.003         5 min

UpdateAttribute
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          1 (146.46 KB)            5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         1 (146.46 KB)            5 min
Tasks/Time  1 / 00:00:00.002         5 min

Name  success
Queued  0 (0 bytes)
```

2. extract data and convert to parquet

```
JoltTransformJSON-ExtractValues
JoltTransformJSON 1.23.2
org.apache.nifi - nifi-standard-nar
In          5 (386.42 KB)            5 min
Read/Write  386.42 KB / 386.4 KB     5 min
Out         5 (386.4 KB)             5 min
Tasks/Time  5 / 00:00:31.189         5 min

Name  success
Queued  0 (0 bytes)

ConvertRecord-JSONToParquet
ConvertRecord 1.23.2
org.apache.nifi - nifi-standard-nar
In          5 (386.4 KB)             5 min
Read/Write  386.4 KB / 383.27 KB     5 min
Out         5 (383.27 KB)            5 min
Tasks/Time  5 / 00:00:01.285         5 min

Name  success
Queued  0 (0 bytes)
```

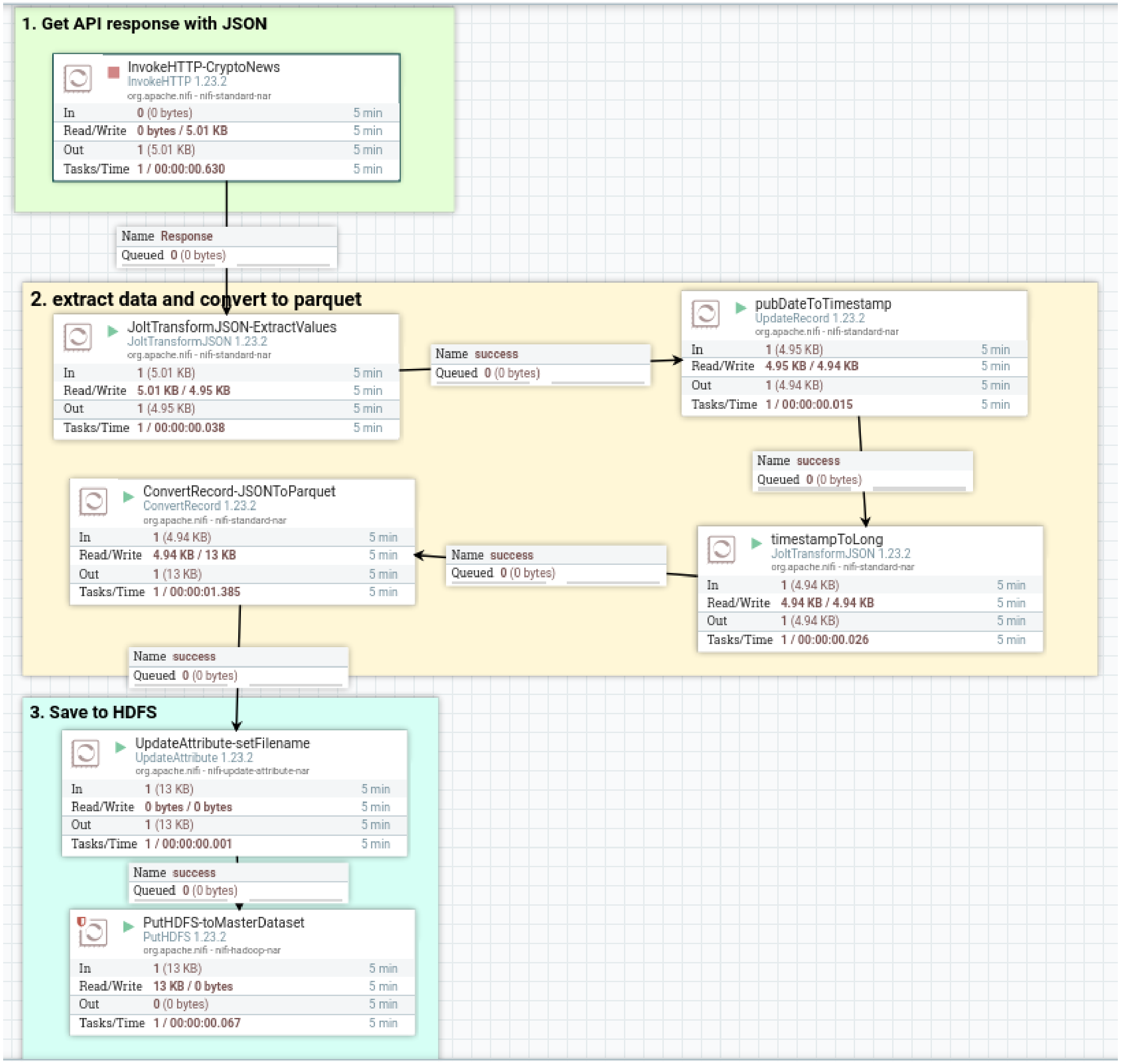
3. Save to HDFS

