Deep learning models as advisors to execute trades on financial markets

CORENTIN ABGRALL
Deep learning models as advisors to execute trades on financial markets

CORENTIN ABGRALL
Abstract

Recent work has shown that convolutional networks can successfully handle time series as input in various different problems. This thesis embraces this observation and introduces a new method combining machine learning techniques in order to create profitable trading strategies. The method addresses a binary classification problem: given a specific time, access to prices before this moment and an exit policy, the goal is to forecast the next price movement. The classification method is based on convolutional networks combining two major improvements: a special form of bagging and a weight propagation, to enhance the accuracy and reduce the overall variance of the model. The rolling learning and the convolutional layers are able to exploit the time dependency to strongly improve the trading strategy. The presented architecture is able to surpass the expert traders.
Sammanfattning

# Contents

1 Introduction 1  
  1.1 Problem statement ................................................. 2  
  1.2 Scope .................................................................. 4  
  1.3 Outline .................................................................. 4  

2 Background 5  
  2.1 Finance basics .......................................................... 5  
    2.1.1 The market ....................................................... 5  
    2.1.2 The order book .................................................. 5  
    2.1.3 Market and limit orders ...................................... 7  
    2.1.4 Candlestick chart .............................................. 8  
  2.2 Strategy ................................................................. 9  
    2.2.1 Long and short orders ....................................... 9  
    2.2.2 Entry and Exit policies ...................................... 10  
    2.2.3 Finance metrics ................................................. 10  
    2.2.4 Backtest .......................................................... 12  
    2.2.5 Setting the strategy ........................................... 13  
    2.2.6 Rolling Learning ................................................. 14  
  2.3 Classical Flaws ....................................................... 14  
    2.3.1 General Trend ................................................... 15  
    2.3.2 Trading fees ..................................................... 15  
    2.3.3 Market Impact .................................................. 16  
    2.3.4 Slippage .......................................................... 17  
    2.3.5 Spread ............................................................ 18  
    2.3.6 Take profit ......................................................... 19  
  2.4 Related works ......................................................... 19  
    2.4.1 Machine learning and financial market ................. 19  
    2.4.2 Deep learning in finance .................................... 21  
    2.4.3 Convolutional neural networks ............................ 21
# CONTENTS

2.4.4 Limitations ........................................ 22

3 Method ................................................. 23

3.1 Binary classification using financial indicators ................. 23
  3.1.1 Input data ....................................... 23
  3.1.2 Label ........................................... 24
  3.1.3 Evaluation ....................................... 24
  3.1.4 Network’s Architectures ........................... 25
  3.1.5 Sparse connected layer .............................. 25
  3.1.6 Special Bagging ................................... 26
  3.1.7 Weight propagation ................................ 27

3.2 Binary classification using the raw data ......................... 27
  3.2.1 Input data ....................................... 27
  3.2.2 Network’s architectures ............................ 28
  3.2.3 Representation of the data ........................... 28
  3.2.4 Regularization .................................... 29

4 Results .................................................. 30

4.1 Binary classification with technical indicators ................... 30
  4.1.1 Classification task ................................ 30

4.2 Binary classification with raw data ............................ 32
  4.2.1 Data representation ................................ 32
  4.2.2 Input size ........................................ 33
  4.2.3 Regularisation ..................................... 34
  4.2.4 Classification task ................................ 35

4.3 Statistical analysis of the results ............................... 36

4.4 Post hoc analysis: Honest Significant Difference ............... 38

4.5 Trading strategy ....................................... 39
  4.5.1 Financial Metrics .................................. 40
  4.5.2 Equity curves ..................................... 40

5 Discussion .............................................. 42

5.1 Implications .......................................... 42

5.2 The deep learning classifier ................................ 43

5.3 Evaluation ............................................ 43

5.4 Future work ........................................... 44
  5.4.1 Interpretability of the forecast ....................... 44
  5.4.2 Integration of other type of information ............. 44
  5.4.3 Expanding the access on data ....................... 44

5.5 Ethics and sustainability .................................. 45
6 Conclusion 46
Bibliography 47
Chapter 1

Introduction

The degree project involves two different topics: financial market and machine learning. On financial market, traders have always been interested into new ways of predicting the evolution of the market prices. To achieve this, they use different tools such as news, market history, temporal signals (price, number of transactions, amount of transactions), statistic analysis, etc. In this context, machine learning methods have shown interesting results regarding direct prediction from technical indicators [1]–[4]. The methods presented aim to forecast the evolution of the stocks on a few days horizon and are based on neural networks or k-nearest neighbours. Another strong approach comes from data mining and uses automatic extraction of information from financial news. The methods involved mainly come from natural language processing with the frequency approach of the TF-IDF [5] and the vectorization of words [6] but not only with the naive Bayesian Classifier [7]. However, internet habits also bring new strategies such as sentiment analysis on Twitter [8], [9] and on Google queries [10]. However, the trading strategies presented in the previously quoted articles claim to be profitable but the reality is more complicated. Indeed, there is no reference data, the authors must collect their own. The profitability of the strategy is also presented on historical data, which requires to be very cautious regarding possible flaws of modelling. There is no mention of these possible issues introduced in work to date. This work follows a technical approach since the input data is based on technical indicators. The presented method aims to extract information from the data in order to forecast the evolution of the prices. A particular attention is given to the quality of the tests on
1.1 Problem statement

From different signals obtained from the stock market evolution, experienced traders define interesting moments to invest and pass orders. They define rules to characterise those moments and can be implemented in filters. The filters will scan the stock market evolution and automatically identify moments considered as favorable to pass orders by the traders. This is where machine learning has potential to contribute to trading.

A filter is based on several rules (also called conditions) \( (\text{cond}_1, \text{cond}_2, \ldots, \text{cond}_c) \). The conditions often rely on several indicators and are computed at a specific moment. To train the model, the filter on the historical data outputs a series of moment where the traders should have bought, denoted as \( T = (t_0, t_1, \ldots, t_M) \).

However, making decision about buying a share based on only few conditions is not really reliable. It does not use enough indicators. Indeed, using more indicators might help us to be more accurate. A first possible improvement is to add, after this filtering part, a binary classifier based on machine learning which is able to automatically check more indicators. The classifier takes as input a vector \( X^{(k)} \) of length \( N \) such that: \( X^{(k)}_i = \text{ind}_i(t_k) \) where \( i \in [0, N] \) and \( k \in [0, M] \). Here \( \text{ind}_i(t_k) \) corresponds to the i-th indicator computed at the time \( t_k \). An indicator is a scalar value. The output of the classifier is binary: an output equals to 0 for \( X^{(k)} \) means that even if the filter considers that \( t_k \) is a good moment to buy or sell, the classifier rejects this opinion. Since the decision of the classifier is based on more information, the trading model will wait and neither buy nor sell at \( t_k \). On the contrary, if the classifier outputs 1, the filter was right and the trading model will act.

For now, the company has found, at least, one filter, denoted \( F \), which is statistically profitable. This filter has been designed by expert traders. In reality, the filter does not take only two conditions but still, their quantity of indicators is still low and it is very time consuming to craft. Therefore, one way for the company to improve the
trading model is to add a classifier, as described in the previous paragraph. Indeed, this method is supported by internal studies showing that a hand crafted classifier outperforms a trading model only based on the filter. This study also shows that the time required to design such classifier by humans is enormous and might be reduced by using machine learning methods. For now, the company has also designed one classifier based on decision trees which has improved the overall performances of the trading model.

To sum up, the company has, for now, three trading models:

- The first one is only composed of the filter $F$. This model will be later refereed as $C_{ref}$.
- The second one is composed of the filter $F$ and a hand crafted classifier added after the filter $C_{hand}$.
- The third one is composed of $F$ and a classifier based on decision trees $C_{tree}$.

Regarding financial metrics the trading models can be ranked as follow (higher the better):

$$C_{ref} < C_{tree} < C_{hand}$$

The company is still working on the trading model $C_{tree}$ trying to improve it to outperform $C_{hand}$. From this work, the company has noticed that increasing the number of indicators used in the classifier tends to improve the performance. However, $C_{tree}$ does not seems to properly handle a large number of indicators for now. The company is therefore also looking for another method of machine learning for the classifier. Among machine learning methods and models, deep learning methods are known for their capacity in discovering complex structure of information in large datasets [11]. The recent breakthroughs in speech recognition, computer vision and natural language processing are made by deep learning methods. This ability to extract intricate structure of information from large dataset is exploited in the finance sector [12], [13]. The architecture used for obtaining this results are Convolutional Neural Networks (CNN). Therefore, such methods could be used to create a fourth trading model $C_{deep}$ using the filter $F$ and a deep learning classifier behind. The degree project focuses on
finding the most adapted classifier based on deep learning methods.

