Department of Computer Science

Submitted in part fulfilment for the degree of BEng.

Can Machine Learning Improve on Standard Portfolio Construction Techniques?

Laura Silaja

Version 1.1, 2022-May

Supervisor: Dimitar Kazakov

Acknowledgements

I would like to thank all my friends and family for supporting me and letting me enjoy university life to the fullest. I would also like to thank my supervisor for his continuous support and guidance for this project.

Contents

List of Figures

List of Tables

Executive Summary

With inflation rates rising and traditional money saving methods providing unsatisfactory interest rates, we end up with money we posses depreciating in real-world value. To prevent this loss of capital we look at alternatives such as the stock market where we construct a portfolio of assets to invest the money in which provides us with a return on our investment that allows us to retain and even increase our capital. In order to construct a portfolio, we have to do portfolio selection where you decide which assets to include in your portfolio and portfolio optimisation where given a set of assets, you decide how much of the total sum of money to invest in each asset. In this paper, we focus on the portfolio optimisation aspect in order to answer the question of, how do we decide the amount of money that should be allocated for each asset in a given set of assets?

The standard technique of doing portfolio optimisation as shown by Markowitz [1] is the Mean-Variance (M-V) Optimisation which optimises the maximum return on our investment for a given level of risk. When implemented in practice, the standard optimisation technique has problems as seen in the studies [2], [3] which we want to address. We can try and improve upon the Markowitz Optimisation using machine learning which allows you to use algorithms to build models that learn from data instead of relying on a predetermined equation as a model [4]. This brings us to the aim of this paper which is to answer the question of, can machine learning improve upon standard portfolio construction techniques?

Specifically, we look at 3 machine learning models, namely, the Hidden Markov Model (HMM), Bayesian Linear Regression (BLR) and a deep neural network that uses Long-and-Short-Term Memory (LSTMs). They represent 3 different approaches one can take to improve the M-V optimisation. The first method uses the HMM to predict future returns in an attempt to improve the estimation error shown by Marakbi [3] in the return input to the M-V optimisation. The second method uses Britten-Jones [5] work in order to have the BLR coefficients predict the optimal portfolio's weights that Markowitz Optimisation provides. Weights in this case refer to the percentage of the overall investment in a portfolio that an individual asset represents [6]. The final method uses the state of the art approach from the study by Zhang [7]. It works by using a LSTM neural network to directly optimise the Sharpe ratio for which the portfolio with the highest Sharpe

ratio value corresponds to the Markowitz Optimisation's optimal portfolio.

In addition to the 3 machine learning models, one of the two benchmarks we need to create is M-V optimisation on historic data in order to be able to compare the machine learning models to the standard optimisation technique. To establish a baseline performance, we create the second benchmark called Equal Allocation (EQ) that doesn't use portfolio optimisation and instead it just equally invests in all assets in a portfolio. Then we would use the EQ benchmark to compare against the M-V benchmark. To go about comparing the benchmarks and machine learning models we use statistical hypothesis tests in order to see 2 things. The first one is to see if the models made any statistically significant improvements to the standard technique or even no portfolio optimisation at all. The second one is to see if the standard optimisation technique makes any statistically significant improvement to doing no optimisation at all.

We can see from the results of the statistical hypotheses tests that we lack sufficient evidence to show any statistically significant difference between the models and benchmarks. This means that the machine learning models used in this paper have failed to improve upon the standard portfolio construction technique. While our paper has failed to demonstrate any improvements, there is literature such as Zhang [7] for LSTM models with direct optimisation of the Sharpe ratio that provide evidence that LSTM models do improve upon the standard M-V optimisation. Even between the 2 benchmarks, there was no significant difference found even though M-V benchmark outperformed all other models and benchmarks. So while there are differences between the model and benchmark returns, they are similar enough that they could come from the same distribution and any observed differences can be attributed to random chance. Additionally, none of the models or benchmarks beat the inflation rate for the UK [8] or USA [9] during the testing period and the BLR, LSTM models and the EQ benchmark had negative returns which means they lost money in addition to the money lost due to inflation.

What these results tell us is that there is no simple and full-proof method of doing portfolio optimisation. One approach isn't guaranteed to work on all market conditions as certain portfolios are specialised for specific conditions. This is due to varying performance of different asset types which was seen by the commodity assets in our test period rising in price when the stock and bonds assets fell in price. There are principles that work when trading in the stock market, but people are greedy and want to make as much money as possible in the shortest amount of time. This makes them vulnerable to misinformation which can cause them to lose money and therefore makes them financially worse off which is a social and economical issue that papers like ours try to combat in order to make things better for everyone.

1 Introduction

With inflation of the UK consumer price index reaching 4.9% in 2021-22 [8] and saving accounts for banks like Barlays having less than a 1% annual interest rate means money being kept in saving is depreciating in real-world value due to inflation. The alternative, for people looking for a way to save money is to turn to investments and financial markets such as the stock market. The simplest long term trading strategy is the classical buy-and-hold strategy where you simply buy an asset and hold it for a long period of time [10]. It works due to the long stock market tendency to accumulate value in the long term even if there are short term loses. Stocks also allow people to grow their investments long-term due to dividends giving them money to reinvest back into the asset.

The question now becomes what assets do we trade? We can find massive lists of every kind of asset imaginable, from basic commodities like water to government bonds. The investor could simply pick a couple of indices from each asset group to construct a portfolio of assets that they believe would perform well and have proper diversification. This still leaves us with the problem of how much of each asset do we trade with? This would require some sort of an optimisation strategy to solve leading us to an area called portfolio optimisation which is the focus of this paper.

1.1 Portfolio Optimisation

The most popular and still used standard portfolio optimisation is the Mean-Variance (M-V) optimisation that Markowitz [1] defines as optimising for maximum return on our investment for a given level of risk. This is because people want to earn as much money possible without undesirable speculation that increases the risk of loosing money. There are optimisations with other objectives too, like the Risk Parity Optimisation that uses risk to determine the allocation in an investment portfolio [11].

The standard M-V optimisation has problems associated with it, such as suffering from a drawback in estimation errors in the return vector and covariance matrix as seen in this paper [3]. The paper focuses on covariance estimation improvement, while our paper for one of the machine learning models will focus on improving the expected return part. We do this because we can't expect the trends, changes and societal/economic developments of the present day to mirror the past. This doesn't negate the usefulness a M-V portfolio based on historic data would give as a benchmark for comparison against the other models and as such we'll be creating one in this paper. Though, the absolute baseline benchmark portfolio would still be an Equal Allocation (EQ) portfolio which gives each asset in a portfolio equal importance. The EQ benchmark would be used in order to check that the M-V benchmark is actually better just saying all assets in the portfolio are equally valuable and not doing any sort of evaluation of assets against each other.

1.2 Machine learning

Machine learning uses data to create predictions without explicitly being programmed to do so. Therefore a lot of statistical models such as a Linear Regression can be counted as a machine learning model. When using machine learning in M-V optimisation, there are three main ways one can approach improving the portfolio optimisation.The first one being predicting the data that goes into the M-V optimisation [3]. As mentioned previously, it's the return vector and covariance matrix that we can try to improve. Our paper will focus on improving the expected return parameter and do this by using a Hidden Markov Model (HMM) to predict future returns. The next method uses Zhang's approach [7] which is a direct optimisation of a performance measurement called the Sharpe ratio in a deep learning model due to the fact that optimal portfolio we are looking for in a M-V optimisation is one with the highest Sharpe ratio [12]. The last method would be to define the model in such a way that it directly corresponds to the Markowitz's portfolio weights as it does with ordinary Least Squares Regression (OLS). This works due to a close equivalence between the M-V Optimisation and OLS as shown by the work of Britten-Jones [5].

In general, the aim of this project is to apply multiple machine learning techniques to improve M-V Optimisation asset allocation and then compare, discuss and evaluate the models. Techniques such as the HMM will be used to predict the future expected return, then use M-V optimisation to get the optimal portfolio. Bayesian Linear Regression (BLR) coefficients will also be used to get the optimal portfolio. The last model will be a deep neural network that will use the Sharpe ratio to directly optimise for the optimal portfolio that M-V optimisation gives. This will cover the 3 core approaches for M-V optimisation. We'll be then performing statistical analysis on the resulting portfolio returns, comparing each model performance and then evaluating all return, underlying asset and portfolio weight data.

1.3 Performance Measurements

We will need a way to compare the models created.To do this, there are numerous performance measurements that can be used. The gold standard for Markowitz Optimisation is the Sharpe ratio defined by Sharpe [13] due to the fact that the optimal portfolio in M-V optimisation would be one with the highest Sharpe ratio [12]. Sharpe ratio calculates the risk-adjusted return by measuring the increase in the expected return per unit of additional standard deviation [14]. Standard deviation is also referred to as risk.