```
UpdateAttribute-setFilename
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar
In          5 (383.27 KB)            5 min
Read/Write  0 bytes / 0 bytes        5 min
Out         5 (383.27 KB)            5 min
Tasks/Time  5 / 00:00:00.009         5 min

Name  success
Queued  0 (0 bytes)

PutHDFS-toMasterDataset
PutHDFS 1.23.2
org.apache.nifi - nifi-hadoop-nar
In          5 (383.27 KB)            5 min
Read/Write  383.27 KB / 0 bytes      5 min
Out         0 (0 bytes)              5 min
Tasks/Time  5 / 00:00:01.302         5 min
```

Figure 4: NiFi flow for news articles data.

It is important to notice, that the input for `HDFS` and `Apache Kafka` are raw data without additional filtering. This step will be done directly before the model training, however, in the middle we will invoke an `Apache Spark` script which will take out the data articles from `Apache Kafka`, and compute the sentiment scores for it, to later put those information in both `Apache Kafka` and `HDFS`.

The next step of the data flow is getting the outcomes from models present in speed layer, and forward them to `Apache HBase` in the serving layer. Additionally, we will also froward the data from `HDFS` archive to `Apache Hive`. The data from both serving layer sources will be finally used in presentation layer.



1. Get API response with JSON

InvokeHTTP-CryptoNews
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 5.01 KB	5 min
Out	1 (5.01 KB)	5 min
Tasks/Time	1 / 00:00:00.630	5 min

Name Response
Queued 0 (0 bytes)

2. extract data and convert to parquet

JoltTransformJSON-ExtractValues
JoltTransformJSON 1.23.2
org.apache.nifi - nifi-standard-nar

In	1 (5.01 KB)	5 min
Read/Write	5.01 KB / 4.95 KB	5 min
Out	1 (4.95 KB)	5 min
Tasks/Time	1 / 00:00:00.038	5 min

Name success
Queued 0 (0 bytes)

pubDateToTimestamp
UpdateRecord 1.23.2
org.apache.nifi - nifi-standard-nar

In	1 (4.95 KB)	5 min
Read/Write	4.95 KB / 4.94 KB	5 min
Out	1 (4.94 KB)	5 min
Tasks/Time	1 / 00:00:00.015	5 min

Name success
Queued 0 (0 bytes)

timestampToLong
JoltTransformJSON 1.23.2
org.apache.nifi - nifi-standard-nar

In	1 (4.94 KB)	5 min
Read/Write	4.94 KB / 4.94 KB	5 min
Out	1 (4.94 KB)	5 min
Tasks/Time	1 / 00:00:00.026	5 min

Name success
Queued 0 (0 bytes)

ConvertRecord-JSONToParquet
ConvertRecord 1.23.2
org.apache.nifi - nifi-standard-nar

In	1 (4.94 KB)	5 min
Read/Write	4.94 KB / 13 KB	5 min
Out	1 (13 KB)	5 min
Tasks/Time	1 / 00:00:01.385	5 min

Name success
Queued 0 (0 bytes)

3. Save to HDFS

UpdateAttribute-setFilename
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar

In	1 (13 KB)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	1 (13 KB)	5 min
Tasks/Time	1 / 00:00:00.001	5 min

Name success
Queued 0 (0 bytes)

PutHDFS-toMasterDataset
PutHDFS 1.23.2
org.apache.nifi - nifi-hadoop-nar

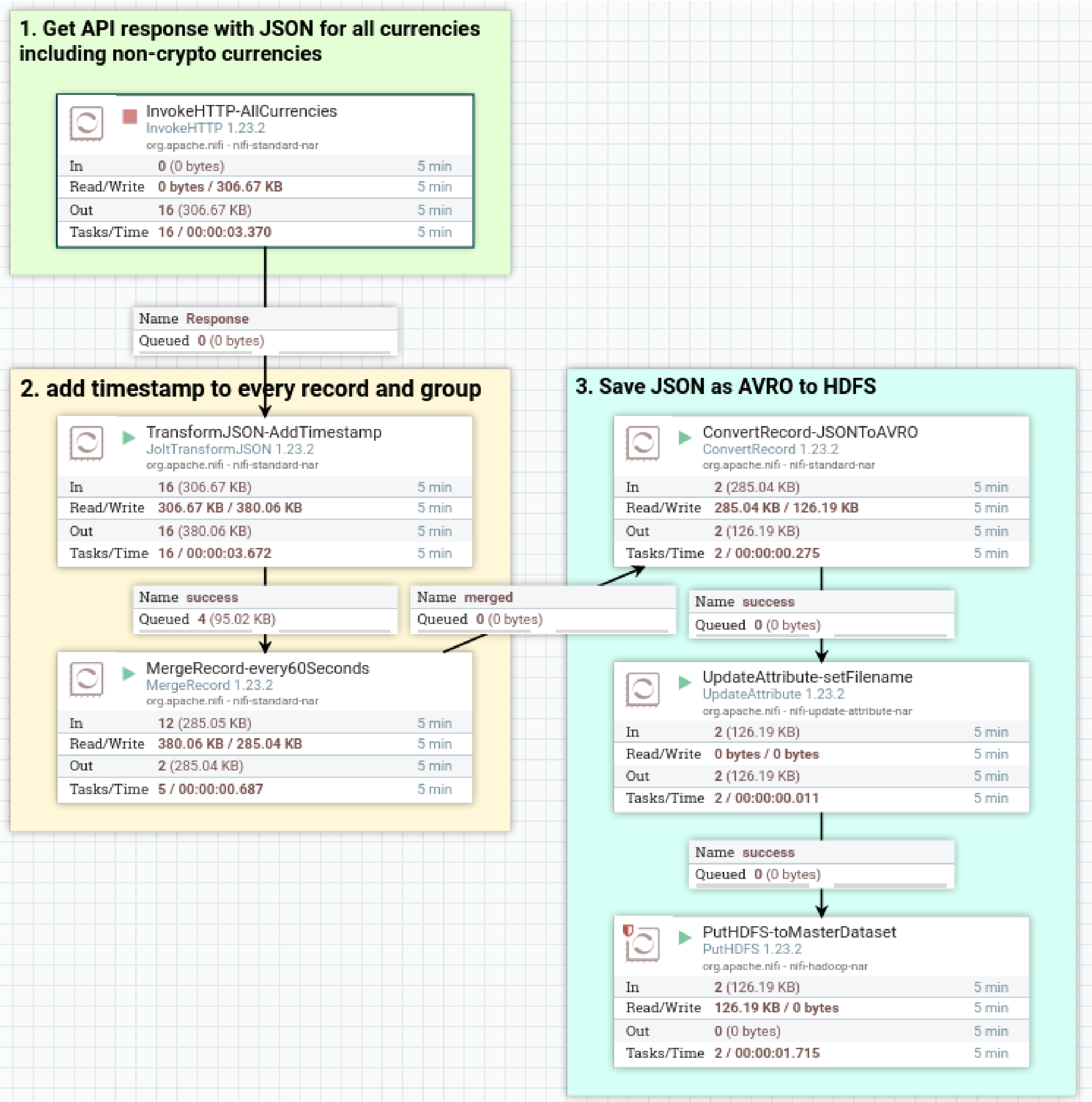
In	1 (13 KB)	5 min
Read/Write	13 KB / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	1 / 00:00:00.067	5 min