This problem uses a specific approach based on a pre-filtering of trade opportunities. Currently, the best prediction accuracies regarding financial metrics are reached by human experts with their hand crafted models $C_{\text{ref}}$ and $C_{\text{tree}}$. However, the time required to achieve this is large and can be improved by machine learning methods. This work aims to examine the performance of the newly proposed $C_{\text{deep}}$ model in relation to well established trading strategies executed by human expert ($C_{\text{hand}}$) and a reference model ($C_{\text{ref}}$) currently in use. We will investigate whether the introduction of convolutional neural networks to the new model results in better prediction performance and discuss its key properties in this regard.

1.2 Scope

This degree project focuses on the possible architectures and deep convolutional neural networks to improve the trading strategy. The study of the other models ($C_{\text{ref}}$, $C_{\text{hand}}$, $C_{\text{tree}}$) falls beyond the scope of this thesis. The project does not focuses on the interpretability of the decisions made by the classifier $C_{\text{deep}}$. The focus is made on the possible architectures and training methods used to improve the overall accuracy of the trading model.

1.3 Outline

The rest of the report continues as follow: the chapter 2 gives the background needed in finance and presents the related work. The 3rd chapter explains the method used for testing the hypothesis in this thesis. The results are presented in chapter 4. The discussion and the analysis of the results are in chapter 5. Finally, the chapter 6 concludes the thesis by recapitulating the results and the implications involved.
Chapter 2

Background

2.1 Finance basics

2.1.1 The market

The market, also called the exchange, is a place made for matching buyers and sellers of financial products. The products are various (bonds, shares, commodities, currencies, etc.) and depends on the market place. Some places are specifically dedicated to one kind of financial products which gives them some specifies. Some markets are very liquid, the opening hours are different, the fees for trading changes making them more attractive for different kind of trading strategy. For instance, high frequency trading is less interesting on a market with high fees and few liquidities.

The price of an asset is the value used in the last transaction executed. On a specific market, the prices are discrete. The smallest difference between two prices is set and is called the tick.

Markets are allowed to interact with very few companies called the brokers. They are the intermediate between the traders and the exchange. Brokers take commission fees to execute the orders.

2.1.2 The order book

The order book is the list of orders from the interested buyers and interested sellers. Since the prices are discrete, it is possible to represent this book with an array. The prices, in the middle column, are sorted. For a given price, the order book aggregates all the interested partici-
pants and sum their shares. This number is on the left (resp. right) if the participants want to sell (buy). An order book is presented in table 2.1.

<table>
<thead>
<tr>
<th>Share(s)</th>
<th>Price</th>
<th>Share(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>-</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.1: The order book. It shows 3 available shares for buying at the price of 11. It is also possible to sell at most 5 shares at 9.

If one seller and one buyer agree on the same price for shares, the transaction will occur and shares will be removed from the order book. That is why it is not possible to have shares on both side at the same time.

In the order book presented in table 2.2, two prices are relevant:

- The smallest price for which a participant is ready to buy is called the ask.
- The biggest price for which a participant is ready to sell is called the bid.

The difference between the ask and the bid is called the spread. The price (sometimes called last price) is the price of the last transaction executed, and can be equal to the current ask or the current bid.

<table>
<thead>
<tr>
<th>Share(s)</th>
<th>Price</th>
<th>Share(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>Ask</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>Bid</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.2: A complete order book. The ask is at 11 and the bid is at 9, thus the spread is 2.
The data available in finance problems is very often limited to the last prices. Having the complete order book brings more information but is harder to collect or more expensive to buy.

2.1.3 Market and limit orders

It is possible to place various kinds of orders on a market place. Only two of them require some attention regarding the scope of this document:

- The market orders
- The limit orders

**Market orders** The market orders are the simplest ones. A market order is executed immediately and at the best price possible. An example of a market order in presented in figure 2.1.

<table>
<thead>
<tr>
<th>Share(s)</th>
<th>Price</th>
<th>Share(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Bid</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

(a) Before the execution

<table>
<thead>
<tr>
<th>Share(s)</th>
<th>Price</th>
<th>Share(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Ask</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Bid</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

(b) After the execution

**Figure 2.1:** Evolution of the order book during the execution of a market order. The market order is executed with 2 shares. The price of the execution is the ask (7).

**Limit orders** A limit order is an order placed at a specified price for which the trader agrees to make the transaction. This order might not be executed since the price might not reach the threshold. The table 2.3 shows the evolution of the order book during while a limit order is placed.
### Candlestick chart

Candlestick charts are a visual way to represent the data. One candlestick represents the prices for a fixed period of time $T$ called the time frame. During this time frame, 4 prices are relevant to build a candlestick:

- The opening price at the beginning of the time frame: $\text{open}$.
- The closing price at the end: $\text{close}$.
- The highest price during this period of time: $\text{high}$.
- The lowest price during this period of time: $\text{low}$.

Those numbers can be plotted under the shape of a candlestick:

- Fig. 2.2: A candlestick with a bullish trend.

The color of a candlestick is given by the sign of the $\text{Close} - \text{Open}$ and shows if the market is bullish or bearish.
The last relevant number is the volume (number of shares exchanged during $T$). It is often plotted with a histogram under the candlesticks.

![Candlestick chart](image.png)

**Figure 2.3:** A candlestick chart (SPY on 26/09/2017, 1 min per candle). The volumes are in grey at the bottom of the figure.

## 2.2 Strategy

### 2.2.1 Long and short orders

**Long order**  The investor buys a share with the expectation of a rise of the price and he owns the share. It is not necessarily a share, it can be a stock, a commodity, etc. The goal is to sell it later at an higher price. If so, the return is positive and the investment is profitable.

**Short order**  In this context, the investor expects the price to shrink. Therefore, the investor borrows a stock from a fund and sells it. Later, the investor buys a share and gives it back to the fund. If the price has decreased, the investment was profitable for the investor.

A trading strategy can combine a routine for each type of positions (long or short) but in the scope of the master thesis, the trading model
used is long. Therefore, it can only loose if the market is bearish.

### 2.2.2 Entry and Exit policies

To automate the trades, it is possible to establish an entry policy and an exit policy. Since, the trading strategy uses only long positions (cf. [2.2.1](#2.2.1)), the entry policy answers the question: when should a stock be bought and the exit policy: when should the stock be sold? A simple entry policy can be: *I buy if and only if the price is lower than x*. Therefore, an automated trading model could be the association of the entry and an exit policy. The thesis focuses on the entry policies. During all the experiments, the same exit policy is used and studying it falls beyond the scope of the thesis.

The exit policy is composed of 3 elements:

- The take profit (TP): it is a limit order executed at a fixed price which is considered to be profitable for the company.
- The stop loss (SL): it is market order executed to limit the loss when the prediction turns out to be wrong.
- The end-of-trade: it is a time limit. If the price has not reached any of the previous thresholds within a certain time, the prediction made is not consistent anymore so one needs to exit the position.

### 2.2.3 Finance metrics

Once the a model is created, it needs to be evaluated. The metrics mainly used are the return, the sharp ratio, the sortino ratio and the equity curves.

**Return**

The return is the benefit (possibly negative) of an investment. Therefore the basic computation is to make the difference between the selling price and the buying price, called the nominal return:

\[
R_{\text{nominal}} = p_{\text{sold}} - p_{\text{bought}}
\]  

The return can also be the return on investment. The benefit are divided by the money allocated to the investment.
Figure 2.4: The market with an exit policy: TP (blue), SL (red), end-of-trade (black), entry moment (grey arrow). TP is set at 12968, SL at 12954 and the end-of-trade at 14:03.

\[ R_{\text{percentage}} = \frac{P_{\text{sold}} - P_{\text{bought}}}{P_{\text{bought}}} \]  

(2.2)

**Sharp ratio**

The Sharp ratio is a financial metric which measures the profitability of a portfolio. This metric can be computed to evaluate a set of trades. \( m_r \) represents the mean of the trades' returns. \( m_a \) is the average rate of return. It represents the risk free strategy: buying at the same time as the oldest trade and selling at the same time as the latest trade. \( \sigma \) is the standard deviation of the returns.

\[ R_{\text{sharp}} = \frac{m_r - m_a}{\sigma} \]  

(2.3)
For this metric, the higher is the better. Indeed, a high sharp ratio can mean several things:

- A high numerator: the mean of the returns is higher than the average rate of return. In this context, the strategy is winning.