Other good performance measurements include Calmar ratio also known as the Drawdown ratio which uses a different type of risk than Sharpe ratio [15]. Calmar ratio calculates risk-adjusted return by taking an annual rate of return of an investment divided by Maximum Drawdown (MDD) [16].

We can also purely look at risk using MDD, which is a indicator of downside risk over a specific time period [17]. Risk containing both upside and downside risk is called volatility and in modern times is taken as standard deviation instead of variance [18]. We can also take a look at just returns instead using Compound Annual Growth Rate (CAGR) which gives us a yearly rate of return, however it assumes it is equal for all years included in the calculation [19].

In the results, we would be looking for a portfolio that would have a high Sharpe/Calmar ratios and CAGR. It would also have a low MDD and volatility. We expect that the machine learning based models will beat the benchmark and regular historic data based M-V Optimisation.

1.4 Summary of Goals

1. Create a M-V optimisation on historic data as a benchmark to compare the machine learning models against.
2. Create an EQ portfolio benchmark to check if M-V benchmark or any other model performs better than it.
3. Create a HMM, a BLR model and a deep neural network model that improve upon the M-V benchmark.
4. Calculate the Sharpe ratio, Calmar ratio, CAGR, volatility and MDD of each portfolio to compare performance.
5. Evaluate model portfolio weights, returns and the asset data used.

2 Literature Review

2.1 Portfolio Theory

Portfolio theory is a wide-ranging topic referring to the analysis and management of assets, portfolios and strategies for their trading. It encompasses investment management, measurements of performance, portfolio optimisation and more. The main focus in this literature review is portfolio optimisation which we can define as the process of selecting the best portfolio, out of all the ones being considered, according to a given objective. An example of this objective according to Modern Portfolio Theory (MPT) would be to maximise expected return for the given cost of risk [20].

2.1.1 Modern Portfolio Theory

The ground work for MPT was laid by Harry Markowitz in his 1952 paper "Portfolio Selection", [1] where he mathematically formulated and discussed the concept of diversifying a portfolio using a risk-return trade-off. The idea of risk diversification in a portfolio depends on how correlated the assets with one another as, for example, two airline companies would both be highly sensitive to rises in fuel prices, while that wouldn't impact a food processor as much as a temporary labour shortage would. Therefore the two less correlated assets when included in a single portfolio would provide a diversification effect mitigating the overall risk of a portfolio by offsetting the risk of individual stocks with those of other stocks. In general, Markowitz's theory solves a M-V optimisation problem by finding the optimal combination of assets that minimises expected portfolio variance for a given level of expected return. This optimal linear combination is what we call an Efficient Frontier and can be represented in a graph as seen in figure 2.1 with expected returns against volatility (risk). M-V optimisation suffers heavily from estimation error as seen in [2] and as such many attempts to improve risk calculation and the expected return estimation have been made.

Jack L. Treynor in 1962 [21], William F. Sharpe in 1964 [22] John Lintner in 1965 [23] and Jan Mossin [24] in 1966 attempted to improve upon

Figure 2.1: E-V rule graph

Markowitz's work by introducing capital asset pricing model (CAPM) to get better expected return estimation. The core idea of CAPM is that it is "based on the idea that not all risks should affect asset prices" [12, p. 3]. As discussed by Sharpe [22], CAPM differentiates types of risk between systematic risk and unsystematic risk. Systematic risk describes the risk inherent to the market as a whole and that correlates all the assets in a market. Since all assets are affected, this type of risk cannot be diversified away. Unsystematic risk on the other hand is risk that is individual to an asset or an asset group, and as a result can be mitigated through diversification of assets. Therefore unsystematic risk can be considered not relevant due to it being eliminated in a well diversified portfolio. Overall, CAPM describes how systematic risk relates to expected returns and allows us to calculate the rate of expected returns for an individual asset or portfolio.

Perold's [12] work shows that CAPM adds to Markowitz's idea of an Efficient Frontier to create an Optimal Risky Portfolio which also corresponds to the portfolio with the highest Sharpe ratio. It does this by introducing a risk-free asset that is usually modelled as a US treasury bond, and introducing a Capital Market Line (CML) which is a line tangent to the Efficient Frontier curve where the tangent point represents the Optimal Risky Portfolio. The

CML takes into account the of asset's sensitivity to systematic risk which we nowdays call β as well as the expected return of the market and expected return of a theoretical asset with zero risk (risk-free asset).

There are alternative asset pricing models like the single index model (SIM) developed by Sharpe in 1963 [25] and multi-factor models like the Fama-French Model introduced by Fama and French in 1992 [26]. SIM uses only a single factor to represent the effects of systematic risk of all stocks, while Fama-French model uses three factors. Either way, both of these models and CAPM focus on creating models that improve the estimated returns of assets to generate better parameters for input to the M-V optimisation in order to reduce the estimation error shown by Marakbi [3].

The biggest issue with the CAPM is it's poor empirical record as evaluated in a paper by Eugene F. Fama and Kenneth R. French in 2004 [27]. In 1992 Fisher Black and Robert Litterman [28] created the Black-Litterman Model in order to address the empirical issues of CAPM as well as the model's extreme sensitivity to expected return assumptions. Looking at the paper [28], we see that the main aim of the Black-Litterman Model is to reduce the problem of return sensitivity in asset allocation. The model combines M-V optimisation with CAPM. From the CAPM model we derive implied market returns to create a neutral weight starting point that can be biased in accordance with the investors views and confidence levels. Then, using the Bayesian mixed estimation technique values for expected returns can be generated. These values are then optimised using the M-V method to construct the Optimal Risky Portfolio. This significantly increases the performance when used properly, especially with regard to expected returns used to derive the optimisation of the portfolio. A major drawback of the Black-Litterman Model is that it requires investor to have specific views about the market without which the model loses a lot of it's advantages [29].

The early focus was on creating models to improve the estimation of expected returns as previously covered. Though you can also vary what you use as risk in general. Instead of variance, Value at Risk (VaR) which measures the potential loss of a portfolio over a specific time period and Conditional Value at Risk (CVaR) that measures the average potential loss over a time specific time period have been used with the M-V optimisation [30]. From the article [11] we can see that you can also try and optimise for the allocation of risk in each asset or asset classes like Risk Parity Optimisation does to distribute risk across portfolio assets and achieve good returns.

In practise, Markowitz assumption of normally distributed returns instead having a fat tail distribution [31]. A fat tail distribution means more extreme events are more likely to occur making more extreme loses and severe drawdowns more likely to occur. To solve this, there have been changes in the optimisation approach used such as applying Extreme Value Theory which is a branch of statistics that focuses on extreme deviations from a probability distribution [32]. The theory together with the GARCH model gives us Dynamic Optimisation which performs better, but is more complex as seen in [33].

2.1.2 Alternative Theories

While Modern Portfolio Theory is widely used now-days, there are other theories like Post-Modern Portfolio Theory (PMPT) [34] which uses downside risk of returns instead of the variance of investment returns used by MPT. Both theories describe how two risky assets should be valued, and how rational investors should utilize diversification to achieve portfolio optimisation. The difference lies in each theory's definition of risk and how it influences returns.

Another theory suggested in the paper [35] would be Behavioral Finance that proposes that psychological influences and biases have an affect on the financial behaviour of investors. This challenges the core assumption in all Modern Portfolio Theories models which is that investors are rational and markets are efficient. This isn't necessarily true as "Behavioral biases lead investors to violate the assumptions of traditional finance" [35, p. 860] thus the investors base their decisions on heuristics instead of using rational means with rational referring to a purely mathematical approach. Since investors are irrational the market efficiency assumption falls flat.

2.2 Machine Learning in Portfolio Optimisation

2.2.1 Existing Bayesian Approaches

Machine Learning opens up a range of new approaches to optimise portfolio performance, as discussed by Derek Snow in his 2020 paper [36]. Snow mentions one of the trends in research that continues to the present day is incorporating financial asset metadata in portfolio optimisation. An example in Snow's paper [36] of this was a Bayesian Sentiment approach which improves upon the Black-Litterman Model and it's subjective market views by taking the views of the public opinion and sentiment from the web instead of the individual investor.