Figure 2: NiFi flow for CryptoNews.

6 Further works

6.1 ML tasks

Our solution consists of three machine learning sub-tasks. The first one, triggered from `Kafka` focuses on analyzing the sentiment of fresh news articles. As a result, we add information about article data, which cryptocurrencies it is describing, and the sentiment values to both `Kafka Buffer`, and `HDFS Archive`. The next tasks are connected to the main topic of our project which is a time-series forecasting of selected cryptocurrencies stock exchange rates. The first one is training the model in a long-term manner. We want to set a time window at night, where we spend one hour on training the model, based on the archive data,



1. Get API response with JSON for all currencies including non-crypto currencies

InvokeHTTP-AllCurrencies
InvokeHTTP 1.23.2
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 306.67 KB	5 min
Out	16 (306.67 KB)	5 min
Tasks/Time	16 / 00:00:03.370	5 min

Name Response
Queued 0 (0 bytes)

2. add timestamp to every record and group

TransformJSON-AddTimestamp
JoltTransformJSON 1.23.2
org.apache.nifi - nifi-standard-nar

In	16 (306.67 KB)	5 min
Read/Write	306.67 KB / 380.06 KB	5 min
Out	16 (380.06 KB)	5 min
Tasks/Time	16 / 00:00:03.672	5 min

Name success
Queued 4 (95.02 KB)

MergeRecord-every60Seconds
MergeRecord 1.23.2
org.apache.nifi - nifi-standard-nar

In	12 (285.05 KB)	5 min
Read/Write	380.06 KB / 285.04 KB	5 min
Out	2 (285.04 KB)	5 min
Tasks/Time	5 / 00:00:00.687	5 min

Name merged
Queued 0 (0 bytes)

3. Save JSON as AVRO to HDFS

ConvertRecord-JSONToAVRO
ConvertRecord 1.23.2
org.apache.nifi - nifi-standard-nar

In	2 (285.04 KB)	5 min
Read/Write	285.04 KB / 126.19 KB	5 min
Out	2 (126.19 KB)	5 min
Tasks/Time	2 / 00:00:00.275	5 min

Name success
Queued 0 (0 bytes)

UpdateAttribute-setFilename
UpdateAttribute 1.23.2
org.apache.nifi - nifi-update-attribute-nar

In	2 (126.19 KB)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	2 (126.19 KB)	5 min
Tasks/Time	2 / 00:00:00.011	5 min

Name success
Queued 0 (0 bytes)

PutHDFS-toMasterDataset
PutHDFS 1.23.2
org.apache.nifi - nifi-hadoop-nar

In	2 (126.19 KB)	5 min
Read/Write	126.19 KB / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	2 / 00:00:01.715	5 min

Figure 3: NiFi flow for cryptocurrencies exchange rates.

stock exchange rates, and news article features (ex. sentiment). It is placed in the batch layer due to the cyclical nature of the task. The second part is however the part of the speed layer, as the trained model will be additionally fine-tuned, based on new data, and it will predict the exchange rates for a few time horizons.

6.2 Use of batch and stream processing

The stream processing will be used to address real-time sentiment analysis, fine-tune the model, and provide forecasts of exchange rates. The batch layer on the other hand will be used to train the model on a cyclical, daily basis, and it will provide archival data information for the serving and presentation layer.

