Mini-Project (ML for Time Series) - MVA 2024/2025
Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting

Félix Fourreau felix.fourreau@enpc.fr
Mathis Wauquiez mathis.wauquiez@enpc.fr

1 Introduction and contributions

Accurate forecasting of financial time series is the dream of every financial companies as it could represent the ability to make lots of money. However these forecasts are particularly challenging due to the stochastic nature of financial data, influenced by factors like market volatility, politics, and investors' behaviors. Previous statistical models, such as ARIMA (Autoregressive integrated moving average model) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity), often fails to capture these nonlinear dependencies, especially in multivariate scenarios, where many variables influence the prediction target. To address these challenges deep learning method may be they key for improvement of theses forecast.

The paper [2] focuses on feature selection and deep learning for financial time series forecasting. It proposes a two-stage feature selection process combining RReliefF for dimensionality reduction and a wrapper-based method using Extreme Learning Machines (ELM) and a multi-objective optimization algorithm (MOGWO). The selected features train a deep learning framework integrating LSTM, GRU, and ConvLSTM architectures to capture dependencies in time series. An error correction model using empirical mode decomposition and Lasso regression further refines predictions by addressing residual errors.

The objective of this report was to understand the method and the theory behind the paper (as you will soon understand, it was not an easy task...), make an implementation from scratch based on the few incoherent descriptions made by the authors, and fetch similar variables from similar financial indexes to compare the results. We will test both its feature selection model, and its deep learning model to measure by ourselves the performances claimed by the authors. Specifically, we will try to show that this paper was probably written by someone who had no idea what he was doing and that this paper should never have been validated by peer review...

The work was divided as follows. Félix was responsible for fetching similar data from the original paper, opening prices from chinese indexes, but also further information like cash flow per share and book value per share by scraping websites like finance.yahoo.com and eastmoney.com to obtain all the variables mentioned in the paper. He also worked on establishing the reference behind every method used in the paper because some theoretical justification seems most of the time weak at best. Mathis focused on implementing from scratch ,with some obscure explanation from the paper, most of the two stage features selection proposed by the author, he also implemented the deep learning model for forecasting, and interpreted the error forecasting part using LASSO. There was no source code provided with the paper, and no similar implementation to start from, only modules for specific algorithms, such as RReliefF or CEEMDAN.

The paper itself conducted limited experiments on only three financial indexes, with little details and shady results (RMSE on test lower than train). To extend theses experiment and verify the truth behind its claim, we will attempt experiments on other indexes with more variable. However, setting up these experiments is challenging because the final predictions heavily depend on the features selected during the first phase (wich itself . We will try to compare the features we obtained with those specified in the paper and evaluate the significance of feature selection by testing the model on all features.

Improving on the proposed method is hard since the method itself is really complex, however we will try to select better features to improve the results and maybe use PCA on the initial features. We will also tweak the model number or parameters or activation function to try to reach better performance.

2 Method

The proposed forecasting framework is composed of a two-stage feature selection process, a deep learning model, and an error correction mechanism to enhance the accuracy of the forecasting.

2.1 Two-Stage Feature Selection

The financial time series used in the paper contain between 7 and 16 features (time series variables), according to the authors selecting too much features would lead to over fitting. Therefore it is crucial in order to improve model robustness to keep an optimal subset of features. The selection process is composed of two stages:

2.1.1 Filter-Based Method: RReliefF Algorithm

The first stage uses the RReliefF algorithm, a filter-based technique, to assign the relevance of each feature independently. The weight W_F of a feature F is :

$$W_F = \frac{p_{\text{diffP}|\text{diffF}} \cdot p_{\text{diffF}}}{p_{\text{diffP}}} - \frac{(1 - p_{\text{diffP}|\text{diffF}}) \cdot p_{\text{diffF}}}{1 - p_{\text{diffP}}}$$

- $p_{\text{diffF}} = p(\text{diffF} \mid \text{NIs})$: Probability of differences in feature F among nearest instances (NIs).
- $p_{\text{diffP}} = p(\text{diffP} \mid \text{NIs})$: Probability of differences in prediction P among nearest instances (NIs).
- $p_{\text{diffP}|\text{diffF}} = p(\text{diffP} \mid \text{diffF}, \text{NIs})$: Conditional probability of differences in prediction P, given differences in feature F.

Features are then ranked based on W_F, with higher weights indicating greater importance. It is not explicitly said in the paper but it seems that the features with weights under a certain threshold are then removed.

