Crypto market analysis prediction in distributed systems

DAT500 - Final Project Report

Asahi Cantu Moreno
University of Stavanger, Norway
a.cantumoreno@stud.uis.no

Daniel Urdal
University of Stavanger, Norway
d.urdal@stud.uis.no

ABSTRACT

The creation of cryptocurrency and its exponential adoption over the last three years has led to the creation of many different exchange and tools to trade it via Internet. Unlike common stock trading, cryptocurrency works nonstop all year with simple and advanced interfaces, allowing anyone with a minimum investment to enter the market and exchange assets in real time, with high liquidity and asset availability. Because of the early state of the technology and the socioeconomic-political factors affecting the regulation of the cryptocurrency markets as well as the speed at which transactions are made has led to a high risk-high volatility market, making very hard to analyze or predict prices that can lead to profits for a chosen currency. This academic report contains a deep analysis on crypto market and aims to generate a prediction model by analyzing historical data, the news and events that can potentially affect the market and its impact on the prices by using distributed systems Hadoop® and Apache Spark® for data storage and interaction. Machine Learning models are implemented as well for a big data set. Further analysis, results and potential for future work is presented in this report.

KEYWORDS

Cryptocurrency, Machine Learning, Hadoop, Spark, Neural Networks, Distributed Systems

1 INTRODUCTION

It was not long ago that bitcoin and the concept of Blockchain appeared for the first time on the Web as a white paper, October 31st 2008 to be more precise "Bitcoin: A Peer-to-Peer Electronic Cash System" [15] Said the document in its heading. It was uncertain by that time how disruptive such technology could be, and the importance of its role nowadays. A decade later, Bitcoin has consolidated as a strong and reliable asset in which people trust to invest and transfer money. The term "Cryptocurrency" was born. Little after its emission and open source code publication, several teams and individuals started re adapting its code and creating different electronic assets, slowly with its adoption a philosophy more people started trading on their private markets. It would not take too long to foresee the creation of a community and a market craving to trade different assets. As a result of this crescent demand Cryptocurrency market exchanges were created. From the financial perspective, cryptocurrency markets are nowadays allowing people who did not have access to credits or bank accounts before to enter into a different economic world were markets do not sleep, moreover assets can be instantly traded, changed and transferred. Years ago, this market consolidated and matured, data is produced massively, regulations or innovations, even political situations have made the market very peculiar and volatile, something unprecedented in the financial trading.

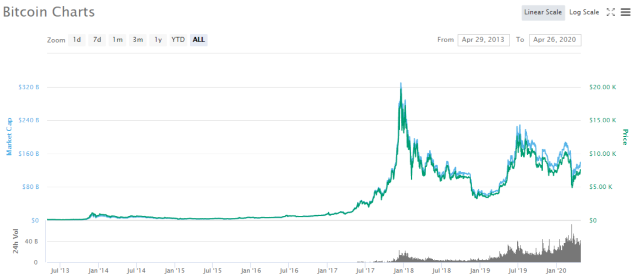


Figure 1: Bitcoin price chart (taken from [11])

Bitcoin price volatility generated a high-risk market, but also a lot of interest and market capitalization has been set to the cryptocurrency assets. On April 2020, the total market capitalization of all cryptocurrencies is up to $219,165,492,833.00 USD [12] which translates in a growing market accessible to anyone with sufficient resources around the world to trade. As the time goes by more resources and assets will be added resulting in an expected increase of profit depending on the investment, but daily fluctuations are visible as well for any asset.

In addition to this, the impact of social media and published news about recent technological progress and Information Technology improvements are expected to be a key player in the decision different traders made to trade cryptocurrency assets. There might be a direct correlation in what is published on social media, news and the price fluctuation of the market in short term basis. It is therefore the aim of this report to explore, discover and predict potential market prices through distributed systems to show the potential of creating a scalable product trained to predict short term prices according to past fluctuations and media publications. The system used, algorithms, data set and brief explanation of the challenges in addition to obtain results will be explained in detail though this report. Provided results can be also used as a basis for future work development and algorithm improvements.

2 BACKGROUND

The growing importance of cryptocurrency in a global market where facing a potential economic crisis in which fiat money might not be anymore the common exchange medium between different

Supervised by Jayachander Surbiryala, Tomasz Wiktorski Tomasz Wiktorski,Markus Pettersen Markus Pettersen.

Project in Computer Science (DAT500), IDE, UiS
2020.

parties. The technological development and amazing availability of growing data to explore are the key factors that led to research on cryptocurrency and try to analyze through machine learning and distributed systems the potential price fluctuations. Another point to consider: Social media plays nowadays an important role in our daily life, more and more politicians use these platforms for instant communications, researchers announce publications and press uses them as a main platform to publish breaking news. All in all, we cannot take out anymore social media from the impact of our economic and political situations. It is therefore important to consider this as a key factor in the market decisions made on daily basis and the price fluctuations of the assets.

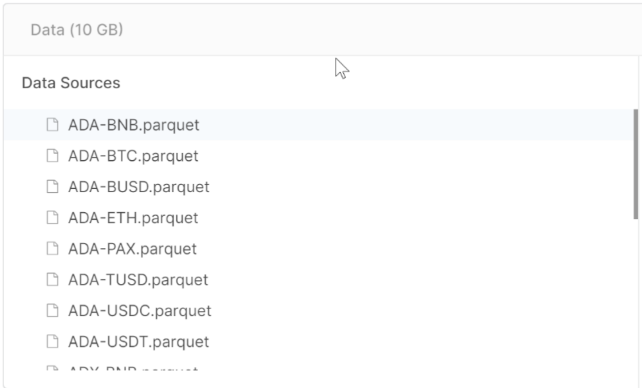
The main idea is to use a big cryptocurrency data set and a trained model of cryptocurrency related news to predict short term future prices for all the provided asset by using Machine learning algorithms.

2.1 Cryptocurrency

Cryptocurrency can be defined as “a digital asset designed to work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets.” [3] Cryptocurrencies use decentralized control as opposed to centralized digital currency and central banking systems, which means no central entity has full control or power over the asset, but the community maintains it through systems known as “nodes” and each transaction is recorded in a public ledger known as the “blockchain”[1]

2.1.1 Cryptocurrency Data set. The data set used for this project is a Binance exchange data set extracted from the portal Kaggle[18]. The data set is titled “Binance Full History” and contains 1 minute candlestick data for a total of 786 cryptocurrency pairs traded on the exchange Web portal Binance.com[6]. The features of this data set are:

- Data types: python-pandas parquet files[2]
- Data size : 10GB
- Number of files : 764



Data (10 GB)

Data Sources

ADA-BNB.parquet
ADA-BTC.parquet
ADA-BUSD.parquet
ADA-ETH.parquet
ADA-PAX.parquet
ADA-TUSD.parquet
ADA-USDC.parquet
ADA-USDT.parquet

Figure 2: Cryptocurrency data set. Parquet file (taken from [18])

2.1.2 Cryptocurrency data set features and details. A minute candlestick represents the maximum and minimum price at which any given asset was sold. It also contains information about the open price (price at the beginning of the time) and close price (end of time).

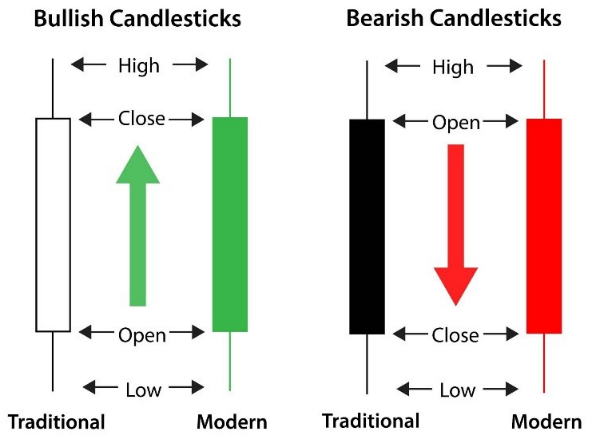


Figure 3: Candlestick Representations

A series of candlestick represent the market fluctuation in a given period of time.

2.2 News data set

The news data set was extracted from the web portal “CoinTelegraph.com”[3] . The data set was created using python scripts due to the lack of availability of a full set.