6.3 Presentation

Our main goal is to deploy the interface with the usage of business intelligence (BI) solutions such as Tableau, or Power BI in order to provide a simple-to-use interface, however, we are not sure if `Apache HBase` and `Apache Hive` are supported within the free tiers of those solutions. In case they will not work we plan to provide a Jupyter Notebook with proper elements.

Regardless of the technical aspect we plan to provide a line plot that shows us the previous values of a given cryptocurrency and mark the forecasts for at least 3 different time horizons. Additionally, we plan to provide a visualization that shows us the appearances of articles regarding one, or more cryptocurrencies in the given time window, and mark whether their sentiment scores were positive or negative.

7 Work division

The Table 1 present the division of work among the team members for whole project.

Table 1: Work division between the team members.

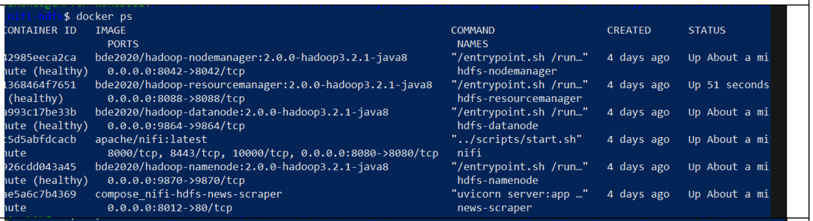
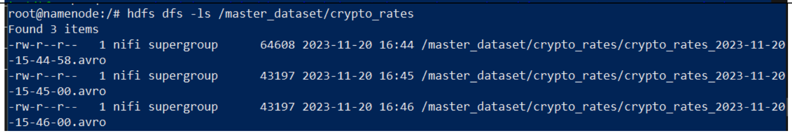
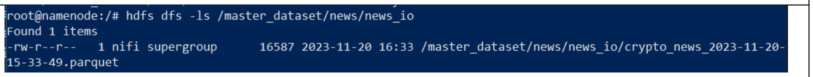
Task Name	**Team Members**
Data Source Analysis	Maciej Pawlikowski, Bartosz Siński
Project Planning	Hubert Ruczyński, Adrian Stańdo
NiFi Flow Implementation	Adrian Stańdo
Speed Layer Implementation (incl. Spark)	Maciej Pawlikowski, Bartosz Siński
Batch Layer Implementation (incl. Spark)	Hubert Ruczyński
ML Implementation	Maciej Pawlikowski, Bartosz Siński
Serving Layer Implementation	Adrian Stańdo
Presentation Layer Implementation	Hubert Ruczyński

8 Technical solution

We plan to use multiple Docker containers to run all the services. If it is not possible to run them all on one machine, we will use a service like Tailscale, which provides a zero-configuration VPN, and we will connect our local machines together.

9 Testing

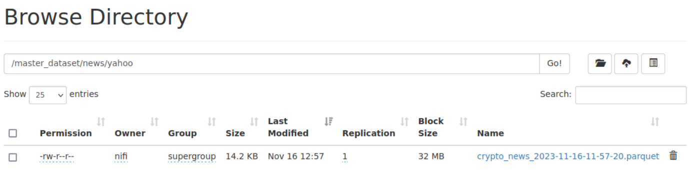
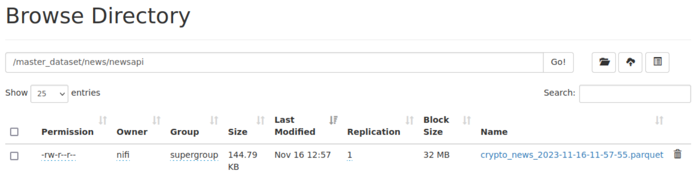
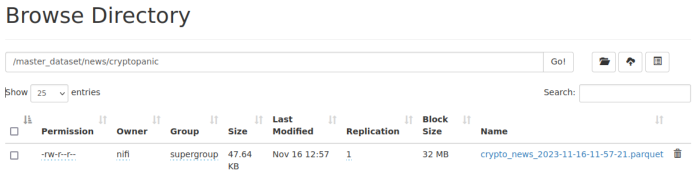
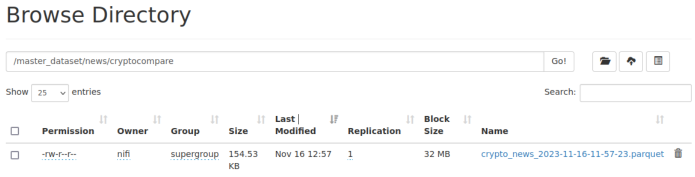
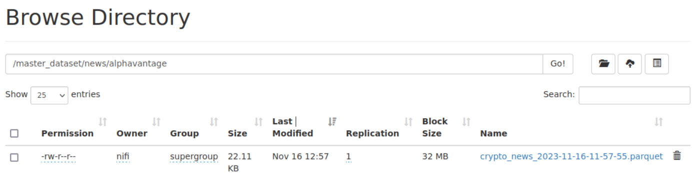
Tests Description

Test Name	Objective	Steps	Expected Results	Actual Results
NiFi and HDFS status	Check if containers with NiFi and HDFS are running correctly.	1.Build the NiFi and HDFS containers executing docker-compose in folder with .yaml configuration file. 2.Run docker ps and check status of containers.	All containers are up and have “healthy” status.	
Load cryptocurrency rates data to HDFS	Check if NiFi Flow that loads data from coincap.io API to HDFS works correctly.	1.Run crypto_rates_api_to_master flow in NiFi. 2. Check if new data appeared in the /master_dataset/crypto_rates on Hadoop server.	Data with current timestamp appears in the HDFS.	
Load news data to HDFS	Check if NiFi Flow that loads data from newsdata.io API to HDFS works correctly.	1.Run NewsAPI_news_api_to_master flow in NiFi. 2. Check if new data apeared in the /master_dataset/news/newsio on Hadoop server.	Data with current timestamp appears in the HDFS.	

