An LSTM and GRU based Trading Strategy Adapted to the Moroccan Market

Yassine Touzani (touzayassine@gmail.com)

Hassan II Mohammedia University Faculty of Science and Technology Mohammedia: Universite Hassan II Casablanca Faculte des Sciences Techniques Mohammedia https://orcid.org/0000-0003-0911-9780

Khadija Douzi

Hassan II Mohammedia University Faculty of Science and Technology Mohammedia: Universite Hassan II Casablanca Faculte des Sciences Techniques Mohammedia

---

Research

Keywords: Machine Learning, Deep Learning, LSTM, GRU, Financial times series, Moroccan stock market

Posted Date: June 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-526234/v1

RESEARCH

An LSTM and GRU based trading strategy adapted to the Moroccan market

Yassine Touzani* and Khadija Douzi

*Correspondence:
touzayassine@gmail.com
Computer Lab of Mohammedia LIM ,Faculty of Sciences and Technology of Mohammedia (FSTM) Hassan II university, Mohammedia, Morroco
Full list of author information is available at the end of the article

Abstract

Forecasting stock prices is an extremely challenging job considering the high volatility and the number of variables that influence it (political, economical, social, etc.). Predicting the closing price provides useful information and helps the investor to make the right decision. The use of deep learning and more precisely the recurrent neural networks RNNs in stock market forecasting is an increasingly common practice in the literature. The Long Short Term Memory LSTM and Gated Recurrent Unit GRU architectures are among the most widely used types of RNN networks, given their suitability for sequential data. In this paper, we propose a trading strategy designed for the Moroccan stock market, based on two deep learning model: Long Short Term Memory LSTM and Gated Recurrent Unit GRU to predict respectively close price for short and mid term horizon. Decision rules for buying and selling stocks are implemented based on the forecasting given by the two models, then over four three-years periods, we simulate transactions using these decision rules with different parameters for each stock. We only hold stocks that ensure a return greater than a benchmark return over the four periods. Random search is then used to choose one of the available parameters and the performance of the portfolio built from the selected stocks will be tested over a further period. The repetition of this process with a variation of benchmark return makes it possible to select the best possible combination of stock each with the parameters optimized for the decision rules. The proposed strategy produces very promising results and outperform the performance of indices used as benchmarks in the local market.

Keywords: Machine Learning; Deep Learning; LSTM; GRU; Financial times series; Moroccan stock market

Introduction

The semi-strong form of market efficiency hypothesis[1] which states that security prices react very quickly to publicly available new information is commonly accepted in the finance field. Furthermore, market data are extremely volatile and noisy, as a result, predicting stock prices is a notoriously challenging task. However, there is a slew of market anomalies that contradicts the efficient market theory [2] hence the need for trading strategy to better trade stocks able to outperform the market. Accurate forecasts provide valuable information to investors and allow them to adjust their position (buy, hold or sell) according to the price trend, it also allows building a winning trading strategy. This will explain the large number of works published in this field in recent years. In all of these articles, we find a lack of works devoted to the Moroccan market, and this is primarily the motivation that drives our work. We propose a new trading strategy tailored to Moroccan market based on

LSTM and GRU models, trained with data from the US and French markets and validated on the Moroccan market. Stock buying and selling is driven by decision rules implemented based on forecasting provided by the LSTM and GRU models, then several simulations and random search are used to fine-tune our trading strategy parameters. The rest of this paper is structured as follows: In the related works section, we go through a non-exhaustive list of previous work in the field of stock price prediction using deep learning, with a focus on the work that uses LSTM and GRU networks. Then, in the following section, we provide a more detailed overview of the proposed trading strategy components. The Experimental results section will be dedicated to the experiments as well as the results obtained, accompanied by the Discussion section in which we comment on the results obtained. Finally, this paper concludes with a conclusion and future work.

Related Work

There is no model adapted to the Moroccan market in the literature; however, many works have been proposed for other markets. In [3] Budiharto propose an LSTM-based approach for stock price forecasting in Indonesia. Yadav et al [4] propose an optimized LSTM for Indian stock market forecasts. The Sri Lanka market was the subject of an RNN model proposal by Samarawickrama et al in [5]. Nti et al [6] use artificial neural network to predict future movement of stock price in Ghana for 1, 7, 30, 60 and 90 days ahead based on public opinion. Regardless of the intended market, the LSTM and GRU models are widely used in financial forecasting purpose. Liu et al [7] proposes a new hybrid system for one day ahead forecasting, closing price data will be decomposed into several components using Empirical wavelet transform algorithm [8], LSTM predictors with a dropout are built to predict the decomposed closing price time series, LSTM model hyperparameters are selected using the PSO algorithm [9]. Error correction is carried out using the ORELM method [10]. Finally the LSTM forecasts and the ORELM error forecasts are added to produce the final closing price prediction. The proposed framework produces excellent results (MAPE $\approx 0.15\%$). In [11] Althelaya et al used GRU and LSTM units to build a stacked and bi-directional architecture for short term (1 day ahead) and long term (30 day ahead) forecasting of the SP500 index. The results of the GRU and LSTM models are very close and outperform the MLP, which was also used in this study. In [12] Patel et al propose a model incorporating LSTM and GRU for one, three, and seven days ahead prediction of Litecoin and Monero cryptocurrency. In order to built a threshold based portfolio Lee and Yoo [13] develop three types of RNN models : classical RNN, LSTM and GRU to forecast one month ahead, the top ten stocks in Standard and Poor's 500 index using monthly data (OHLCV: open,high,low,close and volume). They conclude that the LSTM outperformed the two other architectures. In [14] Cao et al propose a new hybrid financial time series forecasting, decomposition part is carried out using empircical mode decomposition EMD [15] and complete ensemble empirical mode decomposition with adaptive noise CEEMDAN [16], an LSTM model is then trained on the intrinsic mode function IMF including the residual component, the final forecast is obtained by adding the set of predictions obtained for each component.