Bayesian approaches can be used for optimisation and not just opinion inferences. One such method is Bayesian Linear Regression that can be used to directly predict Markowitz's Optimal Portfolio weights. This is due to a close equivalence between the Markowitz's portfolio optimisation and ordinary OLS as shown by Britten-Jones [5]. The close equivalence allows us to reinterpret any machine learning method that yields coefficients of a linear model as Markowitz's Optimal Portfolio weights as discussed in Snow's paper [36]. We haven't been able to find papers that use the Bayesian Linear Regression approach we have discussed for optimisation. There are plenty of recent paper's such as [37] discussing general Bayesian Linear Models and paper's like [38] forecasting just the asset returns, but not any applied directly to asset optimisation. The state of the art method that has been appearing in the past few years of academic research is using Bayesian Inference which derives the Bayesian Efficient Frontier and solve the resulting portfolio choice problem [39]. From the paper [39] we see that the Bayesian Efficient Frontier is the set of optimal portfolios obtained by employing the posterior predictive distribution which is one of the core parts of Bayesian Statistics. Bayesian Efficient Frontier is a recent alternative to regular M-V Efficient Frontier which is just the ratio of highest expected return for a specific risk.

2.2.2 Existing Stock Prediction Approaches

We can also use machine learning to try and improve the return of assets like Fama-French Model, Single Index Model and Capital Asset Pricing Models do. A recent example from the late 2010's research of this would be Hidden Markov Models (HMM) which can be used to predict future stock returns as seen with tech stocks in this paper [40]. The Hidden Markov Models consists of "2 stochastic processes" [41, p. 3] where a stochastic process as defined in [42] is just a family of random variables or in other words a random process. The first process doesn't produce directly observable states and as such those states are called hidden states [41, p. 3], [41, p. 28]. The second process produces observable states which are depended on the hidden states. These stochastic processes make HMM useful in regime switching where abrupt random changes to the market state are detected and model behaviour is switched accordingly allowing us to analyse stock market returns and forecast risks like volatility of VaR [43].

Other methods from research around early to mid 2010's include Autoregressive Integrated Moving Average (ARIMA) models as seen in the 2014 research paper [44] and Support Vector Machine models as seen in the 2013 paper [45]. To summarise, ARIMA Models are designed to predict future values using past data and it works for any time series data. From the book [46], we see that Support Vector Machine model can be used for both regression and classification problems. The model is used to find an optimal hyperplane that separates the data in multidimensional space into classes. A hyperplane itself is a decision plane that separates between a set of values having different class memberships. In regression problems, the hyperplane is the line that will be used to predict the continuous output.

2.2.3 Existing Deep Learning Approaches

Most of the current state of the art research involving machine learning focuses on the subsection called deep learning. This is due to the fact that with huge amount of data, we can in most cases develop more accurate models with deep learning rather than machine learning as seen in paper [47] which focuses on comparing stock prediction models. The deep learning model used in the paper is a convolution neural network (CNN) commonly used for image data. While it's an improvement on the machine learning models, the most up to date research uses Long-and-Short-Term Memory (LSTM) neural networks as seen in the 2021 paper [48]. It is a recurrent neural network technique that remembers values over a specific time interval which makes them well suited for use on time series data.

We can create deep learning models specifically for portfolio optimisation. There are various approaches we can take to optimise a portfolio. Derek Snow in his machine learning paper [36] included two deep learning methods. One is a supervised learning method consisting of a four step portfolio optimisation process which are autoencoding, calibrating, validating and verifying. This method was introduced by Heaton, Polson, and Witte [49]. *Autoencoders*[1] are used to predict the market and then the input are put into a supervised learning method to calibrate and map the market input to a portfolio. After some validation trade-offs between regularisation and errors involved in the previous 2 steps, the verification step creates efficient deep frontier of portfolios from which we pick our portfolio. From Snow's paper [36] another one is reinforced learning method called deep deterministic policy gradient. Reinforcement learning can be very unstable, hard to converge and tend to overfit.

We can also create various deep learning model architectures based on direct optimisation of a measurement like Sharpe Ratio as the highest Sharpe ratio represents M-V's most optimal portfolio. Such a method is presented by Zihao Zhang, Stefan Zohren, and Stephen Roberts 2020 paper [7]. The paper tests multiple neural networks such as fully connected neural networks consisting of only linear layers, CNN's and LSTM's. They found that LSTM's are best for modelling daily financial data.

[1] Autoencoder takes data into a neural network breaking it down to the base features and then recreating the original data in order to teach the network how to generate the input data [50].

3 Problem Analysis

Portfolio construction consists of asset selection and portfolio optimisation. We focus on the portfolio optimisation which tells us, given a set of assets we wish to trade with, how to decide on the weight allocation of each asset in our portfolio. Markowitz's M-V Optimisation is the standard portfolio optimisation technique still used to this day, but it has underlying problems we wish to improve upon in order to improve the practical performance of the portfolio. To do so, we are approaching creating the M-V portfolio in three ways using different machine learning models which will be selected from the ones discussed in the Literature Review. This leads to this papers overarching question of, can machine learning improve upon the standard portfolio construction technique? To answer the question, in this section we define the descriptive statistics, the statistical tests and hypotheses for us to use when we are comparing the models and benchmarks to each other.

3.1 Model Selection and Comparison

As discussed in the Literature Review 2, portfolio optimisation has a long history and a wide variety of approaches. This is why we chose to implement a range of different models to derive M-V portfolio weights as mentioned in the Introduction's Machine Learning section 1.2. The first one being a Bayesian Linear Regression (BLR) model with the Britten-Jones [5] approach of getting Markowitz's Optimal Risky Portfolio weights by equating the weights with BLR coefficients. We chose this type of Bayesian model because, as mentioned previously in Literature Review's Existing Bayesian Approaches section 2.2.1, we've not seen any papers mention direct M-V asset optimisation as a type of Bayesian approach. It's the only type of Linear Regression we've not seen any research on applying a direct asset optimisation approach applied to thus it's an unexplored research area we would like to investigate in our paper. Bayesian methods like the Bayesian Efficient Frontier as seen in this paper [39] are not suitable to use as we are focusing on improving standard M-V portfolio construction and not alternatives to it. While Bayesian Sentiment methods such as the one in Snow's paper [36] are unsuitable as they are cross between a trading strategy and asset optimisation and we are focusing purely on asset optimisation.

The second model we chose to use is the same one as Zhang [7] which is a LSTM neural network that directly outputs the asset weights for the Optimal Risky Portfolio by directly optimising Sharpe's Ratio which works because the M-V's Optimal Risky Portfolio is the one with the highest Sharpe ratio. This model is at the current cutting edge of portfolio optimisation techniques. LSTM unlike CNN and linear fully connected neural networks (FCN) can handle sequences of continuous data delivering the best performance for modeling daily financial data as shown by the Zhang [7]. This is due FCN overfitting and CNN underfitting the training data which in both cases fails to capture the underlying structure of the data causing bad performance on any test data. Approaches such as the Heaton, Polson, and Witte [49] while effective, it attempts to predict the market return using autoencoders and only then optimises for an optimal portfolio instead of doing it directly. We are directly optimising for the optimal portfolio in this model approach which is why we didn't use it here. Reinforcement learning methods as seen in Snow's article [36] can be used to solve complex problems such as portfolio optimisation, but as mentioned previously they can be very unstable, hard to converge and tend to overfit making them hard to implement unlike LSTM neural networks making reinforcement learning undesirable to implement [36].

The third and final model is a Hidden Markov Model that predicts future expected returns on which we apply the M-V optimisation to get portfolio weights. While Support Vector Machine or more specifically the subset of it called Support Vector Regression (SVR) have good accuracy, it is memory intensive and it performs better in lower volatility periods when doing stock prediction as seen in this paper [51]. We expect stock price changes to be fairly volatile making SVR less than suitable to use in this paper. Plus, HMM have more recent research done on them than SVR in regard to stock prediction and as such HMM would be a more up to date model to look at than SVR. ARIMA models on the other hand are simple, fast and robust method for short-term forecasting, but they do not tend to do too well in long-term forecasting as shown in this paper [52]. We want to forecast a year's worth of values for which ARIMA models won't perform well at and as such won't be used in this paper. This leaves us with just HMM and LSTM models to pick from. The HMM stock prediction performs better than LSTM stock prediction as seen in this 2019 thesis paper by Bram de Wit [53]. Due to this better performance, we picked HMM instead of LSTM even if current research trend focus on improving LSTM and other deep learning stock prediction models.

3.2 Statistics for Result Evaluation

3.2.1 Statistical Hypothesis Tests

The question we are trying to answer in this paper is, can machine learning improve upon the standard portfolio construction technique? To answer the question posed, we need to expand on the comparison part of the goals 1 and 2 from the Introduction's Summary of Goals section 1.4. We do this by defining the statistical hypothesis tests which we are going to use to compare the models and benchmarks. From the book [54], we can see that statistical hypothesis tests have something called null hypotheses (H_0) which shows that no statistically significant difference exists in a set of given data samples and any observed difference in the data is due to chance alone. Rejecting the H_0 means we accept the alternative hypothesis which in general means that the observed difference in the data isn't due to chance alone. Something to keep in mind is that statistical hypothesis testing doesn't tell us whether or not the hypothesis is true or not instead it tells us the likelihood of it being true [55]. This means that there is a probability that we erroneously accepting or rejecting the H_0.