- A low denominator: a low standard deviation implies a low risk exposition. If most of the trades are close to the mean, the likelihood to lose the all investment is lower.

Ideally, the trading strategy has a low risk (low denominator) and also a high reward (high numerator), therefore the sharp ratio is high.

**Sortino ratio**

The Sortino ratio is close to the Sharp ratio and can be computed with the same input data. The difference relies in the denominator. Only the trades with a negative return are considered to compute \( \sigma^- \) the standard deviation which is often called the downside deviation.

\[
R_{\text{sortino}} = \frac{m_r - m_a}{\sigma^-} \tag{2.4}
\]

The Sortino ratio does not take into account the volatility of the positive returns. The Sharp ratio punishes the risk even for positive return when the benefits are important. The Sortino ratio compensates this drawback by excluding positive trades from the computation of the downside deviation.

**Equity curve**

The equity curve plots the amount of available money as a function of the time. The quantity decreases when an asset is bought and increases after the asset’s sell. An example is provided in table 2.4.

**2.2.4 Backtest**

Once the trading strategy is designed, the performances and the risks can be evaluated by applying this strategy on the historical data. This is called a backtest. If the backtest returns good financial metrics, this strategy can be implemented and if not the strategy can be discarded or improved to reach better performances. The accuracy of a backtest
Table 2.4: The evolution of the equity curve. At $t$, the available amount of money is 100. At $t + 1$, a contract evaluated at 20 is bought and then sold at $t + 2$ for 30. Thus, the final amount of available money is 110.

<table>
<thead>
<tr>
<th>Time</th>
<th>Equity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>100</td>
</tr>
<tr>
<td>$t + 1$</td>
<td>80</td>
</tr>
<tr>
<td>$t + 2$</td>
<td>110</td>
</tr>
</tbody>
</table>

is crucial because it is a verification step. The historical data should include parts with strong trend (bullish and bearish) and also flat periods. The results should also be read very carefully, a strategy can have great returns but a very low number of trades, meaning that it may not be significant. A backtest must stick to the reality as much as possible.

### 2.2.5 Setting the strategy

The strategy is a set of different elements:

- The entry policy
- The exit policy
- The number of assets

The thesis focuses only on the entry policy. The exit policy will always remain the same. The number of assets is the amount of money allocated to the strategy. It is usual to have several strategies working at the same time, since the total amount of money available for the trader is limited, one needs to properly handle the distribution to maximise the profit. This subject also falls beyond the scope of this thesis.

One major hypothesis is made during the experiments. The entry policies do not depend on the exit policies. This assumption is made for two main reasons:

- The complexity to find strategies is reduced and allows often to find some interesting models which can later be updated or modified without this hypothesis.

- Company’s work shows that this assumption is not inconsistent and proves, under some constrains, that this is approximately true.
2.2.6 Rolling Learning

The rolling learning is a technique used with time-series mainly to avoid overfitting. Since the data is a time-serie, one can defined a window to roll over the whole dataset with a fixed step size. Each window is considered as a train dataset and the next step is the test dataset. Figure 2.5 presents an example.

![Figure 2.5: Rolling Learning. Data starts in 1999 and ends in 2008, with a period of 3 years and a step size of one year, the whole dataset can be splitted into 7 sub-dataset.](image)

This method requires two parameters:

- The period : size of the time frame (size of the training set)

- The step size : size between the current window and the next one (size of the testing set)

It can be seen as a form of cross validation using the time dependencies in the data. The machine learning experiments conducted uses this method for training. The rolling learning is also particularly adapted in this context since the markets are non stationary.

2.3 Classical Flaws

A numerous number of scientific articles in finance are confronted to difficulties concerning the ability to properly test their trading strategy on historical data. To achieve this, one must understand how a market works and the implicit challenges related to limit the impact of classical flaws. This section presents the main and most recurrent carelessness and how to deal with them.
2.3.1 General Trend

The general trend is characterized by the return of the index over a large period of time. One can consider a given instrument where one contract cost 100 at a specific time. If one waits 10 years and sells this contract at 250, the general trend is bullish since the return is positive (150).

This property of the market is extremely important to take into account when trading strategies are tested. If the backtest of a strategy gives an overall return inferior to the general trend, it means that the strategy is worthless. Indeed, one would have beat this strategy by buying at random at the beginning of the studied period and sold it random at the end.

![Figure 2.6: Evolution of the DAX (1983-2017). Red and green colors indicate the trend of one candle (cf. 2.1.4).](image)

The general trend is often taken into account with the metrics used: the Sharp ratio or the Sortino ratio are using the mean return of an index over the concerned period of time.

2.3.2 Trading fees

Between the exchange and the trader, there is an intermediate: the broker. The broker arranges the transaction between the buyer and the seller for a commission. This intermediate is mandatory since being directly in contact with the exchange is reserved to a very few com-
panies which must respect constrains because of their status. Therefore a strategy must take these fees into account. Usually, the fees are a small amount of money asked for each contract executed. A strategy making thousand of trades every year must count the fees.

The trading fees depend on the exchange place and on the broker. Therefore, the amount of fees are not hard to obtain and are easy to integrate into the trading simulation.

### 2.3.3 Market Impact

The market impact is an effect induced by a market participant selling or buying assets. The price increases in case of buying. This impact is important when the amount of assets involved is important, which might be the case for the biggest financial companies. From the point of view of the order book, this means that the ask is going to move up. The executed prices for each of the contracts will therefore not be the same.

When the volume involved in the trading strategy is important, the market impact can drastically increase the cost to enter on the market and must be evaluated before.

![Order Book Example](image)

**Figure 2.7:** The order book during the entry of a market participant.

As an example, if a market participant decides to buy 20 shares
with the order book in figure 2.7 he will pay:

\[ 3 \times 7 + 8 \times 5 + 10 \times 9 + 2 \times 11 = 173 \quad \text{instead of} \quad 7 \times 20 = 140 \]

The difference between the prices is due to the market impact. The market impact is an effect correlated with the volume involved. If the volume of the assets is small, the market impact is negligible.

### 2.3.4 Slippage

The slippage is the gap between the price executed and the expected price of execution. This gap might be due to several causes, it is mainly latency or related to the SL.

**Latency**  A important delay between the trader and the broker can create a latency slippage. Indeed, if the strategy is using a small time frame, the delay must be reduced as much as possible. A delay of 2 seconds will give at least 4 seconds (forth and back) for the price to move against your prediction. This gap could change the evaluation of the risk and change the decision (buy, sell or wait). This is true for your entry policy but also for the exit policy.

In the company, the time frame is not constant but, depending on the experiments and the context, is always over one minute, therefore a delay of a few seconds is negligible.

**Stop Loss**  This explanation requires to understand how an order book works (cf. 2.1.2) and the different kind of existing orders (cf. 2.1.3). The SL is a market order used for limited the loss if the prediction proves to be wrong.

If a market order with a large volume is executed, the ask or bid (depending if we buy or sell) will significantly move. This drift creates a gap between the previous value and the current one. Therefore the other market orders which had a really close expected price will be executed at the new price. Thus, this gap will affect their returns. An example is provided in figure 2.8.

This flaw is much harder to counter. It can not be taken into account into a simulation, the best way to handle is fake online trading. Often brokers can offer a trading live simulation. They receive the decisions
(a) Order book before the slippage. The market is expected to be bullish. Thus, one bought 2 contracts at 9 with TP at 13 and SL at 7.

(b) Order book after another participant decided to sell 40 contracts with a market order.

Figure 2.8: Slippage with SL. The market impact of the other participant makes the bid shrink. Since the market order is executed directly, it is executed before the limit order set at 7. After the execution of the market order, the new price for selling is 4. Therefore, this is an unexpected loss. The gap is the stop loss slippage.

(send by the algorithm and pretend to execute them on the market. In this context, the broker can have the real price.

2.3.5 Spread

As previously defined (cf. 2.1.2), the spread is the gap between the ask and the bid. The financial data available does not always contain the order book but only the last prices. Only having the prices does not tell if the price corresponds to the bid or the ask. The gap between the bid and the ask, the spread, might be important. Therefore, expecting to buy some assets at the last price (which is not the ask) will create a potential difference (the spread times the number of contracts) in the cost.