Proposed Model

In this paper, we propose a new trading strategy tailored to the Moroccan market, based on two deep learning models, LSTM model provides short-term prediction while GRU model predict the medium-term. Once the predictor component provides the price forecasts, a suitable decision rule for each stock should be established to complement the proposed strategy. Before going in depth of the proposed approach, we will give a brief overview of LSTM and GRU architectures.

Long Short Term Memory

Recurrent Neural Networks RNNs are a type of neural network widely used in the field of deep learning [17] [18] [19], it turns out that classical RNNs are extremely difficult to train to handle long term dependency [20] because of gradient vanishing problem (gradient exploding can occur also but very rarely). To overcome the vanishing gradient problem LSTM is proposed initially by S. Hochreiter et al [21], then improved in the paper of F. Gers and J. Schmidhuber [22]. The LSTM unit is the most basic component of an LSTM architecture; it's a series of gates and cells that work together to produce a final result. The forward pass of an LSTM unit is modeled by equations (1-6)

$$f_t = \sigma(W_f.x_t + U_f.h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$\widetilde{c_t} = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

where σ is sigmoid function, f_t is forget gate activation vector, i_t input gate activation vector, o_t output gate activation vector, $\widetilde{c_t}$ cell input activation vector, c_t cell state, h_t output vector of LSTM unit, all W and U are weights, b is bias vector and symbol * for handamart product (element wise product). Weights W, U and biases b are to be learned during the training process. In order to better decipher

the equations cited above, let's start with cell state c_t, it contains two kinds of information: old information to keep from past state c_{t-1} specified using forget gate, it decides the percentage of information to keep by computing a value between 0 (throw completely) and 1 (keep completely), and new information to include in the cell state calculated using input gate i_t and cell activation $\widetilde{c_t}$ which are computed using (2) and (3) respectively. Finally the calculation of the final output value is performed in two steps. A potential value is calculated using (5) for the first time. This value will be regulated using the information present in cell state as indicated in (6). The use of the cell state in the final calculation makes the LSTM powerful in tasks where information must be stored and used later (long term). Language modeling is a simple example of this situation, in which the subject at the beginning of the sentence decides the conjugation of the verb to be used in the middle or even at the end of the sentence.

Gated recurrent unit

Gated Recurrent Unit GRU was introduced in 2014 by Cho et al [23] to solve the vanishing gradient problem experienced by classical recurrent networks. Same as LSTM, the input value interacts with the information from the previous state to calculate the different values of intermediate gates which will subsequently be used to decide on the value to be output. The forward pass of a GRU unit is modeled by equations (7-10)

$$z_t = \sigma(W_z.x_t + U_z.h_{t-1} + b_z) \tag{7}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{8}$$

$$\widetilde{h_t} = tanh(W_h x_t + U_h(r_t * h_{t-1}) + b_h) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{10}$$

where σ is sigmoid function, z_t is update gate activation vector, r_t reset gate activation vector, $\widetilde{h_t}$ candidate vector and h_t is output vector of GRU unit. W and U are weights, b is bias vector and symbol * for handamart product (element wise product). Same as for LSTM, weights and biases are to be learned during the training process. Let us go deep through the equations cited above to better understand how a GRU unit works. First update gate is computed using input vector x_t, output of previous unit h_{t-1} and sigmoid activation function. Then reset gate is calculated in the same way as update gate using its own weights and bias, the reset gate is then involved in the candidate value calculation, determining how much information from the previous state should be preserved indeed, from equation (9) we can

notice that with an r_t value very close or equal to zero, only the input value will be considered in candidate value calculation. Finally, the output value is calculated by calibrating the previous output and the new candidate output. The calibration is carried out by the updating gate z_t. ($z_t = 0$ copy the previous output, $z_t = 1$ generate a new output regardless of the old output). We can observe a kind of similarity between LSTM and the GRU, indeed both implement an intermediate gating mechanism which will be used in the output value calculation. Regarding the two models performance, Chung et al [24] demonstrated that GRU outperforms LSTM in a number of tasks, this observation is also shared by Jozefowicz et al [25] who found that GRU outperforms LSTM in a number of tasks except language modeling. On the other hand Shewalkar et al [26] finds that the LSTM outperforms GRU in the speech recognition task, however, they confirm that the GRU optimization is faster. In general, both LSTM and GRU are powerful and well-suited to sequential data. The GRU has the benefit of faster optimization comparing to an LSTM because it has less parameters.