The features of this data set are:

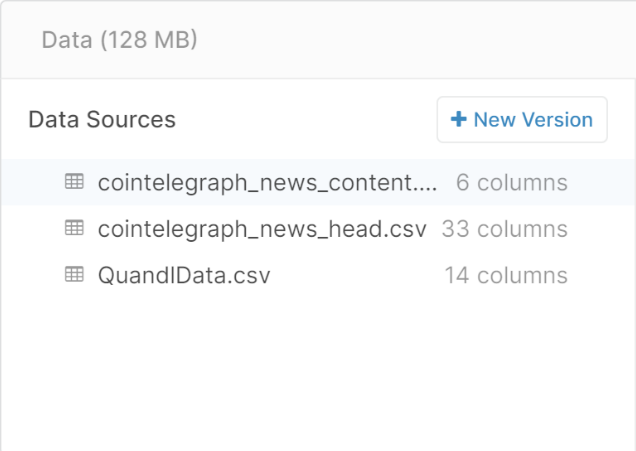
- Data types: 2 csv files total size is 128 MB.
- More than 20,000 different news from 2013 until present for human-written news about cryptocurrency and related subjects.

[1] A blockchain is a time-stamped series of immutable records of data that is managed by a cluster of computers not owned by any single entity. Each of these blocks of data (i.e. block) is secured and bound to each other using cryptographic principles (i.e. chain). public to anyone and resilient to changes. It is therefore virtually impossible to commit fraud if more than 50% of the computational resources belong to the community and not the entity intending to hack or modify the blockchain[5]

[2] pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python [2]. Parquet files are compressed binary files specially built to reduce storage size of big data files and improve loading performance.

[3] The most recent news about crypto industry at Cointelegraph. Latest news about bitcoin, ethereum, blockchain, mining, cryptocurrency prices and more[13]

- Header file. Contains relevant metadata information about the news (such as title, thumbnail, creation date, etc.).
- Content file. Contains the body of the news and other metadata like the number of views and times it has been shared as well as image references for it.



Data (128 MB)

Data Sources

cointelegraph_news_content....	6 columns
cointelegraph_news_head.csv	33 columns
QuandlData.csv	14 columns

Figure 4: Cryptocurrency News Data set [8]

2.3 Related works

Several attempts and previous works from the data science community have been made to predict stock market price using Machine learning. The procedures and models are not so much different from the cryptocurrency model since both are time series models and its current input depend directly on past data. There is, however one big difference: the price volatility, which leads to the research no model training and adjustment to compensate for the possible increases and at the same time adapt the model to avoid overfitting.

For this project and among other sources three jobs were considered as role models:

Xavier approaches in his research document different Machine Learning models and mechanisms to predict tock market prices by performing Moving Average and LSTM[4] training models. His results show the differences in the accuracy for the two different models and make evident that LSTM model performs better, although several adjustments have to be done depending on the asset to model. [4]

Pablo Castilla takes the theory of LSTM into practice in a kaggle document, and figures the basic mecanishm for data processing, normalization and training[16].

Finally the work from Sagar addresses the machine learning mechanisms for cryptocurrency trading. Part of his research is used to tune and develop the Machine learning algorithms.

3 ENVIRONMENT SETUP

The system used for this for this project consisted of four virtual machines (VMs) running Ubuntu Server 16.04. Apache Hadoop version 3.2.1 was installed on each machine, and they were configured to run as a Hadoop cluster using the YARN resource manager. They were configured such that one VM acted as a master node, while the other three acted as slave nodes. Apache Spark was installed using the 3.0.0 preview version, as this was the only version supporting Hadoop 3.2.1. In order to get Spark running on the VMs, Python version 3.8.1 was compiled from source and installed alongside the default Python 2.7.12.

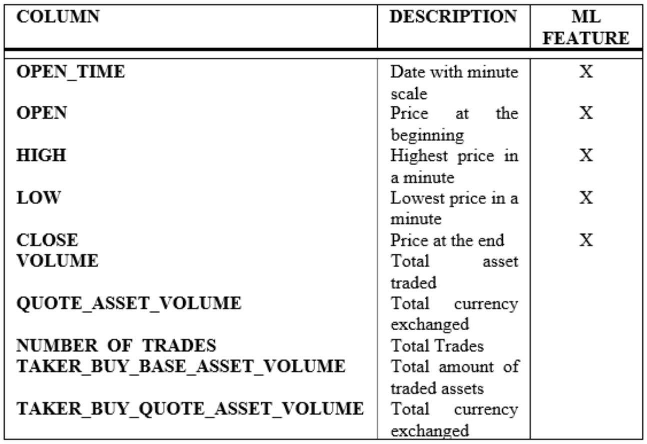
4 DATA PROCESSING

The goal is to combine both data sets, transform them when necessary to reduce size, features and use them with ML training algorithms that enable an accurate cryptocurrency market price prediction.

4.1 Cryptocurrency data set processing

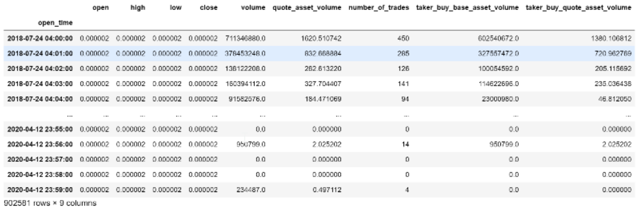
Because this data set was machine-generated little had to be done to process it. It was important, however, to examine the contained information and extract just the necessary one in real time. Because of data size, and the technical resources limitations it was not possible to extract and store the required information, but rather open the file, select the main features and use them in real time for machine learning algorithm to train and generate a model, and the close it. Uncompressed data reached 70 GB of space, whereas compressed data set just 9 GB.

4.1.1 Data set description and feature selection. Each parquet file in the data set contains the corresponding information as shown in Figure 5.



COLUMN	DESCRIPTION	ML FEATURE
OPEN_TIME	Date with minute scale	X
OPEN	Price at the beginning	X
HIGH	Highest price in a minute	X
LOW	Lowest price in a minute	X
CLOSE	Price at the end	X
VOLUME	Total asset traded	
QUOTE_ASSET_VOLUME	Total currency exchanged	
NUMBER OF TRADES	Total Trades	
TAKER_BUY_BASE_ASSET_VOLUME	Total amount of traded assets	
TAKER_BUY_QUOTE_ASSET_VOLUME	Total currency exchanged	

Figure 5: Cryptocurrency features



open_time	open	high	low	close	volume	quote_asset_volume	number_of_trades	taker_buy_base_asset_volume	taker_buy_quote_asset_volume
2018-07-24 04:00:00	0.000002	0.000002	0.000002	0.000002	711346880.0	1620.510742	450	602540672.0	1380.106812
2018-07-24 04:01:00	0.000002	0.000002	0.000002	0.000002	378453248.0	832.668884	265	327557472.0	720.962769
2018-07-24 04:02:00	0.000002	0.000002	0.000002	0.000002	138122208.0	282.613220	126	100054592.0	205.115692
2018-07-24 04:03:00	0.000002	0.000002	0.000002	0.000002	160394112.0	327.704407	141	114622696.0	235.036438
2018-07-24 04:04:00	0.000002	0.000002	0.000002	0.000002	91582576.0	184.471069	94	23000980.0	46.812050
...	...	...	...	...	...	...	...	...	...
2020-04-12 23:55:00	0.000002	0.000002	0.000002	0.000002	0.0	0.000000	0	0.0	0.000000
2020-04-12 23:56:00	0.000002	0.000002	0.000002	0.000002	950799.0	2.025202	14	950799.0	2.025202
2020-04-12 23:57:00	0.000002	0.000002	0.000002	0.000002	0.0	0.000000	0	0.0	0.000000
2020-04-12 23:58:00	0.000002	0.000002	0.000002	0.000002	0.0	0.000000	0	0.0	0.000000
2020-04-12 23:59:00	0.000002	0.000002	0.000002	0.000002	234487.0	0.497112	4	0.0	0.000000

902581 rows × 9 columns

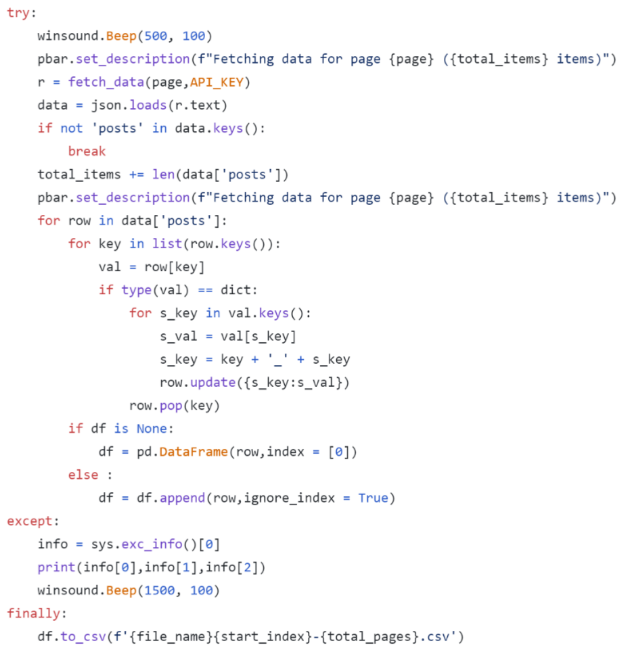
Figure 6: Example of parquet file HOT-ETH.parquet

[4] Acronymm for Long Short-Term Neural network. A special kind of Neuran network model that is trained based on past data.