Another drawback occurs when backtests are done using only the prices. If the SL (resp. TP) are between the ask and the bid and the price is at the ask (resp. bid) the backtest will, in both situations, con-
tinue the trade whereas it should have executed the exit policy.

The best way to handle this flaw is to have the complete data (i.e. the full order book). However, this is very expensive.

2.3.6 Take profit

The last price can reach a certain price but all the contract at this level might not be executed because there are too many of them. Therefore, the executed contracts will be the oldest ones. When all contracts at this price are executed the price will either decrease or increase (if one sells or buys). But the price can also go the other way (if nobody wants to buy/sell anymore at this price). Very often the simulations do not take this into account, which is very important since one might think that the trade is over whereas it is not. The price can then reach the SL. This flaw is classic when the data available is limited to the last prices. During a backtest, two approaches can be considered to overcome this issue: optimistic or pessimistic. In the first one, one considers that our contracts are the first to be executed. On the contrary, the latter will consider our contracts are the last ones.

2.4 Related works

2.4.1 Machine learning and financial market

Various machine learning methods have been created and investigated to forecast stock price evolution. Several main trends have emerged from this large amount of research. The differences are mainly based on the source of the data used.

A common approach uses technical indicators. There are used as inputs for machine learning methods such as neural networks. Technical indicators are very often used by experimented traders, they can characterise different properties of the prices (trend, volatility, oscillation, etc.) at a specific moment. Guo & al. have conducted a study with several indicators as input of a feed-forward neural networks. The goal was to address a classification of 18 well-known finance patterns extracted from the stock exchange of Shangai SE and Shenzen SE.
Leight & al. [2] also used neural networks but combined with genetic algorithms for fine-tuning the hyper-parameters. Their goal was to predict the evolution of the stock’s price on the 20 days horizon. This combination of neural networks and genetic algorithms is also used by Kwon & al. [3] to forecast the short-term variations of the stock prices. The features are a set of 75 technical indicators. The experiments are made on 36 stocks of the NYSE and the NASDAQ (2002-2004). The training is done using the rolling learning (cf. 2.2.6) with a sliding window of 2 years and a step size of 1 year. Teixeira & al. [4] used only 22 indicators but the machine learning algorithm used is the k-Nearest Neighbours. The prediction is made on several stocks of Sao Paulo SE.

The second main approach is the automatic extraction of information from financial articles or short news concerning financial markets, companies or geopolitics. The study [7] used Naive Bayesian classifiers to construct a day-trading strategy. The news talks about over 120 different stocks published over 4 months. The main implication of these articles is the ability of the models to successfully predict the evolution of the stock prices 20 minutes after getting the news. Later, MitterMayer & al. [5] used natural language processing tools on the news: first tokenization and word stemming to simplify the text. The representation of the word into a vector uses the frequency method TF-IDF which is then the input of an SVM classifier. It can emits 3 different opinions: Good, bad or nothing on the evolution of a stock. The prediction of the stock movements is also used by [6] but the vectorization is done using Bag-of-words and Noun Phrases.

Internet habits have given another option exploited more recently: the Twitter feeds or Google queries may reflect the opinion and the changes occurring of the stock prices. The Google queries are used by Reis & al. [10]. The method studies the dependency between the volume of queries concerning a specific stock or bound and the price of this particular stock. Zhang & al. [8] have conducted a study showing a correlation between the frequency of specific keyword messages on Twitter and the evolution of the Dow Jones, NASDAQ and S&P500. The scope of the prediction holds for the several next days. Finally, Bollen & al. [9] used sentiment analysis tools on Twitter’s messages.
combined with a self-organizing fuzzy neural network to forecast the Dow Jones.

This thesis falls in the first approach using the technical indicators as input of the machine learning method. However, the method used in this thesis remains uncommon thanks to the filter. The deep learning classifier is used on pre-filtered data.

2.4.2 Deep learning in finance

Numerous architecture and deep learning methods have been applied to financial market for forecasting. The study conducted by Ding & al. [14] falls into the third category of the previous section. They vectorize textual news which is then given to a DCNN. The novel architecture captures the short and long term variations made by news on the S&P500. According to the authors, this approach outperforms other previously reported systems. Borovykh & al. [15] introduces a architecture based on CNN mixed with WaveNet [16] taking the prices as a time-series which are the input. They want to forecast the evolution of the stocks. They compare their model to baseline neural forecasting models including LSTM. This approach is close to the one presented by Honchar & al. [17]. They also based their architecture on WaveNet and they compares their own network to more classical architectures such as LSTM, regular CNN and the multi-layer perceptron. The data is extracted from the FOREX EUR/USD and S&P500. The results are the same as before: their architecture surpasses the other neural networks.

2.4.3 Convolutional neural networks

Convolutional layers are a particular kind of layer which have been used in various situations. They are able to exploit the local dependency of the input. A lot of research has been successfully conducted to tackle different problems. In computer vision, the state-of-art results are reached with CNN. Graham & al. [18] reached the best accuracy on CIFAR-100 with an adapted version of the CNN. Lee & al. [19] also reached one the top score with the Street View House Numbers (SVHN) Dataset. The local dependency can also be temporal. The recent article published by Zhang & al [20] presents interesting results in au-
Automatic speech recognition (ASR). The architecture keeps the convolutional scheme and integrates LSTM units. Regarding ASR, another study [21] using CNN exploits the specific structure (local connectivity, weight sharing and pooling) to show a form of invariance to small shifts of speech features along the frequency axis. This shift is often caused by speakers or environments variations which is major issue in ASR. The error rate is reduced compared to other deep neural networks. The task is done using the TIMIT phone recognition dataset.

These elements participate in the motivation of starting to explore the possible solutions of the research question with CNNs.

2.4.4 Limitations

A non-negligible part of the authors of the papers quoted above [4], [14], [15], [17] present their strategies as profitable and based on state-of-the-art machine learning. However, it is more complicated, indeed it is really difficult to compare performances for different reasons. There is no reference dataset in finance, for a lot of papers authors had to collect their own dataset, which might raise questions about the method and the quality of the information. Moreover, their model performances are often tested on historical data but replicating a strategy properly requires to avoid many flaws (fees, latency, spread, slippage, etc.). Most of the time, there is no mention of any technique used to counter these flaws. Also, they often compare their strategy to other very simple strategy on very short period of time which weakens the implication of their paper.

The company offers an infrastructure that handles most the flaws and works with datasets bought from professional brokers. The machine learning infrastructure also offers the capacity to backtest strategies with a very high accuracy.
Chapter 3

Method

The goal of this problem is to predict whether the asset is going up or down in the next minutes. Among the historical data of prices, some moments have been selected by experimented traders with the filter $F$. Those moments are called the points of interest (POI). The set of POI of length $M$ is called $T = (t_1, \ldots, t_M)$. The machine learning model takes the moments as inputs and predicts the next move of the price.

First, there is already an existing model able to predict with a certain accuracy the next variation of the prices. This model is based on decision trees and referred as $C_{\text{tree}}$. Therefore, the goal of the first part 3.1 is to make a first benchmark of performances for the new model based on deep learning $C_{\text{deep}}$.

After making this benchmark, in the second part 3.2 the problem remains the same except for the inputs. Indeed, the inputs given to the deep learning model will be adapted to the specific characteristics of a deep learning classifier.

3.1 Binary classification using financial indicators

3.1.1 Input data

For each POI, $t_k$, one computes a limited and fixed list of financial indicators and forms a vector $X^{(k)}$: 

23
\textbf{3.1.2 Label}

The goal of the problem is to predict whether the prices are going up or down in the next minutes. Let’s define the variable $\Delta$ in the next minutes. For each $t_k$, the label $Y^{(k)}$ is defined by:

$$Y^{(k)} = \text{sign}(C(t_{k} + \Delta) - C(t_k))$$  (3.2)

Where $C(t)$ is the price at the time $t$. $\Delta$ is constant and set up to 10 min by the experimented traders. The motivation for a such label is to detect the POI when the variations of the prices are important. The filter is supposed to detect interesting and relevant moments which means that it should be followed by a significant bullish trend (since the model is long). One wants to avoid strong bearish trends and if possible stagnant positions.