To answer the initial question, we will be performing 3 statistical hypothesis tests and the H_0 for them would be the following:

1. H_0: There is no difference between M-V benchmark portfolio daily returns and the EQ benchmark portfolio daily returns.

2. H_0: There is no difference between the portfolio daily returns given by HMM, BLR and LSTM models and the M-V benchmark portfolio daily returns.

3. H_0: There is no difference the portfolio daily returns given by HMM, BLR and LSTM models and the EQ benchmark portfolio daily returns.

Where the testing the H_0 1 and 3 allow us to meet goal 2 from the Introduction's Summary of Goals section 1.4 by the result telling us if any of the models or the M-V benchmark performs better than the absolute baseline benchmark. For example, if H_0 1 does perform better we would get a *p-value*[1] of smaller than 5% allowing us to reject the H_0 and say that the M-V optimisation on historic data makes a significant difference in portfolio construction instead of just assigning all assets the same importance in a portfolio with no evaluation of the assets at all. Doing a statistical hypothesis

[1] p-value is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test [55]. Typically, we take the p-value significance level of 5% when evaluating the null and alternative hypothesis.

test using the H_0 2 would allow us to meet goal 1 which would answer the core question of this paper of whether or not the machine learning models improve upon the standard M-V optimisation.

The statistical hypothesis tests we will be using are the Wilcoxon Signed Rank Test to test the H_0 1 and Friedman Test for the H_0 2 and 3. From the technical report [55], we can see that we need to use Wilcoxon Signed Rank Test on H_0 1 because we are only comparing 2 groups unlike H_0 2 and 3 which require comparing 4 groups and as such Friedman Test is used for them. In general as seen from [56], [57], the two statistical tests are used on paired, non-parametric data which the resulting portfolio returns data is going to be like. We can know that the resulting data from the models would be paired as every data point for a given model's portfolio returns would have a uniquely matching data point in every other model's portfolio returns [58]. This is due to us using the same group of assets over the same time period for every model we will test which allows for easy comparison. We also know that the resulting data from this paper's experiment will be non-parametric due to the portfolio return data violating parametric assumptions such as the equal variance assumption in each data group where the assumption is given here [59] . Each asset has a different variance that varies over time and since we are selecting different amount of different assets, the resulting portfolio variance will be different for each model and thus violating the parametric equal variance assumption.

3.2.2 Descriptive Statistics

As mentioned in the Introduction's Performance Measurements section 1.3, we have performance measures Sharpe ratio, Calmar ratio, CAGR, volatility, MDD and the return of each portfolio that we will calculate and use to compare the performance of the portfolios given by the models relative to each other. In order to properly interpret the measures, we'll establish here what is considered good values for each of the measurements. For Sharpe and Calmar ratio, a value lower than 1 is sub-optimal, greater than 1 is considered acceptable to good, above 2 is very good and above 3 is considered excellent performance of the portfolio or asset [60]. Sharpe ratio results meaning can be used for Calmar ratio as well because the Calmar ratio results would have a similar meaning to Sharpe ratios.

An article from Investopedia [61] shows the usual definition for a bear market or in other words a very bad decline to be a downwards volatility is 20% or even more more. While a less bad, but still notable decline of a downwards volatility of 10% is called a correction. From this we can take the downwards volatility fluctuations of less than 10% to be normal and assume that the upwards volatility to be normal in the same range. Since volatility is both upwards and downwards risk, so we can use the MDD which focuses on downwards volatility in the recent period to identify above normal or extreme risk that might be associated with an asset.

To consider an asset or portfolio to have a good CAGR value the returns need to be greater than the inflation rate which is on average taken to be 2-3%, but it can be higher as seen by the US consumer price index reaching 8.5% in 2021-22 [9]. Lastly, the returns of each portfolio will be taken as the daily percentage change in price. For the returns anything above 0 is good. While negative values are bad as it means we are loosing money instead of gaining it. Overall, using these performance metrics, we can meet the goal 4 mentioned in the Introduction's Summary of Goals section 1.4 as we can now know the standard expected values for these metrics to compare general performance of the models against as well as how they compare relative to each other.

To see how the general trend of the portfolios returns and how they change over the testing period, we will be plotting the Equity Curve where the positive slope means the portfolio is generating a profit and negative slope shows that the returns of the portfolio are generating an overall loss [62]. We can also plot it for the underlying assets to see how they perform and what it would mean for the portfolio performances in general. This would allow us to see the performance of portfolios and how the portfolio with relates to what assets it has in. To actually plot the graph we will need to plot the *Rate of Return*[2] for every day in the testing period. This will tell us what the overall percentage change in the portfolio returns price from the first day of the portfolio to the day we are looking at.

3.3 Ethics Statement

This paper uses only publicly available stock market data for the assets used and hasn't used human data. We adhere to all legislation and appropriate ethics consideration in this project. Everything in this paper is meant for research purposes only and any suggestions made are not proper financial advise. If you are seeking financial advise, do so only from a certified financial advisor that will tailor advise to you individually and give advise specific to your circumstances. If you wish to invest in the stock market or follow any suggestions made in this paper, you are doing it on your own risk and can possible loose all your money causing you financial harm.

[2]Rate of Return as defined in [63] is simply the net change of an investment over a specific time period, expressed as a percentage of the initial investment.

4 Design and Implementation

Before the models can be implemented, good data needs to selected to train with and a trading strategy picked that the portfolios will follow. Then the machine learning models can be created that will produce Optimal Risky Portfolios and we can create the benchmark portfolios to compare the machine learning models against. After creating the models and benchmarks, we will use training data on them to get portfolio weights. These weights then will be used to create portfolio returns for unseen trading year test data to produce realised portfolio returns that will be used in model evaluation.

4.1 Constraints and Trading Strategy

During the construction and analysis of portfolios, the following constraints will be applied:

- No trading costs.
- No Explicit costs like commissions, fees, taxes.
- Infrequent re-balancing of the portfolio. Only every 12 months.
- No short-selling allowed.
- Starting portfolio will have the value of 10000$.

The portfolios will follow a classical buy and hold trading strategy with re-balancing every 12 months. A buy and hold strategy given by [10] is when an investor buys assets and simply holds them for a long time to allow them to accumulate value. It is a long-term passive strategy that doesn't care for short-term asset volatility. Re-balancing as seen in [64] is when you buy and sell assets periodically to maintain the original portfolio weights. It is needed as over time portfolio weights change due to individual assets rising or falling in price.

4.2 Choosing Software and Libraries

The recent trend in academia is to use Jupyter Notebook with Python when creating machine learning models and as such we will also be using it in this paper. Python has a wide variety of libraries available for use which is one of the main reasons for it's popularity. For deep learning, PyTorch is the standard library used in academia. For stock market data yfinance provides an interface to access stock market data from Yahoo Finance and is often used to get stock data for testing models. Yfinance usually used alongside pandas which is a popular Python data manipulation library. Pypfopt is a library that calculates the Markowitz Optimal Risky Portfolio and it'll be used to in this paper when we need to create such portfolios using portfolio returns. For Hidden Markov Models, the hmmlearn library is suitable for our use case and it implements all the maths of the model for me. The final libraries we will use are statsmodel to implement the statistical tests we will be using in the evaluation of the models and matplotlib to create the graphs from our results.

4.3 Data Selection and analysis

You can't use machine learning without any data to build a model from; making data selection a vital part of the project. The data was selected from four asset groups with varying risk levels which are large-cap stocks, small-cap stocks, commodities and bonds. In order to not have to select individual assets from an asset group, we chose to pick two Exchange-Traded Funds (ETFs) per asset group. As seen from the Investopedia [65] description, the ETF derives values from the underlying securities. This allows us to represent the general sectors without worrying about different asset acquisition and the representations of returns. For stocks, I've chosen to include US (GSPC, RUT) and European (FTSE, IEUS) big-cap and small-cap markets. For bonds, I've chosen ETFs representing international (IGOV) and US bond (GOVT) markets. For commodities, we chosen ETFs that represent gold (SGOL) and silver (SIVR) markets.

