Time-Series Foundation Models (TSFM) in Finance

**Project overview:**

For the purposes of this project, we will use the state-of-the-art TSFMs (Lag Llama and TimeGPT) to predict financial time-series data. TSFMs will be pitched against the standard time-series prediction models (benchmark) such as ARIMA and the simple autoregressor.

**Project Aim:**

Main aim of this research is to find out whether TSFMs are good at predicting financial time-series data. Secondary aims are to discover how TSFMs’ performance can be improved and to provide a comprehensive analysis of the results – giving insight into what type of data can TSFMs be used for, and which configuration they should be used in.

**Current progress:**

1. Implemented Lag Llama and benchmark models
2. Implemented the data gathering and preparation functionalities
3. Implemented the fine-tuning framework for Lag Llama
4. Implemented the framework for the evaluation of the models
5. Implemented the evaluation of the predictions
6. Implemented the Time-series cross validation framework
7. Implemented the interactive results visualization capability

**Future steps (in no particular order)**

1. To run the experiment on different types of financial data across different configurations of the Models
2. To implement TimeGPT
3. (possibly) to implement more benchmark models
4. To refine the results visualization capability
5. To implement the results recording and aggregation functionality
6. Thorough analysis of the results

The experiment was run in a very limited parameter space (context\_length = [32, 64, 128], type\_of\_data = [“stock”, “return”], frequency\_of\_data = [“minutely”, “hourly”, “daily”]), covering a very small time period (10 single-day predictions, of the single stock: Apple) and here are the **preliminary results:** (Note: due to very small sample size (10 per each experiment) and very low number of experiments (18), **median** results are used in tables)

A screenshot of a computer

Description automatically generated

1. Overall, autoARIMA and simple autoregressor models make the best predictions in terms of traditional regression metrics, however, fine-tuned Lag-Llama outperforms all in terms of directional accuracy (mda)

A screenshot of a computer screen

Description automatically generated

1. Analysing performance on raw stock prices vs returns, autoARIMA and autoregressor again perform better in terms of traditional regression metrics and however they are on par in terms of directional accuracy. All models perform better on stock prices than they do on stock returns.

A screenshot of a computer

Description automatically generated

1. Looking across different context\_length (training size) values, Lag Llama and fine-tuned Lag-Llama seem to benefit most with the context length of 128 where they outperform autoARIMA and autoregressor in terms of directional accuracy. For other values of context\_length, fine-tuned Lag-Llama is on par with the autoregressor in terms of directional accuracy.

A screenshot of a computer screen

Description automatically generated

1. Looking at different frequencies of data, traditional models again outperform Lag-Llamas, however, fine-tuned Lag-Llama is on par with autoregressor in terms of directional accuracy with hourly data, and both Lag-Llamas beat other models in terms of directional accuracy on minutely data.

Based on these preliminary results, we can make some **preliminary conclusions:**

1. Both raw and fine-tuned Lag-Llamas cannot compete with traditional time-series prediction models in terms of traditional time-series prediction metrics. However, In majority of cases, fine-tuned Lag-Llama exhibits superior ability to predict the direction of movement of the time series (directional accuracy).
2. The evidence for the thesis that fine-tuning Lag-Llama improves its prediction capabilities is inconclusive as its prediction quality seems to be on par with raw Lag-Llama. However, in all the cases fine-tuned Lag-Llama had either better or same directional accuracy than the raw version of Lag-Llama – meaning that the model has some potential to capture hidden patterns in the fine-tune data which are repeated in the test-data. This avenue will be explored in more detail in the future
3. Fine-tuned Lag-Llama seems to perform best (in terms of MDA) with stock price data of daily frequency and context\_length of 128.

**Discussion:**

* Possible reason why Lag-Llama performs so poorly is because of the way it tokenizes the time-series data. Lag Llama’s predictions are too dependant on “lags” which are certain historical values at a fixed time-distance from the time it is trying to predict (for example: when predicting daily data at time **t**, Lag-Llama is using data at times **t-1**, **t-7**, **t-14**, **t-30**, as well as the data about day-of-the-month, day-of-the-week and day-of-the-year whether or not these data are important). ARIMA is superior to Lag-Llama in this way because it doesn’t just blindly take fixed data when making predictions but rather it learns which lags are important for prediction and which aren’t. However, and advantage that Lag-Llama holds over ARIMA is that it is able to uncover more complex, non-linear relationships between previous and future data. This ability manifests iteself in the fact that fine-tuned Lag-Llama beats all other models in terms of directional accuracy even while having this handicap that I mentioned.
* The flaw in these results is that they aren’t statistically significant. They were run only on 18 different configurations (different combinations of context\_length, frequency and type\_of\_data parameters) and furthermore, all these time-series belong to a single stock: Appe and in each configuration, only 10 consecutive single-period predictions were made. To get more reliable results, the experiment should be done with a lot more predictions over multiple different time periods and for different stocks (not just Apple). Another flaw is that the Fine-tuned Lag-Llama was fine-tuned only once per each configuration whereas in theory it should be fine-tuned before each prediction. The reason why this experiment was executed in such a limited way is due to computational constraints. Running the experiment in full capacity would take multiple days on my PC and is simply infeasible. Currently I’m thinking of finding ways to execute the code faster by using cloud computing or to restrict certain parameters to reduce the scope of the experiment.
* To combat the statistical insignificance of the results, I used the **median** metrics for the tables – as median is a more robust metric than the mean. However, using the median isn’t perfect way to display the results as it doesn’t take into account the variability of the results (which is quite large in our case). However I still decided to use it as using the mean wouldn’t make sense as some metrics are on a completely different scale due to diferent frequencies and types types of data so summing them up and dividing wouldn’t make sense.

**Future work:**

* It would be worth to see an adaptation of Lag-Llama with more flexibility when it comes to time-series tokenization.

**Example results of Time-series cross validation (Apple daily Close stock price prediction):**

EVALUATION

A blue square with black lines

Description automatically generated

A blue square with black lines

Description automatically generated

VISUALIZATION

A graph with different colored lines

Description automatically generated

SUMMARY

A screenshot of a computer screen

Description automatically generated