\textbf{3.1.3 Evaluation}

The evaluation is mainly done using the ROC-AUC. Since the training process uses the rolling learning, the evaluation of one metric gives back a vector, one value per period $P$. $L$ is the number of periods.

$$E = \begin{pmatrix}
\text{Metrics}(P_1) \\
\text{Metrics}(P_2) \\
\vdots \\
\text{Metrics}(P_L)
\end{pmatrix}$$  (3.3)

A first tool to compare across all periods can be to take the median or the average of this vector. Since the network returns a value between 0 and 1 and it is a classification problem, the output value can be interpreted as a probability, thus using the ROC-AUC for the evaluation makes sense.
However, comparing models implies comparing the vectors $E$ from each model. To properly compare them, it is important to make sure that the two vectors of evaluation are not drawn from the same probabilistic distribution. To check this property, the ANOVA test is conducted after the evaluation with the machine learning metrics.

### 3.1.4 Network’s Architectures

**Multi layer perceptron (MLP)** The first architecture investigated is the multi-layer perceptron because even if it is not the most adapted solution to the current problem, it is really easy to set up and it will make a first benchmark for the results.

**Convolutional neural network** From the literature, the CNN is a viable option. However, in this first classification problem there is no time dependency between the input. This solution consists in several convolutional layers ending with a softmax layer. The different layers will extract relevant features from the input and the last layer classifies the sample according the features extracted. Since all the layers are convolutional this model is denoted as deep full convolutional network (DFC).

**Inception networks (INC)** Another interesting architecture is based on the inception node [22]. Each block is using several convolutional layers in parallel. This architecture exploits the lack of prior on the solution. During the training, the network can decide which filter seems the most adapted to the current task.

### 3.1.5 Sparse connected layer

The multi-layer perceptron is a very general model with almost no prior. A possible improvement could be to transmit our knowledge of the problem directly into the architecture. A first observation of the financial indicators shows that several of them are correlated to each other when other are clearly not. Making subgroups among the indicators can enhance the performances of the network. In each subgroup, the indicators are correlated between each other. One financial indicator can belong to several subgroups. The information given by
one indicator among his group is not as reliable as if the whole group
send the same information. The latter is more robust. An indicator can
transmit more than one information, therefore it make sense to allow
the indicator to belong to several different groups.

The subgroups are made using the absolute value of the Pearson
correlation. Once the subgroups are defined, they are all linked to one
output neuron. There are $S$ different subgroups. The layer is com-
posed by $S$ nodes, each node is linked to the indicators belonging to
one subgroup. The weights are learned like a usual neural network.
Since it is easy for a neural networks to change a specific weight from
$w_i$ to $-w_i$, the subgroups contain features absolutely correlated between
each other. For a regular layer, the outcome $z_i$ of the i-th node can writ-

ten:

$$z_i = f\left( \sum_{j=0}^{N} w_{ij}x_j \right)$$

where $f$ is the activation function, $x_j$ the j-th input and $w_{ij}$ the
weight between the j-th input and the i-th output. In this sparse con-
nected layer, the outcome can be written as:

$$z_i = f\left( \sum_{j \in S_i} w_{ij}x_j \right)$$

This layer is the first layer of the network.

### 3.1.6 Special Bagging

Since the performances are really close to random and neural net-
works are stochastic, a study of the variance of the results should be
conducted to make sure that the gap between one model and random
is significant. Moreover, studying the variance of the results is also
a way to improve the model. Aggregating several slightly different
models can reduce the variance of a stand-alone model.

This is a special form of bagging since the data is not re-sampled
to fit every model. The difference between every model is based on
the seed of the different sources of randomness (initialisation of the
weights, stochastic gradient descent, etc.). The aggregation of the different classifiers is done by averaging. This aggregation adds a new hyper-parameter: the number of model used.

### 3.1.7 Weight propagation

Between each period the weights of the models are reinitialised. However, this behaviour by default does seem to be a loss of information. For a given period $P_i$, the network $Net_{i-1}$ trained on the previous period $P_{i-1}$ has converged and is in a local minimum. Since a large part of the data belongs to both $P_i$ and $P_{i-1}$ (cf. 2.2.6), the weights of the previous network $Net_{i-1}$ contains information useful for $Net_i$. To transmit this information, the weights of $Net_i$ are initialised with the weights of the previous network $Net_{i-1}$ at the end of the training. An exception occurs for the first model: in this specific case, the network is initialised as usual. This improvement can also increase the speed of training since the network has already learned some part of the data.

### 3.2 Binary classification using the raw data

For now, the network has been crafted to address a specific issue: with the same input and output can $C_{\text{deep}}$ outperform $C_{\text{tree}}$? However, keeping the same input as $C_{\text{tree}}$ is a constrain since the indicators have been crafted and assembled for $C_{\text{tree}}$. Therefore, in this section, the POI remains the same but for each moment $t_k$, the inputs are the prices and not the technical indicators.

#### 3.2.1 Input data

For each POI, the input data is not a set of several financial indicators computed from the prices anymore but the prices directly. For each POI, the input is set of $D$ candles. $D$ is a new hyper-parameter. As defined in 2.1.4, one candlestick contains 5 values (Open, High, Low, Close and the Volume: often called OHLCV).

The input is not a vector anymore but can be seen as a matrix:
$$X^{(k)} = \begin{pmatrix}
O(t_k - D) & O(t_k - D + 1) & \ldots & O(t_k) \\
H(t_k - D) & H(t_k - D + 1) & \ldots & H(t_k) \\
L(t_k - D) & L(t_k - D + 1) & \ldots & L(t_k) \\
C(t_k - D) & C(t_k - D + 1) & \ldots & C(t_k) \\
V(t_k - D) & V(t_k - D + 1) & \ldots & V(t_k)
\end{pmatrix} \quad (3.4)$$

This input is then fed to a convolutional network with a receptive filed of \((\text{kernelSize} \times 1)\) instead of \((\text{kernelSize} \times \text{kernelSize})\) for an image. The input is composed of 5 time series.

### 3.2.2 Network’s architectures

**Mixed convolutional neural network (MCNN)** Another architecture is tested in this problem: a mix between a fully convolutional network and the multi-perceptron layer. The first layers are convolutional and the last ones are fully connected. This new architecture comes from the practical difficulties to train the DFC. Indeed, the DFC appears to be extremely sensible to the hyper-parameters.

### 3.2.3 Representation of the data

One important drawback of the OHLCV is the absolute value of the input. Indeed, the value of the S&P500 evolves around 2640 (March 29th 2018) which is not very well suited to the neural networks. The data needs to be rescaled or pre-processed at least to facilitate the training. Thus, several representations of the data have been considered to better suit the neural network. Instead of giving the OHLCV, the first other representation retained is:

$$\begin{pmatrix}
O_t, H_t, L_t, C_t, V_t
\end{pmatrix} \leftarrow \begin{pmatrix}
\frac{L_t - O_t}{O_t}, \frac{H_t - O_t}{O_t}, \frac{C_{t+1} - C_t}{C_t}, V_t
\end{pmatrix} \quad (3.5)$$

This representation gives the relative position of the Low and High regarding the Open. Since the volume is not a price, it remains untouched.

The second representation is an unscaled version of the previous one: the close of the previous candle is subtracted to every price. The representation allows to only have the variation of the prices. The information linked to the actual price is lost.
\[(O_t, H_t, L_t, C_t, V_t) \leftarrow (O_t - C_{t-1}, H_t - C_{t-1}, L_t - C_{t-1}, C_t - C_{t-1}, V_t)\]

(3.6)

### 3.2.4 Regularization

In this context, the network is still prone to overfit the training dataset. A solution to avoid this and improve the generalisation is to add some regularisation. The overfitting was also visible with the values of the updated weights. To address this issue, one can integrate to the loss a penalisation term:

\[L_{\text{new}}(W) = L_{\text{old}}(W) + \lambda f(W)\]

(3.7)

where \(f\) is a scalar function. This function is usually the \(L_1\) or \(L_2\) norm. Using those functions require to tune the hyper parameters \(\lambda\): if \(\lambda\) is too high the network will only focus on reducing the weights without considering the classification problem and if the \(\lambda\) is too low the network will continue to overfit since there is almost no penalisation.

However, those norms do not correspond to our prior. The latter must contain the fact that values superior to a constant \(\alpha\) must be penalised but not the values under. There is no need to and our knowledge of the problem does not encourage us to have the smallest values as possible. The following function corresponds to this specific need:

\[f(W) = \sum_{w \in W} \max(0, w^2 - \alpha^2)\]

(3.8)

The idea behind this function is to penalise only the outliers and to discourage the neural network to reduce weights which already belong to an acceptable range: \([-\alpha, \alpha]\).
Chapter 4

Results

4.1 Binary classification with technical indicators

4.1.1 Classification task

Experiment

The training is done on a period of 3 years with a step size of 3 months between 2005 and 2015. The time frame of the input data is 1 minute. The training is done with the stochastic gradient descent and with some regularisation. The Nesterov momentum is used with a value of 0.99. There are 97478 trades opportunities. The letters S, W and B added at the end of the models (cf. 4.1) denotes the different options tested. S is the Sparse connected layer presented in 3.1.5, B stands for the special Bagging (3.1.6) and W for the Weight propagation (3.1.7). Therefore, the model INC-SB is based on the inception architecture uses the sparse connected layer at the beginning of the network and also the bagging method.