All data used comes from Yahoo Finance with a starting date of 2012-02-24 and end date of 2022-02-25 giving us ten years worth of data to work with. From all data features, we only use the close asset price. Nine years will be used as training data while the last trading year will be used as the unseen data to test all the model performances and evaluated them. The original asset data in general has different scales, no seasonality and an upwards trend except for the silver ETF which has an overall downwards trend for the 10 year period. Using the Dickey-Fuller Test (DFT) for stationary [66] on the initial data, we get a p-value of above 5% for all assets meaning the data is not stationary. This means that preprocessing needs to be done before the data can be used in the models as data stationary is a requirement for many models as seen in [67]. We can preprocess the data by taking the percentage change between the current close price and the previous elements close price for the entire dataset which gets you assets with daily percentage change in returns. This removes the trend, gives the assets the same scale and make the assets stationary as seen by the DFT giving a p-value of below 5%. Some models will require additional preprocessing specific for them, but in general using the percentage daily change in returns will be all the preprocessing that needs to be done.

4.4 Benchmarks

For portfolio optimisation in machine learning, academically there are two commonly used benchmarks. The first one being EQ portfolio that gives the same importance to each asset in the portfolio. The second one is classical M-V optimisation on historic data. To calculate it we simply use the Pypfop library to find the the Optimal Risky Portfolio weights. We'll be using them both to see if M-V Portfolio beats the absolute baseline EQ portfolio and if the machine learning models manage to improve the classical Markowitz optimisation.

4.5 Bayesian Approach Model

In this section, we are implementing the Bayesian Linear Regression (BLR) model. In order to do so, we are using article by Halford [68] with reference to this book [69, pp.1-60] in order to check the validity of the maths and terminology definitions. All equations and terminology come from Halford [68] as he defines the maths for updating the model which allows for easy implementation and presents the terminology is a simple manner. In Bayesian statistics, the goal is to determine the posterior distribution for the model parameters unlike frequentest statistics where you calculate the single best value of the model parameters. To calculate the posterior, we first need to define the likelihood function. We'll take it to have a Gaussian distribution because of Markowitz's assumption of asset returns being normally distributed making the likelihood function look like the following:

$$p(y_i|x_i, w_i) = \mathcal{N}(w_i x_i^T, \beta^{-1}) \tag{4.1}$$

where $p(y_i|x_i, w_i)$ is the likelihood, which is the probability distribution of an observation y_i conditioned on the current model parameters w_i and a set of data features x_i. Likelihood allows you, for a given state of the model, to see how realistic is it to observe the pair (x_i, y_i). β^{-1} is the precision which is inversely related to the noise variance: $\beta = \frac{1}{\delta^2}$. In our model implementation, we treat this simply as a constant 1, but it can be varied to adjust the model belief of how noisy the target distribution is.

The corresponding appropriate prior distribution for the above likelihood is the following multivariate Gaussian distribution:

$$p(w_0) = \mathcal{N}(m_0, S_0) \tag{4.2}$$

where $p(w_0)$ is the prior distribution of the model parameters, m_0 is the initial mean of the distribution and S_0 is the initial covariance matrix.

Now, we can define the posterior distribution by combining the likelihood of (x_i, y_i) and the prior distribution of the current model parameters w_i:

$$p(w_{i+1}|w_i, x_i, y_i) = \mathcal{N}(m_{i+1}, S_{i+1}) \tag{4.3}$$

where $p(w_{i+1}|w_i, x_i, y_i)$ is the posterior distribution which represents the distribution of the model parameters w_{i+1} once a new pair (x_i, y_i) is given to the model with parameters w_i. m_{i+1} and S_{i+1} are defined as the following:

$$S_{i+1} = (S_i^{-1} + \beta x_i^T x_i)^{-1} \text{ and } m_{i+1} = S_{i+1}(S_i^{-1} m_i + \beta x_i y_i) \tag{4.4}$$

where m_{i+1} and S_{i+1} are the updated mean of the model and covariance matrix of the model for the new data pair (x_i, y_i).

We can go on and define the predictive distribution of the model to predict new y values, but we won't be using that as we are only concerned with knowing the model parameters w for the model trained on all the stock data which knowing the prior and posterior allows us to calculate. The only thing from the predictive distribution that must be kept in mind is that the mean distribution of the model m corresponds to the model coefficients w. With regard to the stock data, we only need to do the standard preprocessing on it. This means that the input stock data needed for this model is just the percentage daily price change in returns for all 8 asset close prices.

Finally, to make sure that the coefficients yields the Markowitz Optimal Portfolio weights, we do a regression of a constant 1 onto a set of asset returns without the intercept term as shown by Britten-Jones paper [5]. It means we are passing into the model a constant y value of 1 for all input data x. This results in the coefficients of the BLR corresponding to the Optimal Risky Portfolio weights which are scaled so that the weights sum up to 1 allowing us to treat the coefficients returned by the trained model as the allocated weights in the portfolio which will be the predicted portfolio for the test data.

4.6 Hidden Markov Model

We are creating a Hidden Markov Model (HMM) to predict the future price of the 8 stock assets before applying M-V optimisation to calculate asset allocation for it. Each asset will have its own HMM model trained for it that will predict the expected returns for the asset. Before an asset can be used in a HMM model, we need to do the standard preprocessing on the input asset by taking the percentage daily price change in returns for asset close.

The HMM itself is defined by 3 model parameters as seen from the book [41, p. 28]. The initial state probabilities π, hidden state transition probabilities A and the observable emission probabilities B. Where B will be the Gaussian emissions in this model implementation due to Markowitz's assumption of normally distributed asset returns. We can write the parameters as $\lambda = (A, B, \pi)$ where λ represents all parameters of HMM. From the book [41, pp. 8-11], we can see that we need to solve the learning problem to determine the optimal set of model parameters λ given a set of observed states O. For that we need to find the maximum for $P(O|\lambda)$. We can't do that globally, but we solve it locally using an iterative procedure such as the Baum-Welch algorithm. Hmmlearn library implements the Baum-Welch algorithm to find the model parameters for us. As such for the Gaussian HMM, we only need to define the number of hidden states which will be 4 and pass the input data for the model in order for it to find the parameters. Now that the model is trained on the training data, we can predict the trading year for the test data and use the resulting returns to get Markowitz Optimal Risky Portfolio weights.

4.7 Deep Learning Model

4.7.1 Model Architecture

For the LSTM model, we took the same general approach as the paper by Zhang [7]. The LSTM neural network's architecture consists of 3 parts, input layer, hidden layer and output layer. The input layer for a LSTM neural network must have a 3-dimensional data input in the shape: batch, sequence and features as seen from the PyTorch documentation [70]. From the article [7] and example LSTM by Rezaeian [71], we can see that a batch represents how many separate data samples are you going to be working on at one time in the network, sequence tells the network how many data points at one time we are looking at in order to form the output prediction and features shows how many independent time series will be given to the network. A single batch will form a pair of data consisting of the current sequence I'm looking at and the labeled future sequence we're going to try to predict. These labeled data batches are then used to train the network. The length of the labeled future sequence we're predicting can be different from the sequence representing the past data, but the features and batch size must match. Our LSTM neural network is structured to use the 252 current trading days that make up a trading year to predict the Optimal Risky Portfolio's 8 asset weights for the next trading year with 25 samples being looked at at the same time. As a result, our input layer shape for labeled data and it's training unlabeled counterpart is $(25, 252, 8)$. This is different from what is used by Zhang [7] as they use 64 samples at the same time with 50 days of past data.

As seen in the article [7], the hidden layers are the core of a network's architecture where the intermediate layers between the input and output do all the computations and extract the features from the data. We have a stack of 2 layers for our hidden layer part and these are a LSTM layer followed by a Linear layer. It's a shallow network, but we don't have a lot of data to make deeper networks necessary.

Finally, we have the output layer that has the shape $(25, 8)$ for our network. We have 1 output value for each of the 8 assets in the portfolio and we have this for each data sample in the batch of 25 samples which is different from Zhang [7] as the input batches size matches the output. To ensure that the output values for the 8 assets properly correspond to portfolio weights, are positive and add up to 1, we need to add a Softmax activation function to the output values. Now any future output predictions will return the optimal portfolio weights for the next year while taking in the previous year as input.

4.7.2 Objective Function

Our approach for the objective function is different from the paper [7] due to PyTorch not having gradient ascent. We have to use stochastic gradient decent (SGD) as an optimisation strategy with a negative Sharpe ratio as a loss function to ensure that the weights of the LSTM neural network are predicting for a portfolio with the highest Sharpe ratio. Overall, the same objective is accomplished and the nearly identical maths defined and used as the paper [7]. The loss function would be defined by the following:

$$L = -\frac{E(R_p) - rf}{Std(R_p)} \tag{4.5}$$

where $E(R_p)$ and $Std(R_p)$ are the mean and the standard deviation of the portfolio returns. rf is the risk-free rate that is taken to be 0% due to there not being any assets that are completely risk-free. SGD optimises for the lowest value instead of the highest. Therefore by taking the Sharpe ratio to be negative we are finding the highest value for the ratio when it is the most negative or in other words when SGD finds the lowest optimal value.