Prediction component

This part aims to forecast short and medium-term price pattern, a high forecasting accuracy will lead to a successful trading strategy. We will be interested in the moving average described using the formula (11) to better capture the trend:

$$y_t^h = \frac{1}{h}\sum_{i=1}^{h} y_{t+i} \tag{11}$$

where y_t is close price at time t, we will use the notation y_t^h to design moving average of next h days at time t. It will be our target variable we are attempting to forecast, it reduces daily price noise and summarizes well the trend information. Fig. 1 is a more accurate depiction of how the price average captures trend data. Once we identify our target variable we need to deal with data issue, indeed the Moroccan market presents peculiarities which distinguish it from other well-developed markets such as the european or american market. The major problem encountered lies in lower liquidity, indeed we observe a discontinuity of exchanges for a large number of shares as shown in Fig. 2. It has often occured that a stock is neither sold, nor bought for several days or even several weeks or months. Which automatically results in missing data for those trading days. In addition the Moroccan market only offers 76 stocks for trading [27]. This number associated with the missing data problem reduces considerably the amount of data obtained over a given number of years. It is well known, acquiring large dataset contributes to enhance the model performance and throw it away from overfitting phenomena [28]. To address the problem of data volume we will use external data from other markets (American and French market). These collected data, make it possible to obtain a large dataset. With regard to the problem of trading discontinuity, we will rule out the stocks that suffer from liquidity issue and only the relatively demanded shares on the market will be retained. At this level, we have all the required elements to train our two deep learning models. Data from the US and French market will be used as training data and morrocan data will act as validation and testing data. Using only

historical prices, two recurring networks will be trained to forecast future average closing prices for the short and medium-term. Although training data was gathered from external markets. The use of local data as validation will allow to adjust the hyperparameters according to the Moroccan market characteristics.

Parameters tuning

To develop a successful trading strategy, forecasts should be followed by good decision rules. Let $\hat{y}_t^m$ and $\hat{y}_t^s$ the predictions given by our predictor component for medium and short-term horizon respectively. We define the following ratio:

$$\theta = \hat{y}_t^m / y_t \tag{12}$$

where y_t is closing price at time t. This ratio indicates how is the medium-term prediction compared to the actual price. A value greater than one mean price is expected to rise during the upcoming days. We will try to identify optimal threshold θ^*, when it is outstripped, the trading is profitable. The methodology used to evaluate the θ^* will be detailed below once all the decision rules of the strategy have been described. In the trading field, buying a financial instrument is known as open position, and selling it is called close position. We only open a position for our proposed strategy when 3 conditions are verified:

- $\theta >= \theta^*$.
- $\hat{y}_t^s > y_t$. where $\hat{y}_t^s$ is the sort-term forecasting and y_t is the actual closing price.
- There is no open position with the stock to buy.

The first condition seems quite clear, medium-term forecasting should exceed a given threshold (tuned previously for each stock). With the respect to the second condition, it completes the strategy and regularizes open position timing. Indeed, even though if we expect prices to increase in the medium-term horizon, price can decrease in the short-term horizon before increasing. In this case we delay the purchase to buy at a lower price. Finally, no open position with the stock to buy for simplification reasons. Finally, to close an open position, two conditions needs to be checked:

- $\theta < 1$
- $\hat{y}_t^s < 1$

Concerning close position conditions, it seems more intuitive, we close the position when prediction is less than the current price for short and medium-term horizon. To secure trading operation, trader typically sets a stop limit, which is the minimum price when it is hit, the operation is immediately closed in order to reduce the losses in case of strong prices drop. Notice that no stop limit is defined in our proposed strategy, it is extremely risky, but the ground truth proves us right, the results obtained support this decision. Now that the strategy is well detailed, we can describe the tuning methodology for θ parameter. We will simulate the strategy for each stock over periods of equal duration (3 years each, for example), and we will track the performance (annualized return) of each stock using various θ values. Throughout all simulation times, we record the return achieved by all stocks for each value of the θ parameter. Then, we will set a return threshold, and we will

retain the shares whose returns exceed this threshold for all time periods with their associated θ parameter. The selected stocks are then used to build a portfolio, and the performance of this portfolio is checked over a new period using the same buy and sell rules mentioned above. For stocks with multiple θ parameter values that meet the selection criteria, we will choose θ at random(Several repetitions will be performed to check various θ values). This whole process will be carried out for a variety of threshold values. The suggested method is a hybridization between grid and random search. It is obvious that raising the threshold value leads to a decrease in the stocks that make up the portfolio, hence the idea of testing different threshold values to evaluate portfolios of différent sizes. Finally, the shares in the portfolio that have the maximum observed return , will be retained by our strategy. It is important to highlight that in all the simulations brokerage fees are estimated and added to buying prices or subtracted from selling prices.

Experimental results

Dataset description

In order to train predictor component models, we will use two sources of data: the first one is data of all stocks that constitute SP500 index (except BRK.B and BF.B discarded due to data loading error) extracted from yahoo finance using pandas datareader library [29]. The second is data of all the CAC40 index stocks collected from investing.com using investpy library [30]. Extracted data are from January 2010 to January 2019. Concerning Morrocan data, to escape discontinuity issue described in prediction component section we will be working with stocks that have a relatively high average volume traded in 2019 and 2020. As a result, out of the 76 stocks available, we choose 32. Data from January 2010 to January 2019 will be used as validation data to tune predictor hyper parameters. Data from march 2019 to March 2021 will be used to test (hold out dataset) our proposed strategy performances. Note that the Moroccan data are extracted in the same way as the CAC40 data.

strategy compoenents

We will try 2 types of RNN architectures for the predictor component. For short-term prediction, we will train an LSTM model, regarding medium-term forecasting, we train a GRU model. All models are designed with Keras API [31], number of layers and unit are determined after a random search. Weights are initialized using He initializer [32] and optimized using adaptive moment estimation (Adam) algorithm [33]. Finally Dropout [34] is used as regularization technique and Gaussian noise is added for better generalization [35]. Once the prediction component is completed, a simulation is run over the following periods from 2010 to 2012, 2011 to 2013, 2012 to 2014 and 2013 to 2015, using a set of values of θ parameter and for all stocks. We calculate stock return obtained over each period using the following formula:

$$r = \frac{y_f - y_i}{y_i} \tag{13}$$

where y_f and y_i are respectively final sell price and first buying price. Then return is annualized using the formula below:

$$r_a = (1 + r)^{1/years} - 1 \tag{14}$$

where r is return calcualted above and years is number total of years, 3 years in our case for each period. The top three results for each simulation time, as well as the parameter used, are shown in Table 1. Then a benchmark return is created. We pick the stocks that outperform the benchmark over all simulation periods, and we calculate the strategy performance with selected stocks combination over a new period (from 2016 to 2019). By varying benchmark value, we obtain a new combination of stocks and new performance, we experimented a set of benchmark values and chose the combination of stocks that yielded the highest return. Note that the amount invested is the same for each stock, the profit generated from trade is fully reinvested and finally the return is calculated at the end of the period using the initial amount and the final amount generated.