Results

The results of the classification problem are presented in table 4.1.

The sparse connected layer directly improves the mean of the ROC-AUC. This is verified for all models. Without this first layer, the achieved means of ROC-AUC are lower meaning that the intuition of directly
<table>
<thead>
<tr>
<th>Models</th>
<th>mean of ROCA</th>
<th>std of ROCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>50.8</td>
<td>3.24</td>
</tr>
<tr>
<td>INC</td>
<td>51.8</td>
<td>2.78</td>
</tr>
<tr>
<td>INC-S</td>
<td>52.6</td>
<td>2.93</td>
</tr>
<tr>
<td>INC-SB</td>
<td>52.9</td>
<td>2.25</td>
</tr>
<tr>
<td>INC-SW</td>
<td>53.3</td>
<td>3.07</td>
</tr>
<tr>
<td>INC-SWB</td>
<td>53.2</td>
<td>2.10</td>
</tr>
<tr>
<td>DFC</td>
<td>51.5</td>
<td>4.35</td>
</tr>
<tr>
<td>DFC-S</td>
<td>52.1</td>
<td>4.12</td>
</tr>
<tr>
<td>DFC-SB</td>
<td>52.0</td>
<td>3.24</td>
</tr>
<tr>
<td>DFC-SW</td>
<td>52.3</td>
<td>3.49</td>
</tr>
<tr>
<td>DFC-SWB</td>
<td>52.5</td>
<td>3.28</td>
</tr>
</tbody>
</table>

Table 4.1: ROC-AUC scores for the classification based on technical indicators. Methods are evaluated at each period (2.2.6) from which the mean and the standard deviation of the ROC-AUC are computed. Best results are **bold**. INC-SB achieves best mean of ROC-AUC but keeps a higher standard deviation of the ROC-AUC compared to INC-SW. Since means of ROC-AUC for INC-SB and INC-SW are very similar but not the standard deviation, **INC-SWB** is considered as best model.

introducing the knowledge of the data through the general architecture of the model is valid and effective. There is no particular effect on the standard deviation of the ROC-AUC.

The special bagging tends to increase the mean of ROC-AUC and also strongly reduces the standard deviation of the ROC-AUC as expected. This hypothesis, formulated in section 3.1.6, is verified. This result is very important since the standard deviation is directly linked to the robustness of the trading model. A classifier with the low deviation over the periods shows a resistance to the non-stationary parameters. The bagging tends to be correlated to higher mean of ROC-AUC but the trend is quite weak.

Regarding the weight propagation, the standard deviation and the mean of the ROC-AUC increase. The increase of the mean is strong and important for the general accuracy of the model. This observation is verified for both models and is also independent from the other im-
Finally, the combination of the special layer and the weight propagation explained the highest score achieved by INC-SWB concerning the mean of the ROC-AUC. Adding the bagging on top allows to decrease the standard deviation of the ROC-AUC. For each architecture this observation is confirmed, showing the independence between the results of the improvement and the results from the architecture.

Two models remain interesting after the tests (INC-SW and INC-SWB) because their mean of ROC-AUC is high. However, the standard deviation makes the difference between the 2 models. Indeed INC-SWB achieves the lowest standard deviation with a large margin. This combination means they outperform the other models on the predictions and they are steady. The prediction score does not vary too much between each period. This property is important and must be seek in the selected models as a trading model needs to be robust. However, the best model remains INC-SWB since his standard deviation of the ROC-AUC is lower. From now, this model is called $C_{\text{deep}1}$.

### 4.2 Binary classification with raw data

#### 4.2.1 Data representation

Several data representations are investigated to improve the general performances of our networks. In 3.2.3, two others representations are presented: the equation 3.5 shows the relative variation of the original data, denoted as RV and equation 3.6 presents the non-relative variation of the data, denoted as V. The OHLCV are denoted O. The data is preprocessed to zero-mean and with a standard deviation of 1. Changing the representation of the input data only prevent comparison with losses but comparing models with ROC-AUC still makes sense. For the 3 networks, the impact of the data representation is presented in table 4.2. The RV representation achieves best mean of ROC-AUC with slightly higher standard deviation than the other representations. A possible explanation of the counter performance of O and V are the dependency of the nominal asset’s price. Indeed, with a such pre-processing two input vectors with the same local variation but with a different average price will have different values after preprocessed-
ing, as presented in table 4.3. For the rest of the experiment, the RV representation is adopted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Data</th>
<th>mean of ROCA</th>
<th>std of ROCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC-BW</td>
<td>O</td>
<td>53.4</td>
<td>2.87</td>
</tr>
<tr>
<td>INC-BW</td>
<td>V</td>
<td>54.0</td>
<td>2.72</td>
</tr>
<tr>
<td>INC-BW</td>
<td>RV</td>
<td>53.6</td>
<td>2.36</td>
</tr>
<tr>
<td>DFC-BW</td>
<td>O</td>
<td>53.5</td>
<td>1.99</td>
</tr>
<tr>
<td>DFC-BW</td>
<td>V</td>
<td>54.1</td>
<td>2.18</td>
</tr>
<tr>
<td>DFC-BW</td>
<td>RV</td>
<td>53.8</td>
<td>2.07</td>
</tr>
<tr>
<td>MCNN-BW</td>
<td>O</td>
<td>54.1</td>
<td>2.43</td>
</tr>
<tr>
<td>MCNN-BW</td>
<td>V</td>
<td>54.5</td>
<td>2.55</td>
</tr>
<tr>
<td>MCNN-BW</td>
<td>RV</td>
<td>54.2</td>
<td>2.23</td>
</tr>
</tbody>
</table>

Table 4.2: Impact of the data representation on the scores achieved by models INC, DFC and MCNN. Best scores are **bold**. RV achieves best mean of ROC-AUC for each model. The standard deviation is slightly higher with RV.

### 4.2.2 Input size

The size of the input is managed by the hyper-parameter $D$ (cf. 3.1.1). A study is conducted to set a proper value to this parameter. For this, the 3 models INC-WB, DFC-WB and MCNN-WB are evaluated with different possible input size. The models are presented in 4.1.1. The results of this study are plotted in figure 4.1.

As showed, there is a decrease of performances if $D$ is too small which is understandable but also when $D$ is too high ($D > 150$). The later can be explained by the fact that a non negligible part of the POI is at the beginning of the day. Indeed, the data provided for a trade starting at 8:30PM when the stock exchange opens at 8PM and with $D = 100$ starts the day before. The data provided contains the 30 first minutes of the day and 70 the last minutes of the day before. Since, the exchange is closed during the night but the information and the news the price can be very different between the closing time and the opening time the day after. This is the gap overnight. In intra-day trading, those gaps are not taken into account, that is why the performances decreases when $D$ is too high.
Table 4.3: Example of sequences of prices and associated input data. O and RV does not provide the same input for two sequences with the same local variation but with an offset. This difference with V might explain the difference of performance in table 4.2.

For the next experiments, the input size is set to 100 which is value where the models seems to all achieve their best mean of ROC-AUC.

4.2.3 Regularisation

Neural networks are prone to overfit the training dataset. Adding a regularisation term, as presented in 3.2.4, is one possible approach to limit this effect. The chosen term is defined in equation 3.8. The hyper-parameter is set to 5 therefore every weight outside this range [-5, 5] is penalised. The regularisation’s effects can be observed through the distribution of the layer’s weights. The second layer’s weights of the network MCNN-WB are plotted as an histogram in figure 4.2. The large part of the weights belongs to [-10, 10].

Applying the penalisation term in the loss leads to a distribution of weights without any outlier, as presented in figure 4.3. The weights with large values are penalised and therefore disappear during training.

Adding the penalisation term improves the training since it limits the overfit. Therefore during the rest of the experiment, the regularisation term is added to the loss. The final value of the hyper-parameter $\alpha$ is set to 5.