From 5 features marked with 'X' where selected to be processed and to train the Machine Learning model.

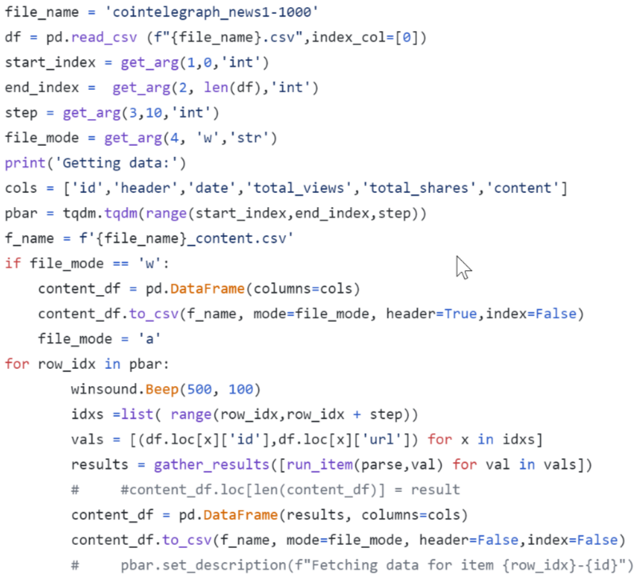
4.2 Cryptocurrency news data processing

This data set was non-existent, it had to be generated by creating a python script to bring the information from the web portal "coinmarketcap.com". The script uses web scraping methodology and extracts the news set according to the given parameters. A total of 24519 news articles had to be gathered and processed to generate the data set in two steps. The first step consists on getting the header information from the web portal, whereas the second brings the content of a specific news article. The script was generated and run in a Hadoop cluster to better control and preserve information in case of data loss. A part of the algorithm is shown, full code description and implementation can be seen in [10].



```
try:
    winsound.Beep(500, 100)
    pbar.set_description(f"Fetching data for page {page} ({total_items} items)")
    r = fetch_data(page,API_KEY)
    data = json.loads(r.text)
    if not 'posts' in data.keys():
        break
    total_items += len(data['posts'])
    pbar.set_description(f"Fetching data for page {page} ({total_items} items)")
    for row in data['posts']:
        for key in list(row.keys()):
            val = row[key]
            if type(val) == dict:
                for s_key in val.keys():
                    s_val = val[s_key]
                    s_key = key + '_' + s_key
                    row.update({s_key:s_val})
                row.pop(key)
        if df is None:
            df = pd.DataFrame(row,index = [0])
        else :
            df = df.append(row,ignore_index = True)
except:
    info = sys.exc_info()[0]
    print(info[0],info[1],info[2])
    winsound.Beep(1500, 100)
finally:
    df.to_csv(f'{file_name}{start_index}-{total_pages}.csv')
```

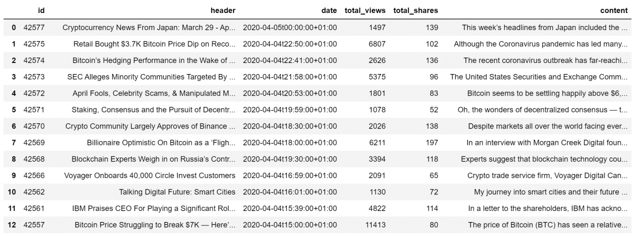
Figure 7: Code implementation for news heading extraction. Data is extracted into a dictionary and later saved on a pandas DataFrame



```
file_name = 'cointelegraph_news1-1000'
df = pd.read_csv (f"{file_name}.csv",index_col=[0])
start_index = get_arg(1,0,'int')
end_index =  get_arg(2, len(df),'int')
step = get_arg(3,10,'int')
file_mode = get_arg(4, 'w','str')
print('Getting data:')
cols = ['id','header','date','total_views','total_shares','content']
pbar = tqdm.tqdm(range(start_index,end_index,step))
f_name = f'{file_name}_content.csv'
if file_mode == 'w':
    content_df = pd.DataFrame(columns=cols)
    content_df.to_csv(f_name, mode=file_mode, header=True,index=False)
    file_mode = 'a'
for row_idx in pbar:
        winsound.Beep(500, 100)
        idxs =list( range(row_idx,row_idx + step))
        vals = [(df.loc[x]['id'],df.loc[x]['url']) for x in idxs]
        results = gather_results([run_item(parse,val) for val in vals])
        #     #content_df.loc[len(content_df)] = result
        content_df = pd.DataFrame(results, columns=cols)
        content_df.to_csv(f_name, mode=file_mode, header=False,index=False)
        #     pbar.set_description(f"Fetching data for item {row_idx}-{id}")
```

Figure 8: Code implementation for news article extraction, data is directly saved in a pandas DataFrame and updated in a csv file. http request interactions are intensive and took close to 3 days of continuous execution to retrieve the full data set

Once created the data set, it was properly cleaned and processed to be used for natural language processing just one file (cointelegraph_news_content.csv) was used to train the model for being the one with the article content.



	id	header	date	total_views	total_shares	content
0	42577	Cryptocurrency News From Japan: March 29 - Ap...	2020-04-05t00:00:00+01:00	1497	139	This week's headlines from Japan included the ...
1	42575	Retail Bought $3.7K Bitcoin Price Dip on Reco...	2020-04-04t22:50:00+01:00	6807	102	Although the Coronavirus pandemic has led many...
2	42574	Bitcoin's Hedging Performance in the Wake of ...	2020-04-04t22:41:00+01:00	2626	136	The recent coronavirus outbreak has far-reachi...
3	42573	SEC Alleges Minority Communities Targeted By ...	2020-04-04t21:58:00+01:00	5375	96	The United States Securities and Exchange Comm...
4	42572	April Fools, Celebrity Scams, & Manipulated M...	2020-04-04t20:53:00+01:00	1801	83	Bitcoin seems to be settling happily above $6,...
5	42571	Staking, Consensus and the Pursuit of Decentr...	2020-04-04t19:59:00+01:00	1078	52	Oh, the wonders of decentralized consensus — t...
6	42570	Crypto Community Largely Approves of Binance ...	2020-04-04t18:30:00+01:00	2026	138	Despite markets all over the world facing ever...
7	42569	Billionaire Optimistic On Bitcoin as a 'Fligh...	2020-04-04t18:00:00+01:00	6211	197	In an interview with Morgan Creek Digital foun...
8	42568	Blockchain Experts Weigh in on Russia's Contr...	2020-04-04t19:30:00+01:00	3394	118	Experts suggest that blockchain technology cou...
9	42566	Voyager Onboards 40,000 Circle Invest Customers	2020-04-04t16:59:00+01:00	2091	65	Crypto trade service firm, Voyager Digital Can...
10	42562	Talking Digital Future: Smart Cities	2020-04-04t16:01:00+01:00	1130	72	My journey into smart cities and their future ...
11	42561	IBM Praises CEO For Playing a Significant Rol...	2020-04-04t15:39:00+01:00	4822	114	In a letter to the shareholders, IBM has ackno...
12	42557	Bitcoin Price Struggling to Break $7K — Here'...	2020-04-04t15:00:00+01:00	11413	80	The price of Bitcoin (BTC) has seen a relative...