2.1.2 Wrapper-Based Method: Multi-Objective Binary Grey Wolf Optimizer with Cuckoo Search and Extreme Learning Machine (ELM)

The second stage employs a wrapper based approach utilizing a multi-objective binary Grey Wolf Optimizer enhanced with Cuckoo Search (MOGWO-CS) working with an ELM to select an optimal feature subset. The objectives is to find the optimal trade off between a low number of features and an high accuracy.

Grey Wolf Optimizer Dynamics The inspiration for the GWO optimizer comes from the social structures of wolfs pack [1]. The fittest solution is considered as the alpha (α, 2nd β, 3d γ) wolf and the other (ω) this wolf in the search for the global optimum :

Cuckoo Search Integration An adaptive Cuckoo Search operator is then used to enhances the population diversity:

$$X_1 = X_\alpha - A_1 \cdot |C_1 \cdot X_\alpha - X|$$
$$X_2 = X_\beta - A_2 \cdot |C_2 \cdot X_\beta - X|$$
$$X_3 = X_\delta - A_3 \cdot |C_3 \cdot X_\delta - X|$$
$$X_1 = \text{Cuckoo}(X_1, X_\alpha, \rho)$$
$$X_2 = \text{Cuckoo}(X_2, X_\beta, \rho)$$
$$X_3 = \text{Cuckoo}(X_3, X_\delta, \rho)$$
$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}$$

Adaptative Cuckoo Search Algorithm:

```
Parameters: $X$, $X_\alpha$, $\rho$, $\beta$, $\lambda$
Compute $\theta = \left(\frac{\sin(\pi\beta/2)\cdot\Gamma(1+\beta)}{\beta\cdot\Gamma((1+\beta)/2)\cdot 2^{(\beta-1)/2}}\right)^{1/\beta}$
for $j = 1$ to size$(X)$ do
    $s \leftarrow X(j)$
    $u, v \leftarrow$ randn
    $\lambda \leftarrow \frac{u}{(\text{abs}(v))^{1/\beta}}$
    $X(j) \leftarrow s + \rho \cdot (0.001 \cdot \lambda \cdot (s - X_\alpha))$
    $\rho \leftarrow \rho \cdot \rho'$
end for
```

ELM Integration The authors then "train" an Extreme Learning Machine (ELM) that consists of one layer (with undisclosed number of hidden weights) with an activation function (undisclosed as well) (there is no further explanation on what this training consists of, we don't even know what the ELM is trying to predict : is it the opening price of the following day ? The input features of the following day ?).

To summarize, MOGWO optimizes the feature subset by evaluating each candidate using the ELM's predictions error. The paper then aim to find a Pareto-optimal solution, however the authors do not chose the optimal solution but rather a worse solution that has less features as input (There is no further explanation on that)

2.2 Deep Learning Forecasting Model

The deep learning component is constructed from three recurrent units : Convolutional LSTM, LSTM, and Gated Recurrent Unit (GRU : LSTM simplified with update and reset gates), with ADAM optimizer. We won't go any further with the deep learning models as the paper does not implement anything new or original. Let's just note that regarding the convolutional lstm, we don't even know on what does the convolution applies. It may be the time but the reference provided by the author [3] apply the convolution on the features (images), however the authors only keep 1 or 2 features at the end so it doesn't really make sense, furthermore they keep talking about a three dimensional tensor for convolutional lstm which doesn't even apply in our case. Finally the authors don't even provide the kernel size for the lstm...

2.3 Error Correction Model

The final prediction are then refined, residual errors from the deep learning model are corrected using an error correction mechanism involving Empirical Mode Decomposition and Lasso Regression (which mean that the previous model also had to forecast the error on prediction, which is never explicitly stated)

During the training part, the residual error e_t is decomposed into Intrinsic Mode Functions (IMFs) using empirical mode decomposition. High-frequency IMFs are discarded to isolate significant error components. The remaining error components are modeled using Lasso Regression

3 Data

The authors of the original paper obtained their data from eastmoney.com, but we were unable to access the same source as it requires a Chinese phone number for registration. Despite this limitation, we encountered the same issue mentioned in the paper: missing data from the source. The authors used cubic spline interpolation to handle missing values. However, upon examining our dataset, we found that there was only one missing data point (the first entry) from 'Change' which was normal since its a difference from the previous day, there was also data missing from price to book and price to earning ratio, however those value were pretty much constant overtime ($\sigma < 10^-15$) therefore the use of interpolation was useless (the data was constant because the P/E and P/B values are updated every quarter of every year) therefore the use of cubic spline interpolation wasn't very appropriate.