Results

LBV (Label vie) and MDP (Med paper) are the stocks to keep in our watched list, with θ parameters of 1.006 and 1.026, respectively. In order to evaluate our proposed strategy performance, we will use data between March 2019 and March 2021. Table 2 illustrates the individual performance of our final stocks, while Fig. 3, 4 represent the buying and selling times for the same stocks respectively, and Fig. 5 highlight return generated per trade. In addition to individual stock results, the performance of our proposed strategy will be compared to the MASI index (Moroccan all share index) as well as the performance of the following sector indices: Distributors, Pharmaceuticals Industry, and Software & IT Services. Sector indices were chosen based on their results (annualized return $> 15\%$). Table 3 highlight this comparison. Notice that the decision to compare our results with indices results is explained by the lack of a publication dedicated to the Moroccan market, as well as the fact that the indices performance have always acted as a benchmark for evaluating UCITS (Undertaking for Collective Investments in Transferable Securities) performance.

Table 1 top 3 stock performance by simulation period

Start date	End date	stock	θ	annualized return(%)
2010	2012	LBV	1.007	18.459
2010	2012	CDM	1.011	13.217
2010	2012	MDP	1.052	10.106
2011	2013	LBV	1.007	14.731
2011	2013	JET	1.010	12.640
2011	2013	BMCI	1.012	12.204
2012	2014	MDP	1.026	16.947
2012	2014	LBV	1.014	16.725
2012	2014	ATH	1.012	15.098
2013	2015	COL	1.007	29.441
2013	2015	ATH	1.009	23.062
2013	2015	SNA	1.010	19.321

Table 2 selected stock performance over period 03/2019 to 03/2021

stock	θ	glboal return(%)	annualized ret(%)	winerate(%)	Number of closed position	hloding time(days)
LBV	1.006	-8.45	-4.50	85.00	20	4
MDP	1.026	55.78	26.02	93.75	16	14

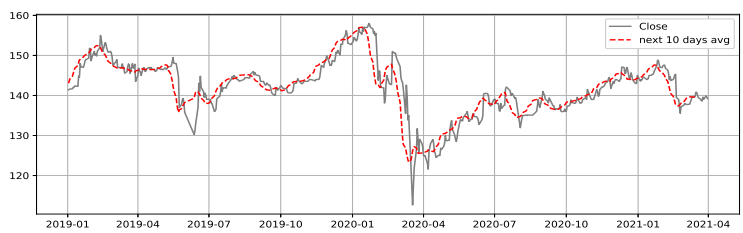

Figure 1 Price with next days Moving average for IAM stock

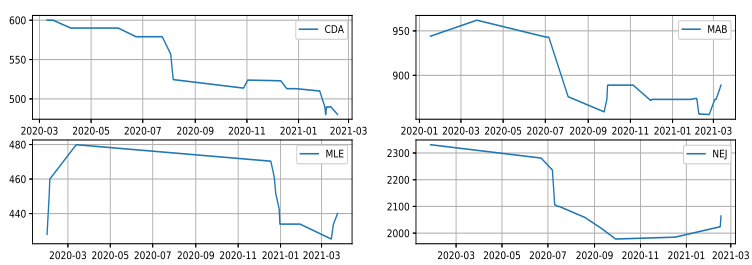

Figure 2 Example of stocks with trading discontinuity

Table 3 comparison of performance over the period from first March 2019 to first March 2021

Index or strategie	monthly return (%)	annualized return (%)
MASI	0.04	0.43
Distributors	1.19	15.24
Pharmaceutical Industry	1.53	19.94
Software & Computer Services	3.34	48.30
Proposed strategy	1.77	23.44

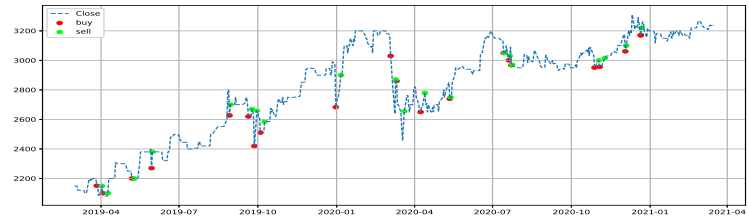

Figure 3 buy & sell moment for LBV stock

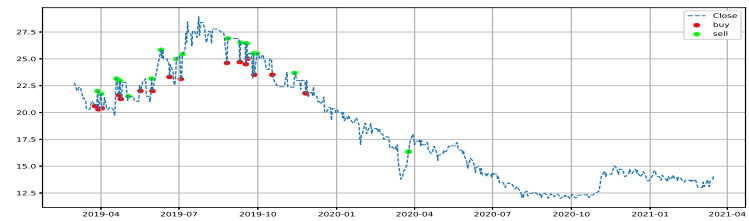

Figure 4 buy & sell moment for MDP stock

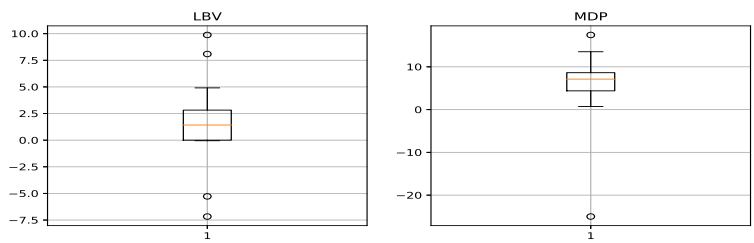

Figure 5 Performance per trad (brokerage fees ignored)