```
nifi-hdfs$ docker ps
CONTAINER ID   IMAGE                                                  COMMAND                  CREATED        STATUS
                 PORTS                                                  NAMES
42985eeca2ca   bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   4 days ago     Up About a mi
nute (healthy)   0.0.0.0:8042->8042/tcp                                 hdfs-nodemanager
1368464f7651   bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   4 days ago     Up 51 seconds
 (healthy)       0.0.0.0:8088->8088/tcp                                 hdfs-resourcemanager
a993c17be33b   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   4 days ago     Up About a mi
nute (healthy)   0.0.0.0:9864->9864/tcp                                 hdfs-datanode
c5d5abfdcacb   apache/nifi:latest                                       "../scripts/start.sh"    4 days ago     Up About a mi
nute             8000/tcp, 8443/tcp, 10000/tcp, 0.0.0.0:8080->8080/tcp  nifi
926cdd043a45   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   4 days ago     Up About a mi
nute (healthy)   0.0.0.0:9870->9870/tcp                                 hdfs-namenode
ae5a6c7b4369   compose_nifi-hdfs-news-scraper                           "uvicorn server:app …"   4 days ago     Up About a mi
nute             0.0.0.0:8012->80/tcp                                   news-scraper
```

```
root@namenode:/# hdfs dfs -ls /master_dataset/crypto_rates
Found 3 items
-rw-r--r--   1 nifi supergroup      64608 2023-11-20 16:44 /master_dataset/crypto_rates/crypto_rates_2023-11-20
-15-44-58.avro
-rw-r--r--   1 nifi supergroup      43197 2023-11-20 16:45 /master_dataset/crypto_rates/crypto_rates_2023-11-20
-15-45-00.avro
-rw-r--r--   1 nifi supergroup      43197 2023-11-20 16:46 /master_dataset/crypto_rates/crypto_rates_2023-11-20
-15-46-00.avro
```