Since we weren't able to get the data from the same source (we used yahoo finance) we had to manually compute the missing features, we quickly came to the conclusion that within the 16 initial features more than half were directly correlated with each other, here are some example :

- **Amplitude**: (Highest Price - Lowest price)/Closing Price
- **Change**: Closing Price(t)-Closing Price(t-1), useless with the use of an lstm
- **Turnover**: $Volume \times Opening\ Price$

However by plotting the correlation matrix

Original Features	**Directly Correlated Features**
Opening Price (F1)	Closing Price (F4)
Highest Price (F2)	Change (F5):Closing Price(t)-Closing Price(t-1)
Lowest Price (F3)	Price Change Ratio (F6): F5 divided by the price of the previous day
Closing Price (F4)	Turnover (F8): Closing Price $\times$ Volume
Volume (F7)	Total Market Value (F11)
Turnover Rate (F9): F7 over a period divided by the total share issued	Circulation Market Value (F12)
Price-to-Earnings (P/E) Ratio (F13): Stock price divided by earnings per share	Amplitude (F10): (F2-F3)/F1
Price-to-Book (P/B) Ratio (F14): Stock price divided by net asset per share	

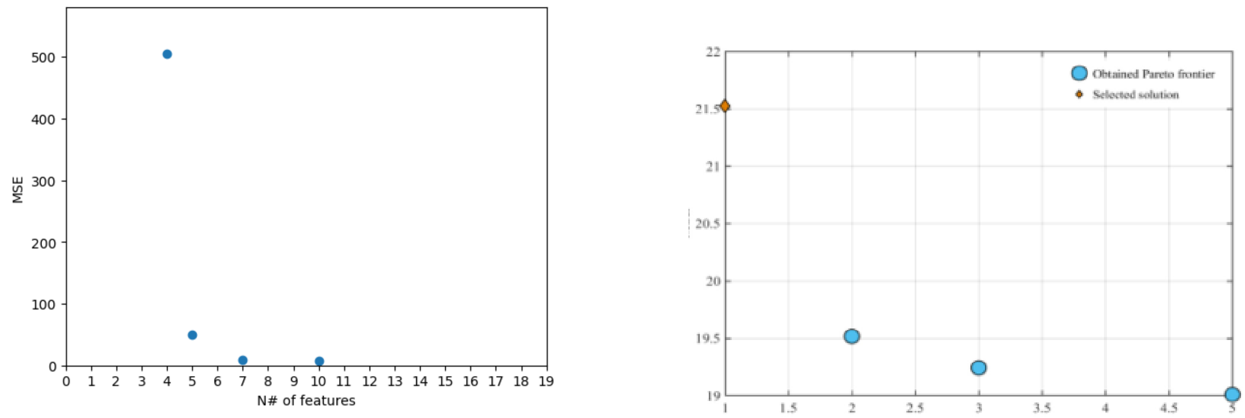
Price-to-Cash-Flow Ratio (F15): Stock price divided by cash flow per share
Price-to-Sales Ratio (F16): Stock price divided by sales per share

We found also that Opening price, Highest Price, Lowest Price and Closing Price are directly correlated (because they follow the same index) Therefore we could have taken Amplitude instead of highest and lowest Price, however we never touched Opening Price as its the output that we are trying to forecast. We did try to follow some feature extraction method from the course however it did not lead anywhere and the two fundamental properties: stationarity and ergodicity were never verified.

Finaly, it's worth noting that the paper did not perform additional feature engineering but instead manipulated the 16 original ones (without normalization)

4 Results

The model performance wasn't what we hoped or believed we will have from the paper promises. We tested the model on three real-world stock market datasets (from 2010 to 2020) obtained via Yahoo! Finance: (1) DIJA (daily), (2) SZFI (daily), and (3) SZCI (daily). Our main objective was to replicate the result presented in the paper. Therefore we used as input the 16 original features used by the author the first features selection method gave :



(a) Our pareto frontier for Dija Dataset (b) The paper's pareto frontier.

Figure 1: Pareto frontier from the paper for Dija Dataset.

We obtain comparable results, however we still did't understand why they chose results with highest MSE (Here they whose the one with fewest features but not everythime, see Fig 4 paper)...

We then attempted to train the deep learning model, however with the number of parameters provided by the paper (around ($\sim$ 10 million) we got direct over fitting on our data (train loss instantly under $5x10^{-3}$) while validation loss struggle to decrease.