Figure 9: Cryptocurrency news content example

5 ALGORITHM IMPLEMENTATIONS

Unlike other Machine Learning algorithm applications to systems and problem, stock trading belongs to a particular branch of algorithm implementation: sequence prediction problems, which tend to prediction of current and next values depends on the history and past records, events do not occur independently. For these situations RNN[5] algorithms are applied to the cryptocurrency candlestick data set[14].

On the other hand, for the news feed data set it was decided to implement NLP[6] algorithms and clustering for unclassified text,

[5] Recurrent Neural Networks, is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence[10]
[6] Natural Language Processing

since the data set is not labeled. The intention of such classification was to better understand the topics for the news, to isolate, classify them and try to find a pattern to categorize the news. Subsequently a sentiment analysis algorithm can be performed to generate a label that determines whether a specific article is positive or negative, and how meaningful it is for the trading in during the next hours or days. The assumption is that by this results forecasts can be more accurate as they can take articles in real time and merge them with the candlestick data to adapt the training model into a more realistic prediction.

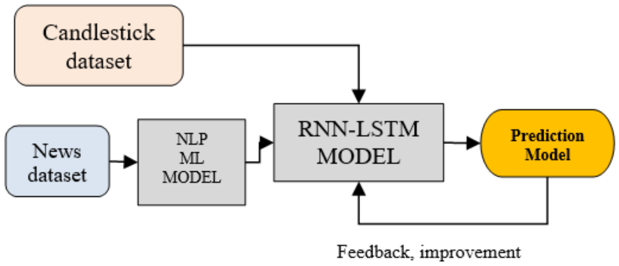


Figure 10: ML Training model prototype

5.1 Cryptocurrency analysis algorithm

Memory-based networks such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks allow for predictions to be remembered and forgotten; this can be useful when predicting extreme changes in a short space of time via the remembering of previous examples. Compared to other models, RNN and LSTM are suited towards sequential data such as stock prices, whereas other models must take a non-sequential, sample-independent interpretation of the data, considering only a single input and output sample at a time [9]

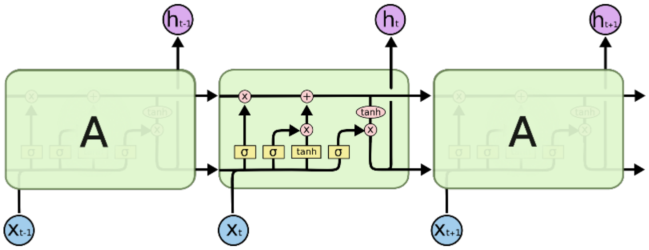


Figure 11: Example of a RNN-LSTM model

To implement RNN-LSTM approach the python TensorFlow[7] deep learning model package was used and adjusted for the best-case scenario in which the loss function and consequent prediction were as accurate as possible without much overfitting. Several examples were observed, the model was retrained for all the cryptocurrency assets using spark functionality with Hadoop ass a file system [20].

5.2 Text and technical sentiment analysis algorithm

News data set was intended to be classified and processed by using Natural Language processing tools and algorithms to find patterns in the whole dataset. Normal processes for NLP processing require (among others):

- String tokenization
- Stop words removal
- Lexicon normalization
- Lemmatization method was applied
- Word2Vet approach implemented as well

Several NLP and sentiment analysis algorithms were used to classify the model and try obtaining the news topics:

- K-Means Clustering
- Treating with an unlabeled data set for cryptocurrency news required a machine learning model to try to process and generate corresponding clusters in order to identify the main topics
- Sentiment analysis
- Common sentiment analysis was performed over the full dataset. This approach aimed to find and tag an approximate positive, neutral or negative sentiment to each document.
- Pre-Trained data models
- The existence of complex machine learning models based on big scientific documents were used to try to identify and label each article
- Text summarization
- Extractive summarization approach was performed to try to identify the main topics

6 CODE IMPLEMENTATIONS

The majority of the algorithms were implemented thinking on a distributed system approach in which data intensive processes and redundancy systems could be able to handle the provided information. The configuration employed is listed below:

- Hadoop Cluster (Ubuntu System)
- 1 Name Node (Master)
- 3 Data Nodes (Slaves)
- Apache Spark as the processing interface with PySpark
- Spark Dataframes
- RDD's and distributed processes
- Jupyter notebook for python scripting
- Tensorflow as the machine learning package
- Intended to use SparkML
- NLP Packages and tools:
- NLTK
- Spacy
- VaderSentiment
- Textblob
- Textacy
- Gensim

[7] TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. Package can be downloaded from https://www.tensorflow.org

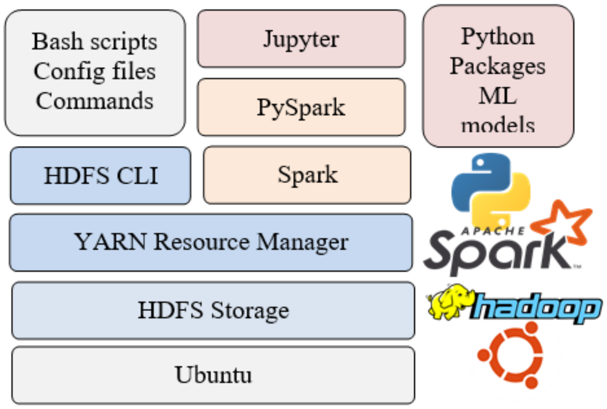
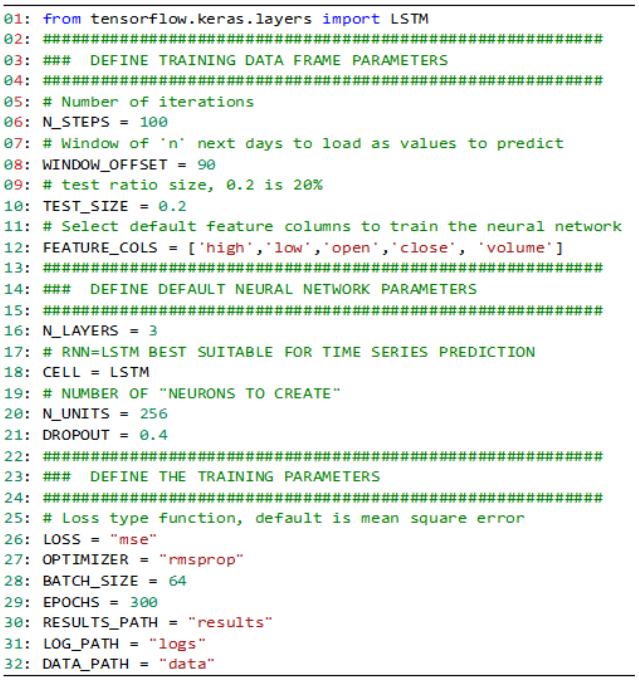