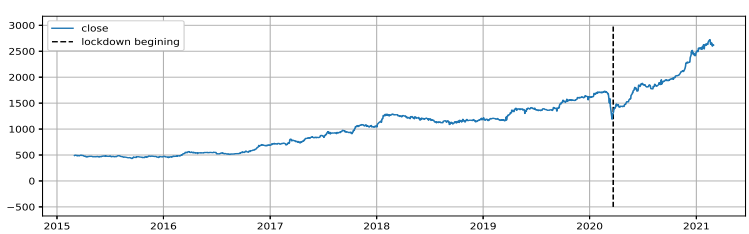

Figure 6 Historical data of Software & Computer Services index

Discussion

Referring to the table 2, we see a contrast in the individual performance of the selected stock. Indeed, the amount invested at the start of the test period decreased by 8.45% for LBV, whereas it increased by more than 55% for MDP. A very strong

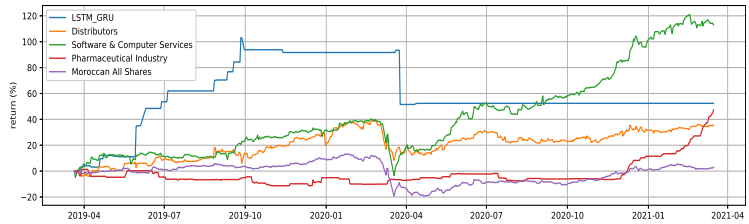

Figure 7 Performance comparaison over time

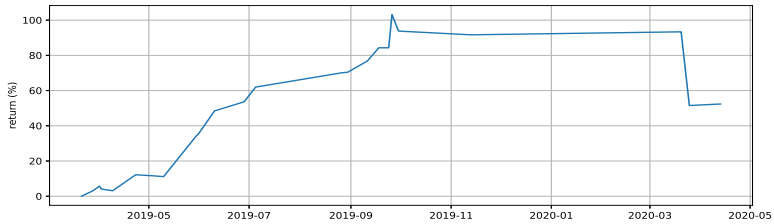

Figure 8 Proposed strategy return over time

observation for both stocks, the win rate ratio is very good, indeed 85% and 93% are quite high indicating that our trade is winning most of time. Fig. 5 helps us better understand the LBV stock's behavior, although the trade is usually a winner (the selling price is higher than the purchase price), the profit margin does not cover the brokerage fee. This helps to understand the poor results. Despite this, the MDP performance (55.78%) largely covers and allows us to achieve a very satisfactory annualized return for our proposed strategy. By comparing the performances from Table 3, the returns of our proposed strategy exceed the returns of all indices except Software & Computer Services index. This index performance has risen dramatically because of the widespread use of remote applications during the COVD19 pandemic. Fig. 6 confirms this finding, the index is at its maximum never observed before during the pandemic phase. To better compare our strategy with the Software & Computer Services index and all other indices, Fig. 7 illustrates the return evolution for our proposed strategy as well as the other benchmark indices over all testing periods. We clearly observe the supremacy of our proposed strategy over the entire part prior covid19 (22 march was the start of lockdown in Morocco). In fact, until April, our proposed method return was near 100%, while Software & Computer Services index return does not exceed 40%. Distribution index return was also around 40%, while the rest of the indices were lower. The COVID19 crisis has halted the evolution of our proposed approach performance, indeed Fig. 8 shows a decrease in total return in March 2020. However, its return has not collapsed during this crisis. The overall result even during the crisis period remains very positive and it is a good point for our approach in terms of crisis management. Notice that pharmaceuticals, distribution, and ICT are among the sectors not impacted by the COVID19 crisis, on the contrary, they are among the sectors that have gained

from it. Despite this, only the ICT performance may outperform the result of our proposed strategy.

Conclusion

This paper proposes a new trading strategy tailored for the Moroccan market, based on the forecasts of the two models LSTM and GRU and the decision rules customized for each stock. The proposed approach allows selecting profitable stocks and creation of portfolio that outperform all indices used as a benchmark except the Software & IT Services index which witnessed an abnormal boom during COVID pandemic time. The proposed strategy is very promising, its performances are overall very satisfactory, future improvements can make it more profitable, especially if applied to a market with more opportunity and more available stocks. In our future work, we will continue to work on deep learning portfolio building techniques, but we will focus much more on prediction of medium and long term horizon, we will also extend preprocessing part by incorporating natural language processing NLP techniques to capture the effect of social media, news and rumors in stock market price.

Acknowledgements
Not applicable.

Funding
Not applicable. This research received no specific grant from any funding agency.

Availability of data and materials
Not applicable. For any collaboration, please contact the authors.

Ethics approval and consent to participate
The author confirms the sole responsibility for this manuscript. The author read and approved the final manuscript

Competing interests
The authors declare that they have no competing interests.

Consent for publication
The authors consent for publication.