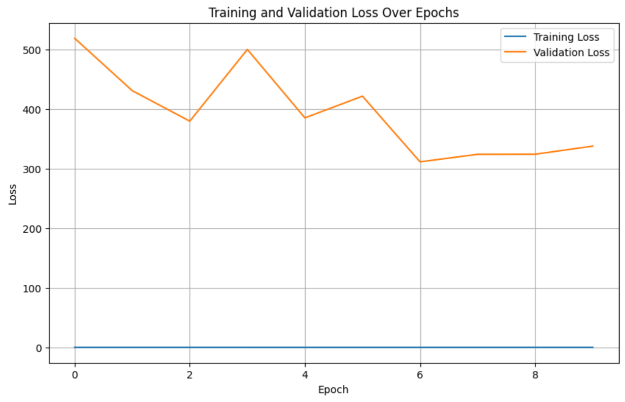


Figure 2: Loss over Epochs on deep learning model (LSTM combination)

Despite multiple attempts, all of our trained models showed severe overfitting or no learning at all. We tried different settings in desperation but they were too many variable,we didn't even know if we had the correct implementation for the convolutional lstm or even if we were predicting the good outputs (as we had no indications).

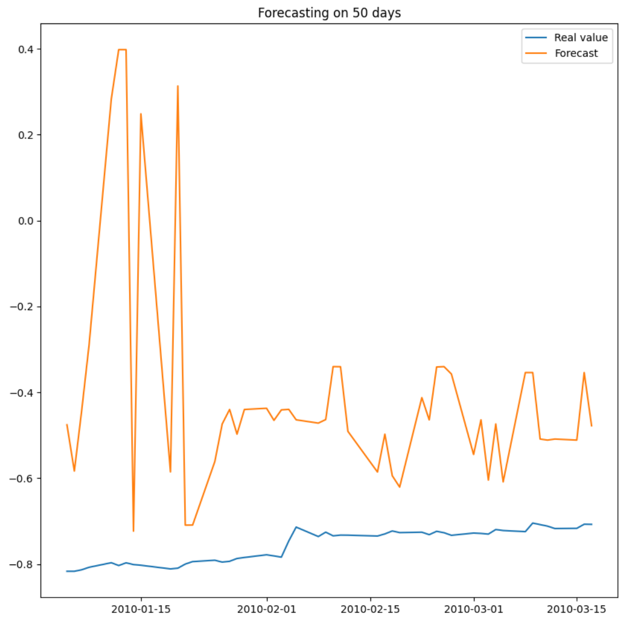


Figure 3: Forecasting on djia on 50 days

Discussion regarding the paper

1. **Overfitting with the Proposed Parameter Scale.**
   The original paper uses $\sim$ 10 million learnable parameters, which seems excessive for our time-series data. This is confirmed by a large MSE difference between training, validation, and testing sets: we obtain a RMSE of 0.2989 for the training set, whereas it is 3.0170 for the validation set.

2. **Unspecified Design Choices.**

Several critical design elements in the original paper—such as convolution kernel size, the exact parts of the data to convolve, and the "shady" feature selection steps—were not fully defined. We tried various kernel sizes, and different artificialy selected feature subsets with no success in achieving robust performance.

3. **Implausible Results in the Original Paper.**

   - *Lower Test Loss than Train Loss* (Table 9):
   - $R^2 \approx 1$ *Despite High Loss* (Table 8):
   - *Learning Range in Figure 11*: Their linear learning range from 0 to 1 is unrealistically large for real-world data, leading to explosive training signals in replication attempts.
   - *Reconstruction Error "Cancellation"*: Forecasting the error to "cancel it out" did not yield the remarkable performance improvements suggested.
   - *No Normalization* (Table 3): The paper strongly implies no normalization was done. It is hard to believe they got theses good results without proper normalization

Qualitative Assessment

Beyond the evident inconsistency of the paper, the core hypothesis—that a large deep architecture with two-stage feature selection plus an error-correction module can simultaneously capture short-term fluctuations and overall market trends— doesn't seem very likely. The first time series course showed that even if the LSTM results were good in appearance, it wasn't really the case as the impressive results were just a shift from the original data.

Conclusion

While working on this paper, we encountered several challenges, related to the lack of methodological rigor and transparency. Some results raised concerns about their validity, seeming implausible. We are not convinced that the authors really grasped all the subjects in their paper. We do not understand how this article passed the peer-review process, and underscore the importance of maintaining high standards for academic contributions.

References

[1] Seyedali Mirjalili, Shahrzad Saremi, Seyed Mohammad Mirjalili, and Leandro dos S. Coelho. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47:106–119, 2016.

[2] Tong Niu, Jianzhou Wang, Haiyan Lu, Wendong Yang, and Pei Du. Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications*, 148:113237, 2020.

[3] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.