Figure 12: Distributed system architecture



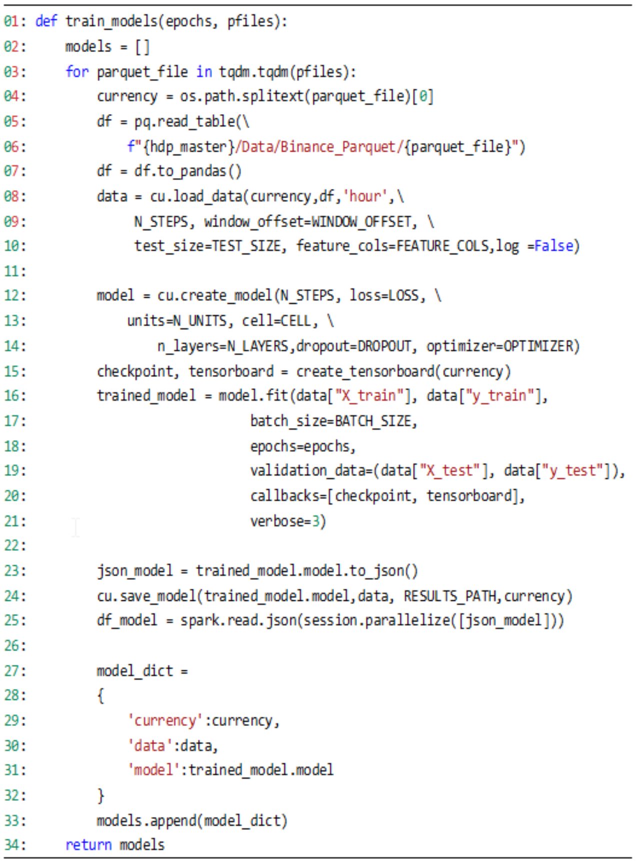
```
from tensorflow.keras.layers import LSTM
##########################################################
###  DEFINE TRAINING DATA FRAME PARAMETERS
##########################################################
# Number of iterations
N_STEPS = 100
# Window of 'n' next days to load as values to predict
WINDOW_OFFSET = 90
# test ratio size, 0.2 is 20%
TEST_SIZE = 0.2
# Select default feature columns to train the neural network
FEATURE_COLS = ['high','low','open','close', 'volume']
##########################################################
###  DEFINE DEFAULT NEURAL NETWORK PARAMETERS
##########################################################
N_LAYERS = 3
# RNN=LSTM BEST SUITABLE FOR TIME SERIES PREDICTION
CELL = LSTM
# NUMBER OF "NEURONS TO CREATE"
N_UNITS = 256
DROPOUT = 0.4
##########################################################
###  DEFINE THE TRAINING PARAMETERS
##########################################################
# Loss type function, default is mean square error
LOSS = "mse"
OPTIMIZER = "rmsprop"
BATCH_SIZE = 64
EPOCHS = 300
RESULTS_PATH = "results"
LOG_PATH = "logs"
DATA_PATH = "data"
```

Figure 13: Fragment from file crypto_params.py

6.1 Cryptocurrency dataset code implementation

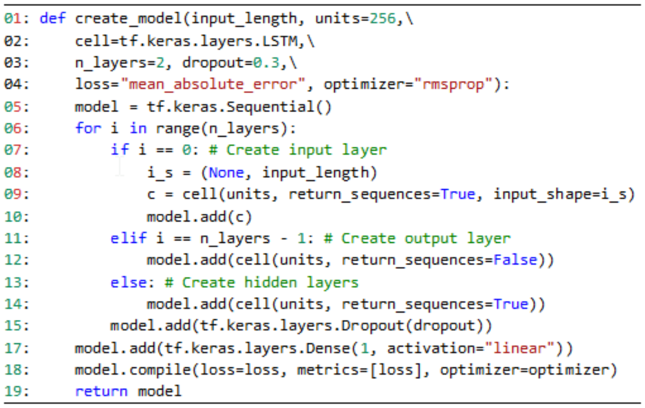
The code for the cryptocurrency candlestick dataset was divided in three different python files

- Settings file **crypto_params.py**
  - Contains a predefined set of variables, resulted from the training and tuning of the RNN-LSTM model. The most optimistic approach ad loss function implemented can be visualized here
- ML algorithm **crypto_utils.py**
  - Contains a code snippet for the ML training model and the configuration of its layers. The way it is called and used by TensorFlow
- Data set reading and training in Spark **sprk.ipynb**
  - Is a Jupyter notebook that contains the main functionality and coordination to call the settings and utils files, respectively. In addition it sets the environment to open a spark environment and connect to the Hadoop YARN resource manager, which allows the system to read the parquet files in a distributed manner.
  - Once the files are called, the ML model is called to be trained based on the obtained data set and the result is saved in a spark json format. This allows spark to save back the trained model in the distributed system, eventually as the data keeps growing the model will be fed in an iterative process to readjust the prediction



```
def train_models(epochs, pfiles):
    models = []
    for parquet_file in tqdm.tqdm(pfiles):
        currency = os.path.splitext(parquet_file)[0]
        df = pq.read_table(\
            f"{hdp_master}/Data/Binance_Parquet/{parquet_file}")
        df = df.to_pandas()
        data = cu.load_data(currency,df,'hour',\
             N_STEPS, window_offset=WINDOW_OFFSET, \
             test_size=TEST_SIZE, feature_cols=FEATURE_COLS,log =False)

        model = cu.create_model(N_STEPS, loss=LOSS, \
            units=N_UNITS, cell=CELL, \
                n_layers=N_LAYERS,dropout=DROPOUT, optimizer=OPTIMIZER)
        checkpoint, tensorboard = create_tensorboard(currency)
        trained_model = model.fit(data["X_train"], data["y_train"],
                           batch_size=BATCH_SIZE,
                           epochs=epochs,
                           validation_data=(data["X_test"], data["y_test"]),
                           callbacks=[checkpoint, tensorboard],
                           verbose=3)

        json_model = trained_model.model.to_json()
        cu.save_model(trained_model.model,data, RESULTS_PATH,currency)
        df_model = spark.read.json(session.parallelize([json_model]))

        model_dict =
        {
            'currency':currency,
            'data':data,
            'model':trained_model.model
        }
        models.append(model_dict)
    return models
```

Figure 14: Fragment from file crypto_utils.py



```
def create_model(input_length, units=256,\
    cell=tf.keras.layers.LSTM,\
    n_layers=2, dropout=0.3,\
    loss="mean_absolute_error", optimizer="rmsprop"):
    model = tf.keras.Sequential()
    for i in range(n_layers):
        if i == 0: # Create input layer
            i_s = (None, input_length)
            c = cell(units, return_sequences=True, input_shape=i_s)
            model.add(c)
        elif i == n_layers - 1: # Create output layer
            model.add(cell(units, return_sequences=False))
        else: # Create hidden layers
            model.add(cell(units, return_sequences=True))
        model.add(tf.keras.layers.Dropout(dropout))
    model.add(tf.keras.layers.Dense(1, activation="linear"))
    model.compile(loss=loss, metrics=[loss], optimizer=optimizer)
    return model
```

Figure 15: Fragment from file crypto_sprk.ipynb

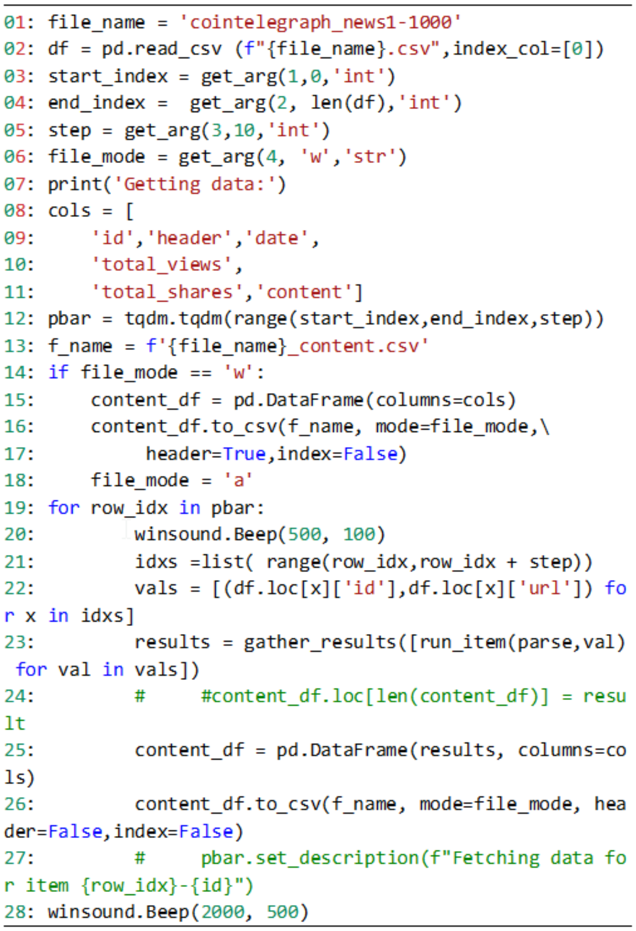
7 NEWSFEED DATA SET CODE IMPLEMENTATION

The newsfeed data set code required a special treatment. It consisted of two stages:

7.0.1 *Stage 1. Newsfeed data set extraction.*

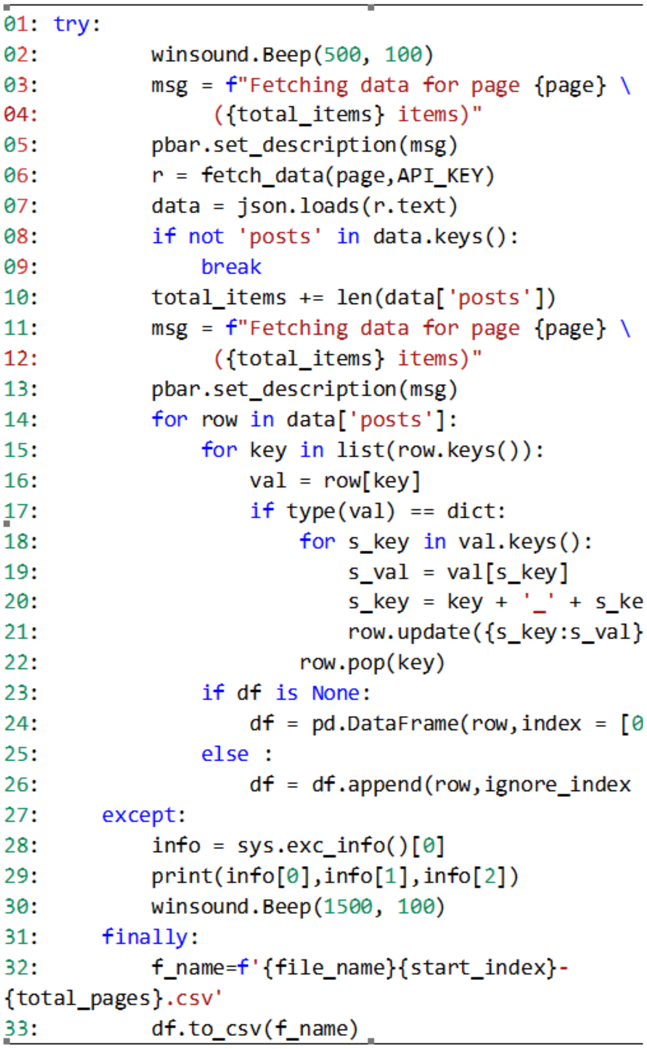
Since the whole data was not available on the web, a python script had to be developed in order to store such the information from the known sources. The implementation of such script is very important due to the lack of the information on the web. The script sets the basis for anyone else to use it and implement the code for any other newsfeed to assemble different data repositories. Both data set and code implementation have been published and made open for anyone to use it and improve it if required. The algorithm was developed in two scripts:

- Newsfeed headings extraction
- Newsfeed articles extraction



```
file_name = 'cointelegraph_news1-1000'
df = pd.read_csv (f"{file_name}.csv",index_col=[0])
start_index = get_arg(1,0,'int')
end_index =  get_arg(2, len(df),'int')
step = get_arg(3,10,'int')
file_mode = get_arg(4, 'w','str')
print('Getting data:')
cols = [
    'id','header','date',
    'total_views',
    'total_shares','content']
pbar = tqdm.tqdm(range(start_index,end_index,step))
f_name = f'{file_name}_content.csv'
if file_mode == 'w':
    content_df = pd.DataFrame(columns=cols)
    content_df.to_csv(f_name, mode=file_mode,\
         header=True,index=False)
    file_mode = 'a'
for row_idx in pbar:
        winsound.Beep(500, 100)
        idxs =list( range(row_idx,row_idx + step))
        vals = [(df.loc[x]['id'],df.loc[x]['url']) fo
r x in idxs]
        results = gather_results([run_item(parse,val)
 for val in vals])
        #      #content_df.loc[len(content_df)] = resu
lt
        content_df = pd.DataFrame(results, columns=co
ls)
        content_df.to_csv(f_name, mode=file_mode, hea
der=False,index=False)
        #      pbar.set_description(f"Fetching data fo
r item {row_idx}-{id}")
winsound.Beep(2000, 500)
```

Figure 16: Code snipped for news headings extraction

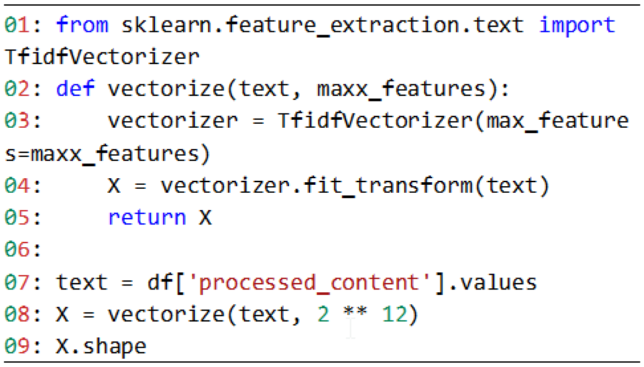


```
try:
        winsound.Beep(500, 100)
        msg = f"Fetching data for page {page} \
             ({total_items} items)"
        pbar.set_description(msg)
        r = fetch_data(page,API_KEY)
        data = json.loads(r.text)
        if not 'posts' in data.keys():
            break
        total_items += len(data['posts'])
        msg = f"Fetching data for page {page} \
             ({total_items} items)"
        pbar.set_description(msg)
        for row in data['posts']:
            for key in list(row.keys()):
                val = row[key]
                if type(val) == dict:
                    for s_key in val.keys():
                        s_val = val[s_key]
                        s_key = key + '_' + s_ke
                        row.update({s_key:s_val}
                    row.pop(key)
            if df is None:
                df = pd.DataFrame(row,index = [0
            else :
                df = df.append(row,ignore_index
    except:
        info = sys.exc_info()[0]
        print(info[0],info[1],info[2])
        winsound.Beep(1500, 100)
    finally:
        f_name=f'{file_name}{start_index}-{total_pages}.csv'
        df.to_csv(f_name)
```

Figure 17: Code snipped for news articles extraction

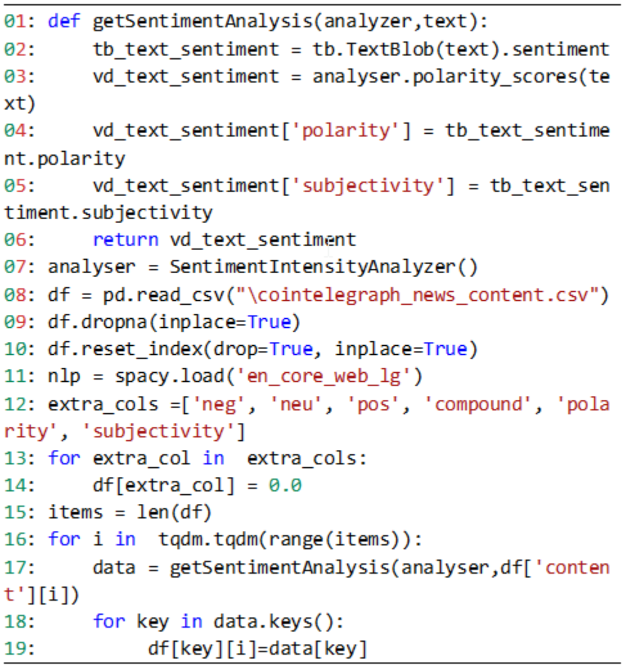
7.0.2 *Stage 2. Newsfeed exploration and classification.* Several attempts to classify the data set and train it to obtain relevant information that lead to decide whether an article is positive, or negative were performed. As stated in section 5.2, different approaches and analysis were tried to obtain a proper classification. Sentiment analysis, clustering, text summarization and deep learning algorithms were applied to the data set, the classification results were neutral and did not provide insightful information for the whole data set to be used and applied as an extra feature for the candlestick data set in the RNN-LTSM algorithm.

The code snippets shown below will demonstrate the use of such algorithms and the way to try to obtain useful information to tag the data set. It was observed, however, that such attempts were not enough and further text analysis was required. This is a limitation will be addressed in the next section.