```
root@namenode:/# hdfs dfs -ls /master_dataset/news/news_io
Found 1 items
-rw-r--r--   1 nifi supergroup      16587 2023-11-20 16:33 /master_dataset/news/news_io/crypto_news_2023-11-20-
15-33-49.parquet
```

Scrape news articles data and load it to HDFS	Check if our article scrapping component works correctly and loads data to HDFS.	1.Check if compose_nifi-hdfs-news-scraper container is up and runing. 2. Run NewsScraped_to_master flow in NiFi. 3. Check if new data appeared in the folders which are corresponding to scrapred data sources in the /master_dataset/news on Hadoop server.	Data with current timestamp appears in the HDFS.	    

Browse Directory

/master_dataset/news/yahoo

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	nifi	supergroup	14.2 KB	Nov 16 12:57	1	32 MB	crypto_news_2023-11-16-11-57-20.parquet

Browse Directory

/master_dataset/news/newsapi

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	nifi	supergroup	144.79 KB	Nov 16 12:57	1	32 MB	crypto_news_2023-11-16-11-57-55.parquet

Browse Directory

/master_dataset/news/cryptopanic

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	nifi	supergroup	47.64 KB	Nov 16 12:57	1	32 MB	crypto_news_2023-11-16-11-57-21.parquet

Browse Directory

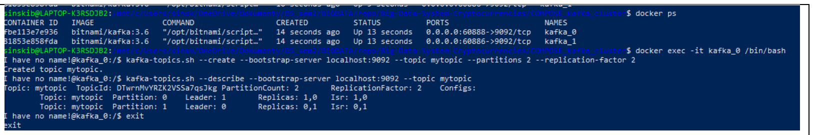
/master_dataset/news/cryptocompare

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	nifi	supergroup	154.53 KB	Nov 16 12:57	1	32 MB	crypto_news_2023-11-16-11-57-23.parquet

Browse Directory

/master_dataset/news/alphavantage

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	nifi	supergroup	22.11 KB	Nov 16 12:57	1	32 MB	crypto_news_2023-11-16-11-57-55.parquet

Kafka Status	Check if containers with Kafka brokers are running correctly.	1.Build the Kafka containers executing docker-compose in folder with .yaml configuration file. 2.Run docker ps and check status of containers. 3.Create sample topic. 4.Check the status of two Kafka brokers on newly created topic.	All containers are up and have “healthy” status. The topic is created and distributed among both Kafka containers.	