A single sample in the batch would have a trading period of $t = \{1, ..., T\}$ giving us:

$$E(R_{p,t}) = \frac{1}{T}\sum_{t=1}^{T} R_{p,t} \text{ and } L_T = -\frac{E(R_{p,t})}{\sqrt{E(R_{p,t}^2) - (E(R_{p,t}))^2}} \tag{4.6}$$

where $E(R_{p,t})$ and L_T represent the expected mean return of the portfolio for the trading period and the total negative Sharpe ratio loss for the that specific batch in that period. To actually calculate the $R_{p,t}$ which is the portfolio return over the n assets used at time t, we would do the following:

$$R_{p,t} = \sum_{i=1}^{n} w_{i,t} * r_{i,t} \tag{4.7}$$

where w_i is an asset's weight in the portfolio at time t and r_i is it's corresponding asset returns at time t. Now that we have defined the custom loss function that will calculate a criteria we want to optimise the network for, we can use SGD defined in [72] to update the underlying LSTM network's weights. This SGD with a notation change would look like the following:

$$\theta_{i+1} = \theta_i - \alpha * \frac{\partial L_T(x^j, y^j)}{\partial \theta} \tag{4.8}$$

where θ is the network's weights instead of the portfolio's, α is a an adjustable hyperparameter called the learning rate and $L_T(x^j, y^j)$ is the loss function for a single training example where x^j and y^j would represent the training sequence and predicted testing sequence respectively. We use the gradient of a single training example $\frac{\partial L_T(x^j, y^j)}{\partial \theta}$ to update all parameters in the network while Zhang [7] use the whole batch. This backwards pass through the network is called backpropagation that Pytorch does this for us using the SGD optimiser.

4.7.3 Training and Prediction

We have defined the network architecture and how to update it to get the Optimal Risky Portfolio. Now, we have to train the network and predict the weights for the test year. We take the initial 8 asset percentage daily price change in returns for the entire 9 year trading period that comprises the training data where trend is removed and the data is stationary. Split it into trainX containing all possible 252 data point sequences which would be equivalent to a trading year and trainY that contains the corresponding 252 data point labeled future sequences for each individual sequence in trainX. Now we loop through the entire dataset updating the network weights each time using the SGD optimiser with the negative Sharpe ratio as a loss. Each time pass through the entire dataset is called an epoch and the network stops training after 30 epochs. We train less than the 100 epochs that the study [7] does as longer training sessions causes the loss function to worsen instead of improve. We are satisfied with a learning rate of around 0.20-0.25 due the Sharpe ratio loss properly converging with that hyperparamater value. Finally, to predict the weights for the test data, we just input the the the last year in the training data as an input into the LSTM neural network and get out the predicted weight allocation for the M-V Optimal Risky Portfolio for the test trading year.

4.8 Create Test Portfolio Data

From each model, a Optimal Risky Portfolio for the trading year starting with 2021-02-04 will be given. The trading year which will be the test data that has not been seen by any of the models during the model training. Only the training data between 2012-02-04 to 2021-02-24 would have been seen by the models. For example, the HMM will train on all the training data to predict the testing data and the M-V optimisation will give the Optimal Risky Portfolio weights which will be used to calculate the initial weighted portfolio and its realised returns over the test trading year. The calculation will look like:

$$T = \sum_{i=1}^{n} \frac{w_i * p}{m_i} * s_i \tag{4.9}$$

Where T is the total portfolio price at a single point in time, p is the initial portfolio starting price, w_i is the weight of an asset in the portfolio, m_i is the corresponding initial asset price for that weight and s_i is the corresponding asset price for that weight a single point in time. This calculation is repeated for every single date in the test trading year to give the portfolio realised returns over the trading year. To get individual asset realised returns it would look like:

$$A = \frac{w * p}{m} * s \tag{4.10}$$

Where A is the asset we are looking at while w,p,m and s are the have the same meaning as the previous equation just without the summation element. The realised weight and portfolio returns calculation is then repeated for every single model and benchmark portfolio weights to get the test data results for everything in order to evaluate the models.

5 Results and Evaluation

5.1 Discussion of Portfolio Performance

For each model, the portfolio Rate of Return (RoR) is plotted for all days in the test year as can be seen in the figure 5.1 and all of the performance measurements for the test year can be found in the table 5.1. An important fact to note is that since we are testing on exactly 1 trading year the last day's RoR value in figure 5.1 gets inherently annualised making that RoR value the same as CAGR value in the table 5.1 due to the fact that CAGR is just annualised RoR.

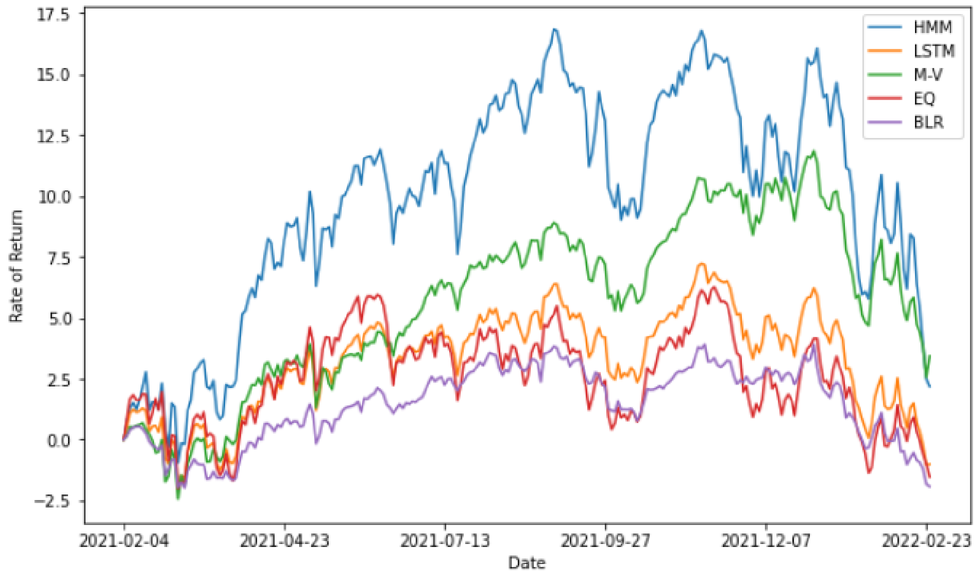

Figure 5.1: Portfolio Performance on the Test Trading Year

In the figure 5.1, we can see that while the HMM portfolio outperforms the benchmark M-V portfolio for most of the year, the HMM portfolio return falls bellow the M-V portfolio return at the very end. This is supported by HMM and M-V portfolios having the CAGR values of 2.18% and 3.42% respectively as seen from the table 5.1. The M-V portfolio returns beats the average inflation rate of 2-3% while the HMM portfolio falls within it, but for the test year 2021-22 the inflation rate was higher than average for the

	Portfolio Type				
Measurements	EQ	M-V	BLR	HMM	LSTM
Sharpe Ratio	-0.110	0.450	-0.350	0.228	-0.0852
Calmar Ratio	-0.146	0.447	-0.323	0.246	-0.0894
Volatility	9.71%	8.26%	5.19%	13.5%	8.08%
MDD	-7.34%	-8.31%	-5.61%	-12.5%	-7.70%
CAGR	-1.52%	3.42%	-1.92%	2.18%	-1.00%

Table 5.1: Performance Measurements of the Test Data

whole world. Comparing to both the US and UK inflation rate of 8.5% and 4.9% respectively, we can see that we lose money in both portfolios due to inflation as we haven't grown our initial portfolio investment enough.

From the figure 5.1, we can also see that the BLR, the LSTM and the EQ portfolios perform similarly to each other throughout the year and end up with negative returns which means we have have lost money in our portfolio instead of gaining it. The table 5.1 CAGR also shows that those portfolios have lost money by being negative for them. The portfolio with the least loss out of the 3 negative return portfolios is the LSTM portfolio of -1% CAGR, while the worst performing one is BLR portfolio of -1.92% CAGR. What this means for these 3 portfolios is that we are losing money in addition to the money lost due inflation which is counterproductive when we choose to use the stock market in the first place to at least save money lost due to inflation.

The bad return performance of the models and benchmarks could be explained by the fact that most of the assets used in the portfolio don't perform well in the test year as seen in the figure 5.2 which shows all asset RoR for every day in the test year. We see that GSPC, FTLC and SGOL are the only assets with a positive returns at the end and every other asset for that year is not. This could suggest that the overall stock market did not perform particularly well in 2021-22 as the ETFs chosen in this study represent entire markets for the US stocks and bonds, the EU stocks and world markets for bonds and commodities. If we look at the table 5.2 which shows the percentage of the total portfolio value that each asset represents, we can see that the 2 profitable portfolios HMM and M-V have a high percentage of the total portfolio invested in GSPC which seeing the assets performance in the figure 5.2 explains why they performed so well in comparison to the other 3 portfolios.