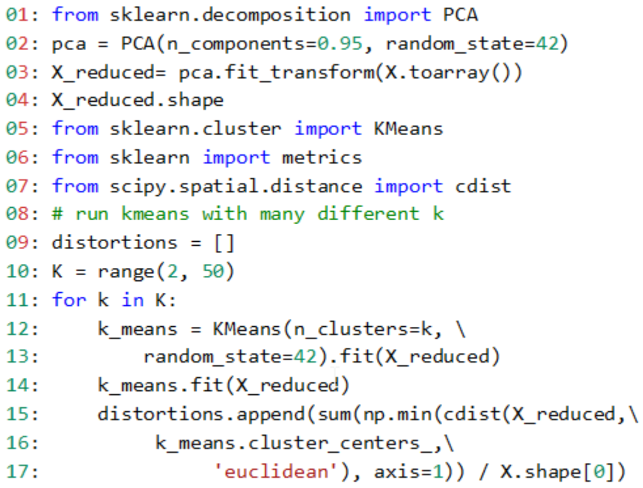
```
from sklearn.feature_extraction.text import TfidfVectorizer
def vectorize(text, maxx_features):
    vectorizer = TfidfVectorizer(max_features=maxx_features)
    X = vectorizer.fit_transform(text)
    return X

text = df['processed_content'].values
X = vectorize(text, 2 ** 12)
X.shape
```

Figure 19: Code snippet for text vectorization



```
def getSentimentAnalysis(analyzer,text):
    tb_text_sentiment = tb.TextBlob(text).sentiment
    vd_text_sentiment = analyser.polarity_scores(text)
    vd_text_sentiment['polarity'] = tb_text_sentiment.polarity
    vd_text_sentiment['subjectivity'] = tb_text_sentiment.subjectivity
    return vd_text_sentiment
analyser = SentimentIntensityAnalyzer()
df = pd.read_csv("\cointelegraph_news_content.csv")
df.dropna(inplace=True)
df.reset_index(drop=True, inplace=True)
nlp = spacy.load('en_core_web_lg')
extra_cols =['neg', 'neu', 'pos', 'compound', 'polarity', 'subjectivity']
for extra_col in  extra_cols:
    df[extra_col] = 0.0
items = len(df)
for i in  tqdm.tqdm(range(items)):
    data = getSentimentAnalysis(analyser,df['content'][i])
    for key in data.keys():
        df[key][i]=data[key]
```

Figure 18: . Code snippet for sentiment analysis attempt



```python
from sklearn.decomposition import PCA
pca = PCA(n_components=0.95, random_state=42)
X_reduced= pca.fit_transform(X.toarray())
X_reduced.shape
from sklearn.cluster import KMeans
from sklearn import metrics
from scipy.spatial.distance import cdist
# run kmeans with many different k
distortions = []
K = range(2, 50)
for k in K:
    k_means = KMeans(n_clusters=k, \
        random_state=42).fit(X_reduced)
    k_means.fit(X_reduced)
    distortions.append(sum(np.min(cdist(X_reduced,\
         k_means.cluster_centers_,\
              'euclidean'), axis=1)) / X.shape[0])
```

Figure 20: Code snippet for KNN clustering classification

8 MODEL TRAINING AND TUNING

Several iterations were required to get proper adjustments to the LSTM model for candlestick data and KNN for the NLP processing. The iterations involved observing the loss function through the deep learning model using TensorBoard and analyzing its precision, overfitting and underfitting on both models.

8.1 Cryptocurrency data set training and tuning

The final adjustments for the training of the neural network can be seen in [12]. The neural network implementation, training and performance for every cryptocurrency trained on the spark environment can be visualized as well in the provided citation. Each candlestick data set is processed in a pandas DataFrame. The loss function and window offset (period in which the neural network predicts the values based on past data) were carefully chosen to approximate the prediction as much as possible for all the sets. Primarily, it was observed that on average most of the models adapt properly at 300 epochs[8] with "Mean Square Error" as loss function and 256 neurons on 3 layers (one hidden). The model can mutate, and it is let to the reader to try to find a better tuning mechanism. It is important to highlight that for the given model the data set was aggregated to present hourly data instead of minute. The reason for this was for the expensive of the process and the amount of resources a training model takes for a single currency. In more powerful environments the model can mutate to observe minute predictions.

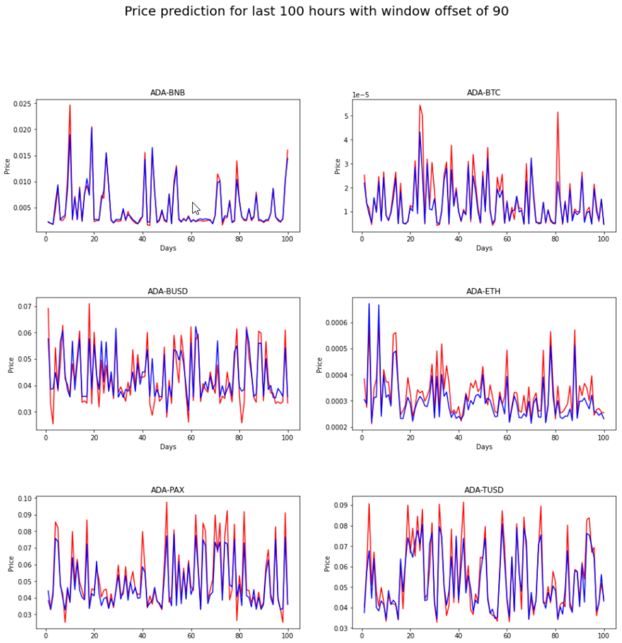
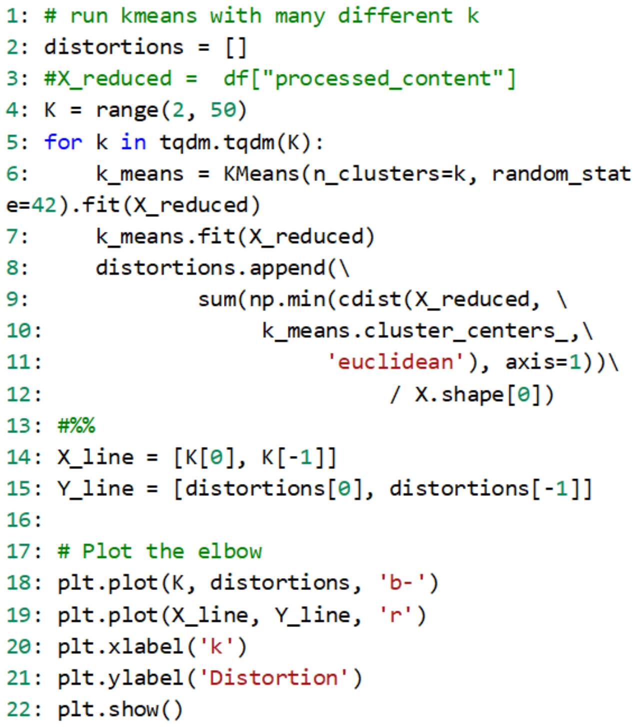


Figure 21: Different predictions after the model has been trained

8.2 Newsfeed data set training and tuning

Trying to tune and adjust the newsfeed data set was even more complicated. The KNN algorithm was used under the premise that as the number of clusters grow for the model to be trained, more differences can be spotted after text processing. The distances between the clusters should increase and get to a state enough to assert and properly distinguish the topics for the articles. On the other hand, a normal sentiment analysis from previously trained models were made just to visualize how much the algorithms could predict the positive or negative impact on specific articles. It was however not possible to segregate the clusters to a distance considerable enough to define topics on the news. This is a limitation addressed in the next section.

[8] The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training data set[7]



```
# run kmeans with many different k
distortions = []
#X_reduced =  df["processed_content"]
K = range(2, 50)
for k in tqdm.tqdm(K):
    k_means = KMeans(n_clusters=k, random_state=42).fit(X_reduced)
    k_means.fit(X_reduced)
    distortions.append(\
            sum(np.min(cdist(X_reduced, \
                 k_means.cluster_centers_,\
                      'euclidean'), axis=1))\
                           / X.shape[0])
#%%
X_line = [K[0], K[-1]]
Y_line = [distortions[0], distortions[-1]]

# Plot the elbow
plt.plot(K, distortions, 'b-')
plt.plot(X_line, Y_line, 'r')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.show()
```

Figure 22: Code snippet to determine the best number of clusters

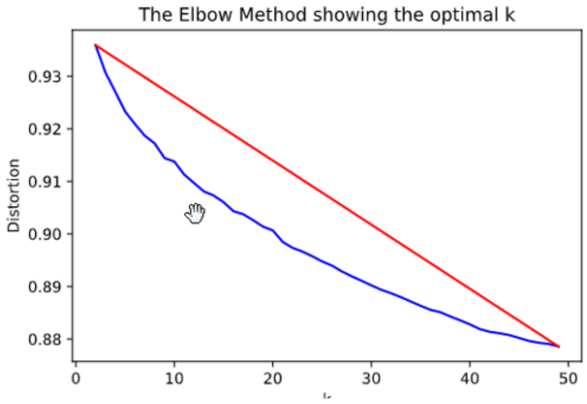


Figure 23: Results plot to determine proper number of clusters

After executing the code from 23, the results to determine a suitable distance between clusters was performed. It was not possible to find a proper distinction between the news, the best fit for KNN cluster classification was 2, which is not suitable for the problem to solve. As the number of clusters increase, it is evident the classification algorithm is not able to distribute properly information among different groups, and therefore different approaches must be performed.