Authors' contributions
Y. Touzani has first author role so he performed the literature review, implemented the proposed model, carried out the experiments and wrote the manuscript. K.Douzi has a supervisory role, she oversaw the completion of the work. All authors read and approved the fnal manuscript.

Author details
Computer Lab of Mohammedia LIM ,Faculty of Sciences and Technology of Mohammedia (FSTM) Hassan II university, Mohammedia, Morroco.

References

1. Fama EF. Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance. 1970;25:383–417.
2. Green J, Hand JRM, Zhang XF. The supraview of return predictive signals. Rev Account Stud. 2013;18:692–730.
3. Budiharto W. Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM). J Big Data. 2021;8:47.
4. Yadav A, Jha CK, Sharan A. Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science. 2020;167:2091–100.
5. Samarawickrama AJP, Fernando TGI. A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. 2017 IEEE International Conference on Industrial and Information Systems (ICIIS). 2017. p. 1–6.
6. Nti I kofi, Adekoya A, Weyori B. Predicting Stock Market Price Movement Using Sentiment Analysis: Evidence From Ghana. Applied Computer Systems. 2020;25:33–42.
7. Liu H, Long Z. An improved deep learning model for predicting stock market price time series. Digital Signal Processing. 2020;102:102741.
8. Gilles J. Empirical Wavelet Transform. IEEE Transactions on Signal Processing. 2013;61:3999–4010.
9. Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks. 1995. p. 1942–8 vol.4.

10. Horata P, Chiewchanwattana S, Sunat K. Robust extreme learning machine. Neurocomputing. 2013;102:31–44.
11. Althelaya KA, El-Alfy E-SM, Mohammed S. Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). 2018 21st Saudi Computer Society National Computer Conference (NCC). 2018. p. 1–7.
12. Patel MM, Tanwar S, Gupta R, Kumar N. A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. Journal of Information Security and Applications. 2020;55:102583.
13. Lee SI, Yoo SJ. Threshold-based portfolio: the role of the threshold and its applications. J Supercomput. 2020;76:8040–57.
14. Cao J, Li Z, Li J. Financial time series forecasting model based on CEEMDAN and LSTM. Physica A: Statistical Mechanics and its Applications. 2019;519:127–39.
15. Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences. 1998;454:903–95.
16. Torres ME, Colominas MA, Schlotthauer G, Flandrin P. A complete ensemble empirical mode decomposition with adaptive noise. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2011. p. 4144–7.
17. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–44.
18. Goodfellow I, Bengio Y, Courville A. Deep learning [Internet]. MIT Press; 2016. Available from: https://www.deeplearningbook.org
19. Schmidhuber J. Deep Learning in Neural Networks: An Overview. Neural Networks. 2015;61:85–117.
20. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks. 1994;5:157–66.
21. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997;9:1735–80.
22. Gers F, Schmidhuber J, Cummins FA. Learning to Forget: Continual Prediction with LSTM. Neural Computation. 2000;
23. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:14061078. 2014
24. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:14123555. 2014
25. Jozefowicz R, Zaremba W, Sutskever I. An Empirical Exploration of Recurrent Network Architectures. International Conference on Machine Learning. PMLR; 2015. p. 2342-50.
26. Shewalkar A. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. Journal of Artificial Intelligence and Soft Computing Research. 2019;235–245.
27. http://www.casablanca-bourse.com/bourseweb/Liste-Societe.aspx?IdLink=20&Cat=7 1. The Casablanca Stock Exchange. http://www.casablanca-bourse.com/bourseweb/Liste-Societe.aspx?IdLink=20&Cat=7. Accessed 07 Feb 2021.
28. Cogswell M, Ahmed F, Girshick R, Zitnick L, Batra D. Reducing Overfitting in Deep Networks by Decorrelating Representations. arXiv:151106068. 2016
29. pandas-datareader https://pandas-datareader.readthedocs.io/en/. Accessed 12 Feb 2021.
30. investpy 1.0.2 documentation, (n.d.). https://investpy.readthedocs.io/. Accessed 12 Feb 2021.
31. Team K. Keras documentation: Getting started. https://keras.io/getting_started/. Accessed 13 Oct 2020.
32. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) [Internet]. USA: IEEE Computer Society; 2015 . p. 1026–1034.
33. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980. 2017
34. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014;15:1929–58.
35. Zheng S, Song Y, Leung T, Goodfellow I. Improving the Robustness of Deep Neural Networks via Stability Training. arXiv:160404326 [cs] . 2016