<table>
<thead>
<tr>
<th>Sequence 1</th>
<th>$t-1$</th>
<th>$t$</th>
<th>$t+1$</th>
<th>$t+2$</th>
<th>$t+3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>950</td>
<td>1000</td>
<td>1030</td>
<td>1050</td>
<td>1010</td>
</tr>
<tr>
<td>V</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>-40</td>
<td></td>
</tr>
<tr>
<td>RV</td>
<td>0.0526</td>
<td>0.0300</td>
<td>0.0194</td>
<td>-0.0381</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence 2</th>
<th>$t-1$</th>
<th>$t$</th>
<th>$t+1$</th>
<th>$t+2$</th>
<th>$t+3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1950</td>
<td>2000</td>
<td>2030</td>
<td>2050</td>
<td>2010</td>
</tr>
<tr>
<td>V</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>-40</td>
<td></td>
</tr>
<tr>
<td>RV</td>
<td>0.0256</td>
<td>0.0150</td>
<td>0.0099</td>
<td>-0.0195</td>
<td></td>
</tr>
</tbody>
</table>
4.2.4 Classification task

The results of the classification problem with the raw data are presented in table 4.4.

Models using bagging still achieve lower standard deviation of ROC-AUC. This observation remains true. However, the best mean of ROC-AUC are now reached by model using bagging. Whereas before the trend was weak, this time the trend seems stronger. Indeed, according to table 4.1, the gain of mean of ROC-AUC was, in the classification task using the technical indicators between $-0.1$ and $0.3$. Now the gain is between $0.3$ and $0.7$.

The weight propagation also presents the same consequences as presented before. The general mean of the ROC-AUC increases along with its standard deviation.

The combination of weight propagation and bagging still achieves the highest mean of ROC-AUC. However, the gain of mean is, in this context of binary classification from raw data, not as strong as before. The difference might come from the absence of the special layer. With this format of input data, a such layer is not relevant.

Finally, the best score is achieved by MCNN-BW with a mean of ROC-
AUC of 54.5 and a standard deviation of 2.55. The best model of table 4.4, **MCNN-BW**, is now called $C_{\text{deep}2}$. This model reaches the best mean of ROC-AUC but also has one of the lowest standard deviation.

### 4.3 Statistical analysis of the results

The results tend to show differences between the means of each model. However, the ANOVA test can be conducted to assess that the differences are significant. The results of the relevant models are presented in table 4.5 and in figure 4.4.
Figure 4.3: Distribution of the layer’s weights trained with regularisation. The weights with highest absolute value have disappeared because of the regularisation function added in the loss.

Figure 4.4: Comparison of the 3 methods ($C_{\text{tree}}$, $C_{\text{deep}1}$, $C_{\text{deep}2}$) evaluated with ROC-AUC as a function of the period.

The null-hypothesis would be that every results are drawn from the same distribution of results. In order to exclude this hypothesis, the F-ratio is computed in table 4.6.

The p-value is very low therefore it allows to say that the null-hypothesis can be rejected with a probability of $1 - (p - \text{value}) = 0.999$. Thus the distribution from which the results are drawn is not the same.
Table 4.4: ROC-AUC scores for the classification with the raw data. Best scores are **bold**. MCNN outperforms with a large margin the other methods (INC and DFC). Combined with the bagging and the weight propagation, **MCNN-BW** achieves the best performances. Even if the lowest standard deviation of the ROC-AUC is reached by **DFC-BW**, **MCNN-BW** keeps a relatively low ROC-AUC std compared to the other methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>ROCA mean</th>
<th>ROCA std</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>52.9</td>
<td>3.58</td>
</tr>
<tr>
<td>MLP-B</td>
<td>53.2</td>
<td>2.84</td>
</tr>
<tr>
<td>MLP-W</td>
<td>52.7</td>
<td>4.85</td>
</tr>
<tr>
<td>MLP-BW</td>
<td>53.4</td>
<td>2.79</td>
</tr>
<tr>
<td>INC</td>
<td>52.8</td>
<td>4.78</td>
</tr>
<tr>
<td>INC-B</td>
<td>53.6</td>
<td>3.05</td>
</tr>
<tr>
<td>INC-W</td>
<td>53.7</td>
<td>5.01</td>
</tr>
<tr>
<td>INC-BW</td>
<td>54.0</td>
<td>2.72</td>
</tr>
<tr>
<td>MCNN</td>
<td>54.0</td>
<td>3.54</td>
</tr>
<tr>
<td>MCNN-B</td>
<td>54.2</td>
<td>2.34</td>
</tr>
<tr>
<td>MCNN-W</td>
<td>54.1</td>
<td>3.67</td>
</tr>
<tr>
<td><strong>MCNN-BW</strong></td>
<td><strong>54.5</strong></td>
<td>2.55</td>
</tr>
<tr>
<td>DFC</td>
<td>53.0</td>
<td>3.37</td>
</tr>
<tr>
<td>DFC-B</td>
<td>53.5</td>
<td>2.36</td>
</tr>
<tr>
<td>DFC-W</td>
<td>53.8</td>
<td>3.71</td>
</tr>
<tr>
<td>DFC-BW</td>
<td>54.1</td>
<td><strong>2.18</strong></td>
</tr>
</tbody>
</table>

4.4 Post hoc analysis: Honest Significant Difference

A post-hoc analysis can help us to distinguish groups between each other. For this, the Honest Significant Difference (HSD) can be run to find out which group is different from each other. This test compares all three means between each other. The general criteria is determined by:
Table 4.5: ROC-AUC scores for each model. Best scores are bold. Even if $C_{tree}$ achieves the best ROC-AUC mean, the ROC-AUC std is lower for $C_{deep1}$ and $C_{deep2}$ by a large margin.

<table>
<thead>
<tr>
<th>Models</th>
<th>Architecture</th>
<th>ROCA mean</th>
<th>ROCA std</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{deep1}$</td>
<td>INC-SBW</td>
<td>53.2</td>
<td>2.25</td>
</tr>
<tr>
<td>$C_{deep2}$</td>
<td>MCNN-BW</td>
<td>54.5</td>
<td>2.55</td>
</tr>
<tr>
<td>$C_{tree}$</td>
<td>–</td>
<td>56.6</td>
<td>4.05</td>
</tr>
</tbody>
</table>

Table 4.6: Computation of the ANOVA test on the results of ($C_{tree}$, $C_{deep1}$, $C_{deep2}$). DF, SS and MS respectively denotes the degrees of freedom, which are intermediate computation for the F-ratio. The high value of the F-ratio allows to reject the null hypothesis and to consider the margin between the ROC-AUC mean as statistically significant.

<table>
<thead>
<tr>
<th></th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>2</td>
<td>1.46.10^{-2}</td>
<td>7.32.10^{-3}</td>
<td>7.61</td>
<td>0.001</td>
</tr>
<tr>
<td>Within Groups</td>
<td>72</td>
<td>7.01.10^{-2}</td>
<td>9.73.10^{-4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results of the ANOVA test and the post hoc analysis show that the results of each models are significantly different from each other, meaning that the models are different except between $C_{deep1}$ and $C_{deep2}$ for which the risk to assess it is quite large.

4.5 Trading strategy

The evaluation of the trading model is made with several indicators:

- The returns (2.2.3)
### Table 4.7: HSD test. The table presents the numerical application of the HSV. $C_{\text{tree}}$ can be considered as different from $C_{\text{deep1}}$. However, the two models $C_{\text{deep1}}$ and $C_{\text{deep2}}$ are quite close. Even if they could be considered as drawn from different distribution the p-value is high.

<table>
<thead>
<tr>
<th>Models</th>
<th>Mean diff</th>
<th>p-value</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{\text{tree}}$</td>
<td>0.0215</td>
<td>0.045</td>
<td>True</td>
</tr>
<tr>
<td>$C_{\text{tree}}$</td>
<td>0.0338</td>
<td>0.001</td>
<td>True</td>
</tr>
<tr>
<td>$C_{\text{deep1}}$</td>
<td>0.0123</td>
<td>0.350</td>
<td>False</td>
</tr>
</tbody>
</table>

### Table 4.8: Financial scores of the trading models. The trading models are composed of the pre-filtering followed by the classifier. $C_{\text{tree}}$ surpasses every other model but $C_{\text{deep2}}$ is close. The lowest exposition time is achieved by $C_{\text{deep2}}$.