9 LIMITATIONS

Several limitations were encountered during the project. In some cases, the limitations could be partially avoided by modifying our use case, but in other cases intended implementations had to be dropped and noted as possible future work.

9.1 Environment resources

One issue was regarding storage space. The three Virtual machines configured as slave nodes, which would be the machines to hold the data, each had 40 GB of storage. In addition to needing some storage for the operating system and applications needed to run the cluster, the Hadoop Distributed File System (HDFS) would store data across all nodes with redundancy, giving us a limited amount of storage. The Binance data set consisted of 9.36 GB of data in parquet files, and in order to convert this data into a plain text format for use in Machine Learning, it would require up to 72GB on each data node.

This limitation was addressed by performing a parquet file decompression mechanism from Spark, work with the model in memory and saving the trained model once the process was finished. This was a resource intensive mechanism and took more than 6 days to train the whole data set.

9.2 Data set availability

No cryptocurrency-related newsfeed data set could be found for use in Machine Learning. Normally to process NLP corpora and perform sentiment analysis over text is now available through different libraries, however it was observed that no pre-trained model nor available data set open to the public has been yet generated. The solution was to create Python scripts for collecting news data (explained in a previous section) and try to exploit the obtained data to obtain relevant results that allowed our data set to be classified. Because the data was unlabeled, using it to train a Machine Learning model would require much more analysis and processing in order to classify the data. Accurate results would require to manually read and label more than 50% of the news catalog (an equivalent of 15,000 different news) and afterwards used them to train a model and test it over the unlabeled data. This task was so much time consuming and neither of the NLP tools employed to explore the data set provided relevant information that allowed the process to be automated or slightly simplified.

9.3 Spark state of the art

The last big limitation encountered, was regarding the execution of Deep Learning algorithms in a distributed fashion. Although already existent SparkML for distributed Machine Learning training on clusters, there is no currently an algorithm available for deep learning process, something necessary required for the provided dataset. Fr this reason TensorFlow must be configured in the cluster to work with Spark, which also created a challenge with regards to distributing the resources when training Machine Learning models. Nevertheless, before implementing Tensorflow natively in python in the cluster an exhaustive research was made to try to address such limitation. It turns out there are already attempts to make deep learning-TensorFlow algorithms to work in a distributed manner under Spark clusters, SparkFlow[1] is a library that tries to implement deep learning on Spark, however, after several attempts to make it work and configure with the TensorFlow algorithms used for the project, several incompatibility issued arose. The final solution therefore was to keep the non-distributed pySpark-TensorFlow approach.

10 FUTURE WORK

Possible future work includes classifying the collected news data, addressing the limitation discussed in section 8.2. This would allow the training of a Machine Learning model able to perform sentiment analysis on extracted news, labeling data as positive or negative with regards to cryptocurrency market impact.

Combining historical price data and labeled news data, a more accurate model for predicting cryptocurrency prices could be realized. Implementing automatic collection of price and news data on the Hadoop cluster, with the help of real-time sentiment analysis, would allow for a continuously improving prediction model.

Possible further improvements could be made by also considering factors such as technological improvements and social adoption.

11 CONCLUSIONS

The idea of combining social media information with cryptocurrency trading remains unexplored for the lack of information available. This presents a potential to mature the project as a product with models to be trained on daily basis once new data is released. Although social media keeps nowadays a very important role for cryptocurrency trading decisions, there is no open source dataset nor code available to explore. The code developed for this project is now available on GitHub [8] and Kaggle [7]. It is therefore led for future work to exploit the findings on this project and tune the newsfeed dataset to properly classify it. Social media data such as Twitter would be interesting to be added to find the correlations and perform a Sentiment analysis for the trading decisions on daily basis. There would be, however, an inherent need to previously analyze such dataset. It is important to mention that beforehand the intention was to use any other social media dataset, no results were found, therefore the need to create one by itself. Also the state of the art for distributed systems is not mature enough to address this kind of problems since deep learning cannot be run on clusters yet for parallel computations and calculations, will be very interesting to retake the project and explore future functionalities once this features are released in future versions.

REFERENCES

[1] [n. d.]. lifeomic/sparkflow. https://github.com/lifeomic/sparkflow original-date: 2018-04-07T14:43:16Z.

[2] [n. d.]. pandas: powerful Python data analysis toolkit. https://github.com/pandas-dev/pandas

[3] [n. d.]. What are cryptoassets (cryptocurrencies)? http://www.bankofengland.co.uk/knowledgebank/what-are-cryptocurrencies Library Catalog: www.bankofengland.co.uk.

[4] Roshan Adusumilli. [n. d.]. Predicting Stock Prices Using a Keras LSTM Model. https://towardsdatascience.com/predicting-stock-prices-using-a-keras-lstm-model-4225457f0233 Library Catalog: towardsdatascience.com.

[5] Ameer Rosic. [n. d.]. What is Blockchain Technology? A Step-by-Step Guide For Beginners. https://blockgeeks.com/guides/what-is-blockchain-technology/ Library Catalog: blockgeeks.com Section: Blockchain 101.

[6] Binance. [n. d.]. Bitcoin Exchange | Cryptocurrency Exchange. https://www.binance.com Library Catalog: www.binance.com.

[7] Jason Brownlee. [n. d.]. When to Use MLP, CNN, and RNN Neural Networks. https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/ Library Catalog: machinelearningmastery.com.

[8] Asahi Cantu. [n. d.]. Cryptocurency CoinTelegraph Newsfeed. https://kaggle.com/asahicantu/cryptocurency-cointelegraph-newsfeed Library Catalog: www.kaggle.com.

[9] Kai Chen, Yi Zhou, and Fangyan Dai. [n. d.]. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE International Conference on Big Data (Big Data)* (2015-10). 2823–2824. https://doi.org/10.1109/BigData.2015.7364089

[10] Christopher Olah. [n. d.]. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11] CoinMarketCap. [n. d.]. Bitcoin price, charts, market cap, and other metrics. https://coinmarketcap.com/currencies/bitcoin/ Library Catalog: coinmarketcap.com.

[12] CoinMarketCap. [n. d.]. Cryptocurrency Market Capitalizations | CoinMarketCap. https://coinmarketcap.com/

[13] CoinTelegraph. [n. d.]. Cointelegraph Bitcoin & Ethereum Blockchain News. https://cointelegraph.com/

[14] Aditi Mittal. [n. d.]. Understanding RNN and LSTM. https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e Library Catalog: towardsdatascience.com.

[15] Satoshi Nakamoto. [n. d.]. Bitcoin: A Peer-to-Peer Electronic Cash System. ([n. d.]), 9.

[16] Pablo Castilla. [n. d.]. Predict stock prices with LSTM. https://kaggle.com/pablocastilla/predict-stock-prices-with-lstm Library Catalog: www.kaggle.com.

[17] Abhinav Sagar. [n. d.]. Cryptocurrency Price Prediction Using Deep Learning. https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep-learning-70cfca50dd3a Library Catalog: towardsdatascience.com.

[18] Jorijn Jacko Smit. [n. d.]. Binance Full History. https://kaggle.com/jorijnsmit/binance-full-history Library Catalog: www.kaggle.com.

[19] Alexandre Xavier. [n. d.]. Predicting stock prices with LSTM. https://medium.com/neuronio/predicting-stock-prices-with-lstm-349f5a0974d4 Library Catalog: medium.com.

[20] Yacoub Ahmed. [n. d.]. Predicting stock prices using deep learning - Towards Data Science. https://towardsdatascience.com/getting-rich-quick-with-machine-learning-and-stock-market-predictions-696802da94fe