Figures

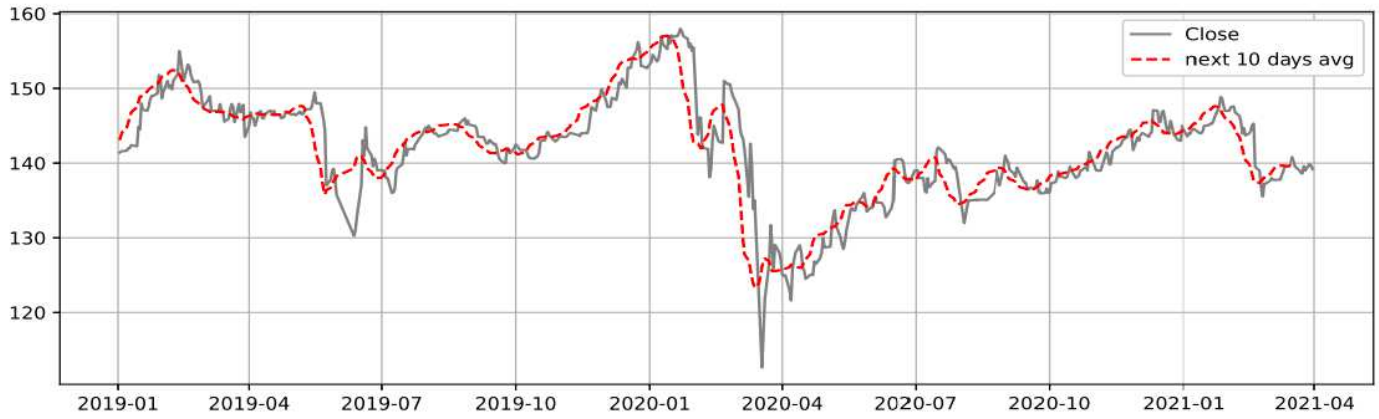

Figure 1

Price with next days Moving average for IAM stock

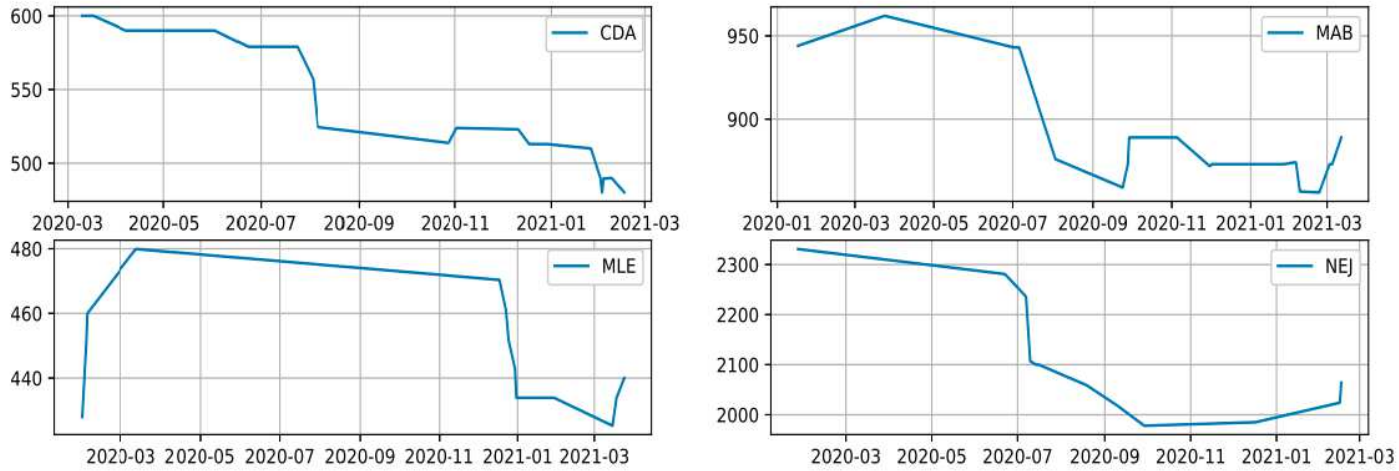

Figure 2

Example of stocks with trading discontinuity

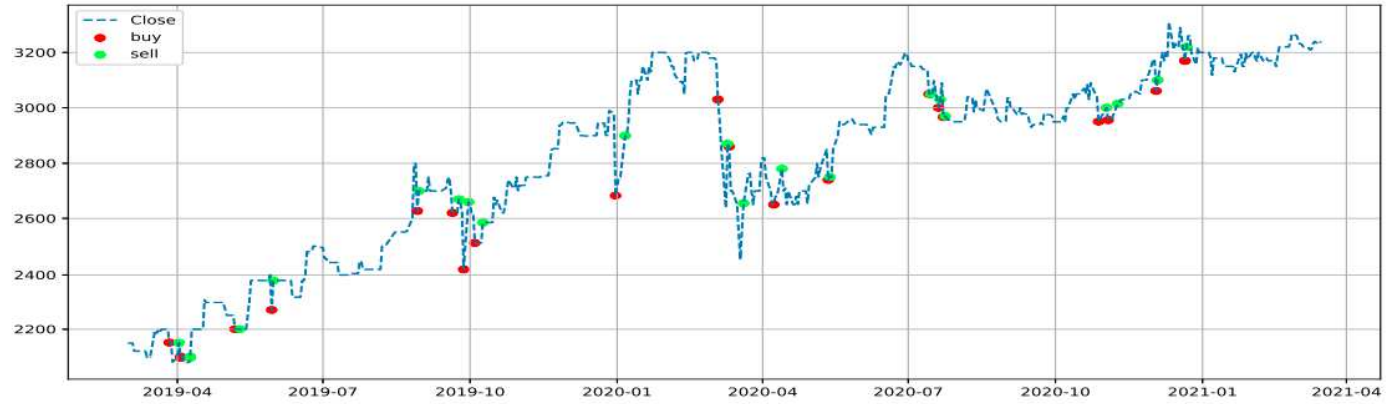

Figure 3

buy & sell moment for LBV stock

Figure 4

buy & sell moment for MDP stock