Something else to note from the figure 5.2 is that the all stock and bond assets start to drop from January 2022 except for commodities that rise instead which is why the positive trend of the portfolios in the figure 5.1 reverses around the same time becoming a rather sharp negative trend for

Figure 5.2: Asset Performance on the Test Trading Year

	Portfolio Type				
Asset Type	EQ	MV	BLR	HMM	LSTM
^GSPC	12.5%	51.7%	15.3%	37.9%	20.9%
^FTLC	12.5%	0.00%	5.57%	4.22%	6.72%
^RUT	12.5%	0.00%	0.60%	0.00%	9.84%
IEUS	12.5%	0.00%	4.89%	53.6%	11.2%
GOVT	12.5%	48.3%	62.0%	4.28%	34.6%
IGOV	12.5%	0.00%	7.86%	0.00%	7.80%
SIVR	12.5%	0.00%	2.82%	0.00%	4.18%
SGOL	12.5%	0.00%	0.97%	0.00%	4.83%

Table 5.2: Portfolio weights for each Asset

all portfolios giving us a curved graph of a rise and a fall of returns. This suggests that a single type of portfolio can't handle all market conditions which is why as mentioned in Literature Review's Existing Stock Prediction Approaches section 2.2.2 regime switching exists and it could have improved performance if implemented for this test year. The portfolio least affected by this trend would be the EQ portfolio in this case due to the 25% investment in commodities which would balance the losses from stocks and bonds somewhat, but not entirely as seen by the fact that the EQ portfolio returns still have a downwards trend just like all the other portfolios as seen in the figure 5.1.

From the figure 5.1, we can also observe that the HMM portfolio fluctuates the most out of the 5 portfolios which is supported by the volatility and MDD measures of 13.5% and -12.5% found in the table 5.1. Due to the HMM portfolio having MDD of -12.5% where the negative is just referring to the fact that it is downwards risk, we can say that the portfolio has experienced a correction and general volatility can be considered abnormal due to being greater than 10%. This can't be considered too unusual when you look at the table 5.2 because nearly all of the portfolio value is being invested in GSPC, IEUS and FTLC stocks which are more volatile than IGOV, GOTV bonds and SGOL commodity as seen in the figure 5.2. If we look in the table 5.1 at how other portfolios volatility is like, we can see that the rest of the portfolios lie within the expected less than 10% range with the benchmark EQ portfolio having the second highest volatility of 9.71% and the BLR portfolio having the lowest volatility of 5.19% due to the high investment in the bonds asset class as seen by the 62% investment in GOVT and 7.86% in IGOV assets from the table 5.2. The table 5.1 also shows that M-V portfolio has the second highest MDD of -8.31% which is higher than its general volatility of 8.26% suggesting that the downwards movement of the M-V portfolio could be more volatile compared to it's upwards movement.

All portfolios have Calmar and Sharpe ratio of under 1 meaning all of them have sub-optimal performance in this test year. The LSTM, the BRL and the EQ portfolios are negative meaning that the investment return is lower than the risk-free rate which we have taken to be 0 in this paper as there is no such thing as a truly risk-free asset. In other words, the negative Calmar and Sharpe ratio means that by investing into the stock market you are loosing more money than by not doing any investment at all.

Something interesting to note about the BLR and the LSTM model portfolios from the table 5.2 is that they have invested into SIVR which as mentioned in the Methodology's Data Selection and Analysis section 4.3 is the only asset that has loss value over the entire data period we are looking at. The BLR and the LSTM models created portfolios that still put value into an asset that has been decreasing in price which is rather counterproductive to the goal of earning money, but it does have a similar property as the SGOL commodity of having a positive return trend when the rest of the market has a negative return trend.

5.2 Results of Hypothesis Testing

We have tested the hypotheses defined in the Problem Analysis section 3.2.1 and the results are as follows:

1. H_0: Wilcoxon Signed Rank Test gives a p-value of 0.788 or given as a percentage 78.8% which is way above the 5% significance threshold. This means that we cannot reject the H_0 and it means that there is

no statistically significant difference between M-V benchmark and EQ benchmark portfolio distributions of daily returns. This could mean that both of the benchmarks come from the same distribution of returns and any observed difference in the daily distribution returns come from chance alone or that we simply don't have enough evidence to reject the H_0 and we need to increase the power of the hypothesis test. Either way, our statistical test accepts the H_0 which is surprising, but might be due to the fact that we focus on variety of asset types with a small number of overall assets that represent whole market groups instead of just a huge number of individual stocks ensuring a more similar distribution of returns.

2. H_0: Friedman Test gives a p-value of 0.420 or given as a percentage 42% which is above the significance threshold of 5% allowing us to accept the H_0 and say that there is no significant difference between the HMM, the BLR, the LSTM models and the benchmark M-V portfolio distributions of daily returns. This means that there is insufficient evidence to suggest that any of the 3 machine learning models made any significant improvements to the Markowitz portfolio optimisation. This also means we have failed to improve in any statistically significant way upon the standard portfolio construction technique using machine learning.

3. H_0: Friedman Test gives a p-value of 0.530 or given as a percentage 53% which is above the significance threshold of 5% allowing us to accept the H_0 and say that there is no significant difference between the HMM, the BLR, the LSTM models and the benchmark EQ portfolio distributions of daily returns. This means that there is insufficient evidence to suggest that any of the 3 machine learning models made any significant improvements to just doing no analysis, evaluation or optimisation of the assets and assigning equal weighting of all assets in the portfolio. If we were including transaction costs there might have been a difference due to the penalty of the extra costs when buying many different types of assets and thus making optimisation for the best assets actually beneficial with this dataset.

5.3 Implementation of Goals

In this section we will consider how well we reached each of the goals listed in the Introduction's Summary of Goals section 1.4.

1. We successfully used M-V optimisation on historic data to create the portfolio weights that were used on the test year data to create the

benchmark M-V portfolio and compared the machine learning model portfolios against it. We did this by implementing the M-V optimisation on training data using pypfopt library and then used the resulting portfolio weights on the test data to create the benchmark for the test trading year. Finally, we used the Friedman Test for the statistical hypothesis testing on the HMM, the BLR, the LSTM machine learning portfolios and the benchmark M-V portfolio in order to compare them. The test's results showed that there is insufficient evidence to suggest a statistically significant difference between any of the 3 machine learning portfolio daily returns and the M-V benchmark portfolio daily returns.

2. We successfully used equal weighting of each asset on the test year data to create the benchmark EQ portfolio and compared both the M-V benchmark portfolio and other machine learning model portfolios against it. We use the Wilcoxon Signed Rank Test as the statistical hypothesis test to compare the 2 benchmark portfolios and saw that there is no significant difference between the returns of the 2 benchmarks. For the 3 machine learning portfolios and the EQ benchmark, we used the Friedman Test instead and saw that there was also insufficient evidence to suggest that there is a statistically significant difference between any of the model portfolios and the benchmark.

3. While we successfully implemented the HMM, the BLR and the deep neural network using LSTMs, we failed to improve upon the benchmark M-V portfolio in any statistically significant way as seen by the statistical test done for comparison between the M-V benchmark the machine learning models. We implemented the LSTM neural network by using the pytorch library with a custom created loss function to calculate the Sharpe ratio for direct optimisation of the network. The HMM was implemented by using the hmmlearn library and the BLR was implemented by coding the maths described in the Methodology's Bayesian Approach model section 4.5.

4. We successfully calculated the Sharpe ratio, Camlar ratio, CAGR, volatility and MDD for each model and benchmark portfolios. Then we successfully used them to discuss, compare and evaluate the performances of all the portfolios against each other.

5. We successfully evaluated and discussed portfolio weights, returns and asset data used for the model and benchmark portfolios.

6 Conclusion

The benchmark M-V portfolio outperformed the EQ benchmark and the HMM, the BLR and the LSTM machine learning models. Though when compared using the Wilcoxon Signed Rank Test against the EQ benchmark, the M-V benchmark failed to show sufficient evidence to suggest that there was any statistically significant difference between the returns of the 2 benchmarks. The HMM, the BLR and the LSTM machine learning models also failed to show any statistically significant difference in returns when compared using the Friedman Test against the benchmarks. What this means is that the machine learning models used in our paper have failed to improve upon Markowitz portfolio optimisation. Therefore the answer to the initial question posed by this paper is that while there is other literature suggesting that we can improve the standard portfolio construction techniques using machine learning models, our paper fails to provide any statistically significant evidence to support this claim.