<table>
<thead>
<tr>
<th>Models</th>
<th>Returns</th>
<th>Sharp</th>
<th>Sortino</th>
<th>Exposition (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{\text{ref}}$</td>
<td>53</td>
<td>4.8</td>
<td>3.5</td>
<td>95</td>
</tr>
<tr>
<td>$C_{\text{hand}}$</td>
<td>35</td>
<td>4.4</td>
<td>3.9</td>
<td>84</td>
</tr>
<tr>
<td>$C_{\text{deep1}}$</td>
<td>62</td>
<td>5.2</td>
<td>4.6</td>
<td>116</td>
</tr>
<tr>
<td>$C_{\text{deep2}}$</td>
<td>103</td>
<td>6.8</td>
<td>5.4</td>
<td>58</td>
</tr>
<tr>
<td>$C_{\text{tree}}$</td>
<td>142</td>
<td>7.2</td>
<td>5.9</td>
<td>72</td>
</tr>
</tbody>
</table>

- The sharp ratio (2.2.3)
- The sortino ratio (2.2.3)
- The equity curves (2.2.3)
- The exposition, this criteria evaluates the risk taken for a strategy. It is the average duration of a trade.

#### 4.5.1 Financial Metrics

The results given by the backtest are presented in table 4.8.

#### 4.5.2 Equity curves

Figure 4.5 presents the equity curves for the tested models. The fees (cf. 2.3.2) are included.
CHAPTER 4. RESULTS

Figure 4.5: Equity curves of the final models ($C_{ref}$, $C_{deep1}$, $C_{deep2}$, $C_{tree}$, $C_{hand}$). Curves are computed on the historical dataset starting in 2005 and ending in 2015. All the models are vulnerable to long period of decrease: in 2008 the financial crisis impacts all trading models. $C_{tree}$ and $C_{deep2}$ achieved the best return of the historical data with a large margin.

The equity curves are computed over the complete historical dataset. All models are impacted with the financial crisis in 2008. This observation definitively makes sense since all the trading strategies are long (2.2.1) and therefore expect a bullish trend. However, smaller and shorter crises do not impact all the models. Some of them ($C_{deep1}$ and $C_{hand}$) are even able to be profitable during the small crash of 2011 (August and September). It is possible since the trading model is risk averse and the crash only lasts for one period. The other model are relatively flat during this period. The strategy developed by human expert ($C_{hand}$) is profitable till 2013 and keeps loosing money after. This observation is expected. Indeed, the development of this method on the historical data before 2011-2010. Therefore the hand crafted strategy cannot hold and be profitable too long after without being updated.
Chapter 5

Discussion

5.1 Implications

The main implication of this project is the promising results obtained for the presented method. Indeed, scanning the history to select possible situations where a deep learning classifier can help traders to make decisions is unusual in the literature. After only a few months the possibility to implement this strategy can be seriously consider. Even if the model $C_{tree}$ performs better, the time spent on each model is largely different: a few month for $C_{deep}$ compared a couple of years for $C_{tree}$. Moreover, the strong focus given to avoid numerous flaws into the backtest and the focus given into the quality of the data strengthens even more the future opportunities of this model.

The degree project falls between two very different topics: machine learning and financial markets. A recurrent problem in such situation occurs: lack of knowledge might appear in one topic. This thesis also aims to present and explain implicit challenges induced by financial markets. Chapter two focuses on explaining technical aspects of the trading processes to avoid failure during the creation of a trading strategy. As explained in section 2.4 many papers ([1], [4], [14], [15], [17]) omit to explain their solutions to deal with market impact, slippage or spread. This project contributes to clarify underlying processes and consequences of trading which is essential for building accurate and efficient trading strategies especially without any background in finance.
5.2 The deep learning classifier

The first observation from the result is the important gain provided by the bagging method. Indeed, the special bagging brings a reduced variance of the model, which was expected but also slightly increases the classification ability of the classifier. The reduction of variance is correlated to the robustness of the model.

The second observation concerns the weight propagation. The gain of performance is significant. It can be exploited to perform a first general quick study over a large kind of architecture and/or over several hyper-parameters. Since weight propagation also increases the overall performances of the trading model, this improvement could be investigated in other field using rolling learning.

The format of input also changes and validates the intuition that CNNs are able to exploit the time dependency of the time-series thanks to the specificity of the convolutional layer. The importance of the format and the pre-process of the input data is also demonstrated since one method surpasses the others.

5.3 Evaluation

Even if this thesis provides answers and solutions to numerous issues related to backtest modelling and rigorous methods of evaluation, several aspects could have been improved. As mentioned in chapter 2, there is no public finance dataset. This paper does not solve this issue and remains in the continuity of the research in this field. Therefore, there is still no solution to properly compares different methods without implementing them. As an example, in computed vision, for detection and segmentation several datasets like COCO or PASCAL are widely used. It really simplifies the comparison problem and therefore accelerate the research in this topic.

Another possible drawback of this paper is the lack of implementation of other methods for comparison. It is mainly due to two reasons: intractability to implement other papers and time constrain. Indeed,
chapter 2 presents a large amount of interesting papers having methods to tackle same issues. The lack of information regarding hyperparameters and the amount of time required to gather the data make the implementation time consuming and uselessly complicated. That is why, this project instead has focused on the quality of the financial market modelling.

5.4 Future work

5.4.1 Interpretability of the forecast

An important element to look into is the interpretability of the classifier’s outcome. Understanding the underlying reasons of the decision made by the classifier could improve and help to understand the weaknesses and the strengths of the strategy. But it also could reveal flaws or unknown situations where the classifier performs poorly.

5.4.2 Integration of other type of information

Finally, from a more pragmatic point of view the study could be expanded in several ways. The network lacks of certain vital information available for traders such as time and price. Making the network understand that several days or hours during the week the behaviour of the market is not the same: the market can act differently with the same conditions except for the hours. As an example, one can think of the release of economical statistics on weekly basis. The exchanges are often very tense a few hours before the releases.

5.4.3 Expanding the access on data

Another possible way to improve the network could be to feed it with different time frames. Indeed, the input consists of 5 times series of 100 elements. Since the time frame is set to one minute, the oldest information given is 100 minutes old. Combining the current input with other times series of different time frames (for instance the Open price every one hour) could enlarge the amount of information available without flooding the network into too much data. In this thesis, only one time frame is used.
5.5 Ethics and sustainability

Automatic trading might raise concern regarding ethics and sustainability. I will not expand on the trading itself which is, since a long time, a reality. However, the method presented in this context focuses on intraday trading, meaning that every position is sold before the closing hour. The intraday trading is also very different from the high frequency trading. As exposed in 4.1, the mean exposition time is at least several minutes. Meaning that humans can compete with the algorithm. High frequency trading exploits the communication between different market places of the same exchange. It requires very specific and expansive materials to make benefits. Therefore, it is not unethical to get advised by an algorithm which is just able to use more information than a expert.

The main issue is based on the interpretability of classifier’s decisions. Indeed, without properly knowing the important criterion for the classifier, the decision can be based on illegal tricks. As an example, the classifier could learn its market impact (2.3.3) and exploit it to increase benefits. This is unfair and illegal since the others traders must follow the law which bans market manipulations. Indeed, with large and available assets, it is possible to manipulate the evolution of prices. If the decision of the algorithm can not explained, it is hard to exonerate it from any manipulations.

Another important concern is the responsibility of the decision made. If a crash occurs because of several algorithms taking incomprehensible decisions between each other. This might have consequences on real life, companies might go to bankruptcy, might lose investors. In this context, the moral responsibility is not clear: the designer of the algorithm, the responsible of training, the owner of the product or the user. This problematic belongs to a larger issue which is not only related to financial market: decision making based on algorithms and more particularly based on methods linked to deep learning since the interpretability is often hard to explore and understand.
Chapter 6

Conclusion

In this thesis, we examined whether a classifier based on CNN can surpass the performance reached by experiments traders in pre-filtering trade classification. To test this hypothesis, the network was trained on financial data. The input were the prices and the network has been customised to perform this task and handle this kind of data. The training has been done using the rolling learning and was evaluated on each period. To make sure that the results obtained were significantly different from each other, the ANOVA test and a post-hoc analysis were performed.

We found that the deep learning classifier is able to surpass humans but it is not the only model. From a financial point of view, the classifiers based on machine learning $C_{tree}$ and $C_{deep}$ are able to outperform experts. The results based on machine learning metrics obtained for the models are different enough to consider that the models are significantly independent. The several improvements made during the training on the networks (pre-process on data, special bagging and the weight propagation) all participate in the achievement of those results.

All in all, these results suggest that this specific approach of pre-filtered trading opportunity improved with a CNN classifier is viable and could perform well if the method was successfully implemented.
Bibliography