```
sinskib@LAPTOP-K3RSDJB2:/mnt/c/Users/sinsk/OneDrive/Dokumenty/DS_sem2/BIGDATA/repo/Big-Data-System-Cryptocurrencies/COMPOSE_kafka_cluster$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS          PORTS                      NAMES
fbe113e7e936   bitnami/kafka:3.6    "/opt/bitnami/script…"   14 seconds ago   Up 13 seconds   0.0.0.0:60888->9092/tcp   kafka_0
31853e858fda   bitnami/kafka:3.6    "/opt/bitnami/script…"   14 seconds ago   Up 13 seconds   0.0.0.0:60886->9092/tcp   kafka_1
sinskib@LAPTOP-K3RSDJB2:/mnt/c/Users/sinsk/OneDrive/Dokumenty/DS_sem2/BIGDATA/repo/Big-Data-System-Cryptocurrencies/COMPOSE_kafka_cluster$ docker exec -it kafka_0 /bin/bash
I have no name!@kafka_0:/$ kafka-topics.sh --create --bootstrap-server localhost:9092 --topic mytopic --partitions 2 --replication-factor 2
Created topic mytopic.
I have no name!@kafka_0:/$ kafka-topics.sh --describe --bootstrap-server localhost:9092 --topic mytopic
Topic: mytopic  TopicId: DTwrnMvYRZK2VSSa7qsJkg PartitionCount: 2       ReplicationFactor: 2    Configs:
        Topic: mytopic  Partition: 0    Leader: 1       Replicas: 1,0   Isr: 1,0
        Topic: mytopic  Partition: 1    Leader: 0       Replicas: 0,1   Isr: 0,1
I have no name!@kafka_0:/$ exit
exit
```