Thesis

Jeremias Lukas Aechtner

May 29, 2025

acknowledgment

This is my acknowledgement

Contents

1 Introduction

ML has become a very important technology in time series forecasting over various fields where researchers apply it to real world challenges. These fields include Healthcare[1], Energy[2], Agriculture[3], Climate & Weather[4] and financial data. Especially in financial data people have a huge interest in making accurate forecasts about developments of for example companies and stock prices given the huge potential for profit in trading.

In this paper we want to evaluate different Machine Learning models against each other based on their performance of forecasting different stock prices and indices. We therefore observe the current literature in this field:

2 Literature Review

The paper [5] from Kumbure et al. (2022) showed in a comprehensive review the technological development in the field of stock forecasting. The review includes 138 articles that have been published between 2000 and 2019. It looks on machine learning techniques that have been applied, features that were considered for training and metrices with which results have been measured. Like that they propose a differentiation of features into 4 categories which are "Technical Indicators", "Macro-Economy", "Fundamental Indicators" and "others". They explain the importance and use frequency of these features. In section 5.4.8 that paper[5] illustrates what families of machine learning models have been applied to the task of forecasting stock prices. It shows that at least since the year 2000 Artificial Neural Networks (ANNs) in various forms were experimented with excluding Deep Learning models that are listed separately. That differentiation shows that deep learning models have especially been investigated heavily in the more recent years with the first reviewed paper applying them in 2017.

One representative of Deep Learning is mentioned very regularly in research – Long Short-Term Memory (LSTM). The paper [6] by Matharasi et al. (2025) evaluates two models against each other which are LSTM and XGBoost. Both delivered promising accuracy in forecasting AMD inc. stock data. While the LSTM model delivered slightly better accuracy in forecasts, the XGBoost model could excel in computational efficiency. Both models have been shown to have outstanding predictive performance when forecasting certain stock or index data. Teixeira, D. M., & Barbosa, R. S. (2025) have conducted another experiment [13] that shows their effectiveness in capturing the complexity of stock prices. They evaluate the models LSTM and XGBoost as well as other architectures such as GRU, CNN, RNN, but also different hybrid combinations of these like LSTM + CNN, RNN + GRU, GRU + CNN, LSTM + GRU, and RNN + LSTM. The study trains all of these models on historical stock data from Apple inc. and comes to the result that the standalone models of GRU, LSTM and XGBoost perform the best for the Apple inc. stock data. This is a little counterintuitive since one might think that hybrid models usually outperform their standalone component models since they might merge strengths of both models but as evidenced that is not generally the case. Under section 5 they discuss that the performance depends furthermore on the timeframe of the training data. They state that the gap of the performance between the models is smaller for a shorter time frame of training data. They state that the reason for that is the lower complexity of that time frame with fewer large amplitude oscillations. When choosing a longer time frame for training with a more complex data course, the three earlier mentioned models GRU, LSTM and XGBoost outperform the other models. Another paper [7] by Li, Siyuan. (2024) investigates the performance of a Kalman Filter and different types of LSTM models such as a single layer LSTM, a stacked LSTM, a bidirectional LSTM and a hybrid convolutional neural network (CNN)-LSTM. From those the bidirectional LSTM performed best for low-volatility stocks and the CNN-LSTM model performed best for high volatility stocks. Here the hybrid CNN-LSTM model in fact outperformed the standalone LSTM model. XGBoost models as mentioned in the two papers above seem to have a reasonable performance in forecasting stock prices under certain circumstances. This is evidenced by another experiment [10] by Aiyegbeni Gifty, & Dr. Yang Li. (2024) that was conducted to compare the performance of LSTM, ARIMA and XGBoost models. They also come to the conclusion that the XGBoost model outstands in performance under certain circumstances. They trained and evaluated the models on Google inc. stock data. This again emphasizes the impact that the data has on the performance of each model and it shows as the other studies mention as well, that certain models have advantages in dealing with certain characteristics in data like volatility making the choice of a suitable model very much not trivial. They also emphasize the importance of hyperparameter tuning. Like that they could reduce the Mean Absolute Error (MAE) from 17.63 without hyperparameter tuning down to 15.98 and the Root Mean Squared Error (RMSE) from 30.24 down to 27.34.

While the mentioned approaches of deep learning and ensemble methods have been looked into particularly heavy in the last years as shown in the previously mentioned papers and under section 5.4.8 of paper [5], the new transformer based architecture has been proven effective in forecasting stock prices. The paper [14] by Yao, Y. (2025) shows its superiority over the previous mentioned models such as XGBoost, LSTM and CNN under certain circumstances.

The three key statements are:

1. LSTM, XGBoost, CNN are usually the best models for our forecasting task.
2. Whether a hybrid model performs better than the standalone models it consists of is not trivial
3. Their performance depends strongly on the characteristics of the data. For example whether it is highly or less volatile.

3 Research Question

From these papers we can see that currently there are a lot of comparative studies done, which aim to evaluate different Machine Learning algorithms against each other based on different data and hyperparameters. Many of the reviewed papers mention the importance and the impact that the data has on the performance of the models. Like that different data with different characteristics lead to different results. That reveals a potential research gap, because since a picture of the performance of single machine learning models based on volatility of the data seems to be taking shape, we couldn't find any accurate evaluation of that. The result we aim for in this study should be a comprehensive analysis that shows the performance of each model based on different volatility. We expect each model to have a certain degree of volatility for the input data at which it performs best and we expect to see the model perform worse for more or less volatile data than that.

4 Machine Learning

5 Methodology

In this part we define our experimental setup that we use for this study

5.1 Model Selection

our Models are LSTM, XGBoost and CNN for time series forecasting

5.2 Volatility

Volatility is known as the degree to which data disperses over time or in other words “Volatility is a statistical measure of the dispersion of returns for a given security or market index. It is often measured from either the standard deviation or variance between those returns.”[15]. So that means that volatility has to be measured to deliver meaningful results. The most common way to measure the volatility of stock courses in finance is by calculating the standard deviation or more precise the sample standard deviation of logarithmic returns [16]. To do so, we first need to calculate the logarithmic returns. We choose to calculate logarithmic returns instead of simple returns. We can understand why we choose the logarithmic return over the simple return by looking at a simple example.
We calculate the simple return with...

$$r_i = \frac{p_i - p_j}{p_j} \tag{5.1}$$

where...

r_i = stands for the current data point’s simple return
p_i = stands for the current data point’s value
p_j = stands for the previous data point’s value

Let's say we have a stock price development for a company X of the following...

5.3 Stock Selection

5.4 Data Preprocessing

5.4.1 Feature Engineering

5.4.1.1 Relative Strength Index (RSI)

5.4.1.2 Rate of Change (ROC)

5.4.1.3 Moving Average Convergence Divergence (MACD)

5.4.1.4 Stochastic Oscillator

5.4.1.5 Commodity Channel Index (CCI)

5.4.2

Introduction to my thesis

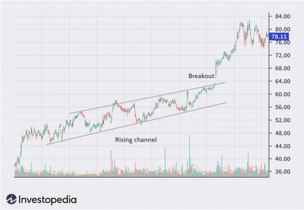

Figure 5.1: Trading Strategy

One trading strategy is channels pictured in 5.1

5.4.3 Formula

sdsd

5.4.3.1 subsub

sdsd

LaTeX is the program we are using right now it is good to write

$$E = mc^2 \tag{5.2}$$