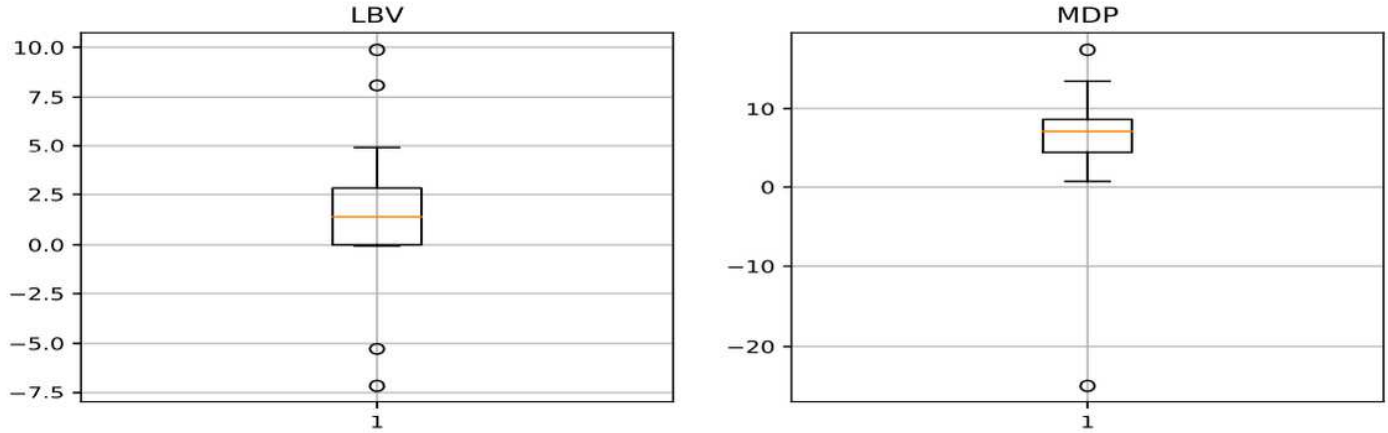

Figure 5

Performance per trad (brokerage fees ignored)

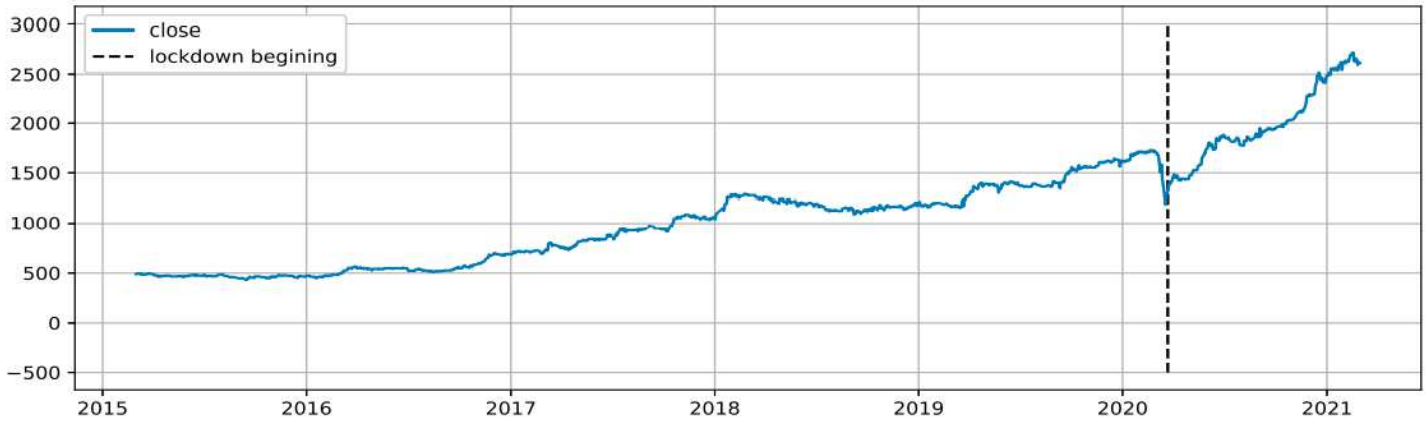

Figure 6

Historical data of Software & Computer Services index

Figure 7

Performance comparaison over time

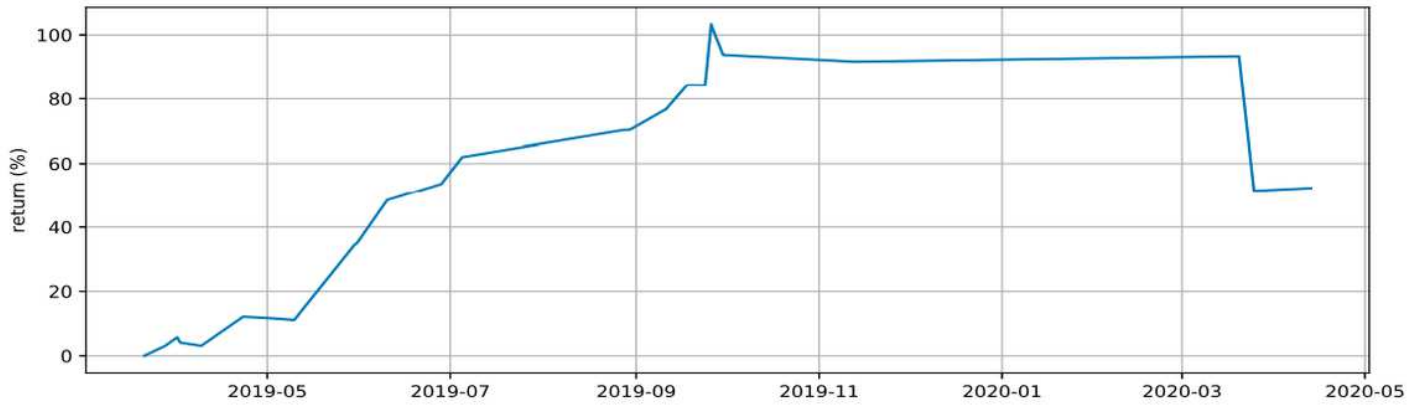

Figure 8

Proposed strategy return over time