The LSTM portfolio bad performance in this paper is inconsistent with other research papers such as [7], [48] that show the model having good performance. The bad performance in our paper could be because we are looking at a single trading year for which the underlying assets don't perform well in general. While the general approach was the same as in the paper [7], the slight differences between our models definition and implementation could be the reason for the inconsistent performance between the LSTM models. Further research on the LSTM neural network with varying asset types on multiple years is required to check if we can consistently create good performing portfolios in different settings. We can also create different variations of the LSTM models and test on the same time period to see how the model performance compare then. This research is important to do because we have to ensure we have enough evidence to say with confidence whether or not the model performance is good in general and not in a specific case. If it does provide a good general use case then it would be significant accomplishment as we would have gained a model that performs well overall in practise which many financial models fail to do.

While the BLR performs the worst out of the 3 machine learning models, it is the most stable with small everyday volatility which could imply that BLR creates less volatile portfolios in general. Though the BLR model return performance suffers in exchange for that stability in volatility. With no other literature to use as comparison it is impossible to draw any definitive conclusions and as such further research is required. The research should test on multiple years of data with varying asset classes, so that the yearly difference in performance can be compared within the study in addition to other studies since a single year testing might not be very representative of how a portfolio with a long term buy and hold strategy performs. The results of this research is important as low volatility is important when you withdraw money from your portfolio which people depended on the income from their portfolios will be doing and as such research into stable reasonably well performing portfolios is vital for such individuals.

While the HMM in future stock prediction does outperform all excluding the benchmark M-V portfolio, it still isn't very accurate when predicting the direction of the stock price. Each time we predict a sample future prediction with HMM, the prediction changes and as such the performance of the model can vary drastically unlike the other models which would predict the same output for the test data given the same input training data. This is consistent with other literature that suggests the HMM is only right 51% of the time [53] meaning that HMM is barely better than randomly guessing making it a very unreliable model. As always further research is required, but we don't believe this particular area should be the focus when wanting to improve the portfolio construction due to the inconsistent and unreliable results produced by future stock prediction.

Overall, the recommended further research is important be done in order to improve standard portfolio construction techniques using machine learning. This would allow us to provide people with a more reliable method of ensuring that their saving aren't lost to inflation increasing the financial security of the general public. This doesn't just apply to individual people who wish to save money, it also applies to to governments as they invest people's pensions into assets in order to retain the value of the pension thus making this a vital area of research.

A Diagrams

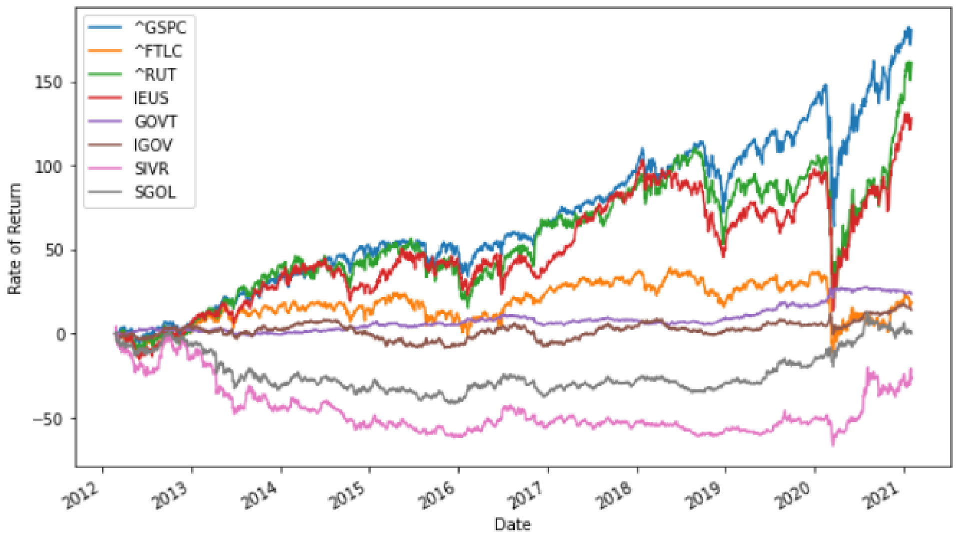

Figure A.1: Asset Returns for the Training Data

Bibliography

[1] H. Markowitz, 'Portfolio selection,' *The Journal of Finance*, vol. 07, pp. 77–91, Mar. 1952. DOI: https://doi.org/10.2307/2975974.

[2] A. Palczewski and J. Palczewski, 'Theoretical and empirical estimates of mean–variance portfolio sensitivity,' *European Journal of Operational Research*, vol. 234, no. 2, pp. 402–410, 2014. DOI: https://doi.org/10.1016/j.ejor.2013.04.018.

[3] Z. Marakbi, 'Mean-variance portfolio optimization: Challenging the role of traditional covariance estimation,' Master thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2016. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1060405/FULLTEXT01.pdf#chapter.1.

[4] MathWorks, *Machine learning*, uk.mathworks.com, Accessed: May 9, 2022. [Online]. Available: https://uk.mathworks.com/discovery/machine-learning.html.

[5] M. Britten-Jones, 'The sampling error in estimates of mean-variance efficient portfolio weights,' *The Journal of Finance*, vol. 54, no. 2, pp. 655–671, 1999. DOI: https://doi.org/10.1111/0022-1082.00120.

[6] J. Chen, (Mar 30, 2020), *Portfolio weights*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/p/portfolio-weight.asp.

[7] Z. Zhang, S. Zohren and S. Roberts, 'Deep learning for portfolio optimization,' *The Journal of Financial Data Science*, vol. 2, no. 4, pp. 8–20, 2020. DOI: 10.3905/jfds.2020.1.042.

[8] C. Payne, 'Consumer price inflation, uk:january 2022,' Office for National Statistics, Newport, United Kingdom, Accessed: May 8, 2022. [Online]. Available: https://www.ons.gov.uk/economy/inflationandpriceindices/bulletins/consumerpriceinflation/january2022.

[9] 'Consumer price index - march 2022,' Bureau of Labor Statistics, Washington, DC, United States, Accessed: May 8, 2022. [Online]. Available: https://www.bls.gov/news.release/pdf/cpi.pdf.

[10] B. Beers, (May 13, 2020), *Buy and hold definition*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/b/buyandhold.asp.

[11] J. Chen, (Dec 31, 2021), *Risk parity*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/r/risk-parity.asp.

[12] A. F. Perold, 'The capital asset pricing model,' *Journal of Economic Perspectives*, vol. 18, no. 3, pp. 3–24, Sep. 2004. DOI: 10.1257/0895330042162340.

[13] W. F. Sharpe, 'Mutual fund performance,' *The Journal of Business*, vol. 39, no. 1, pp. 119–138, 1966. [Online]. Available: http://www.jstor.org/stable/2351741.

[14] wikipedia, (Mar 13, 2022), *Sharpe ratio*, en.wikipedia.org, Accessed: May 9, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Sharpe_ratio.

[15] W. Kenton, (Mar 04, 2021), *The calmar ratio*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/c/calmarratio.asp.

[16] CFI, *Calmar ratio*, corporatefinanceinstitute.com, Accessed: May 9, 2022. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/calmar-ratio/.

[17] A. Hayes, (Mar 04, 2021), *Maximum drawdown (MDD)*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp.

[18] A, Hayes(Oct 30, 2021), *Volatility*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/v/volatility.asp.

[19] J. Fernando, (Oct 30, 2021), *Compound annual growth rate (CAGR)*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/c/cagr.asp.

[20] The Investopedia Team, (Sep 10, 2021), *Bear market guide*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/b/bearmarket.asp.

[21] J. L. Treynor, 'Toward a theory of market value of risky assets,' unpublished. DOI: https://ssrn.com/abstract=628187.

[22] W. F. Sharpe, 'Capital asset prices: A theory of market equilibrium under conditions of risk*,' *The Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964. DOI: https://doi.org/10.1111/j.1540-6261.1964.tb02865.x.

[23] J. Lintner, 'The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets,' *The Review of Economics and Statistics*, vol. 47, no. 1, pp. 13–37, 1965. [Online]. Available: http://www.jstor.org/stable/1924119.

[24] J. Mossin, ‘Equilibrium in a capital asset market,’ *Econometrica*, vol. 34, no. 4, pp. 768–783, 1966. [Online]. Available: http://www.jstor.org/stable/1910098.

[25] W. F. Sharpe, ‘A simplified model for portfolio analysis,’ *Management Science*, vol. 9, pp. 277–293, Jan. 1963. [Online]. Available: http://www.jstor.org/stable/2627407.

[26] E. F. FAMA and K. R. FRENCH, ‘The cross-section of expected stock returns,’ *the Journal of Finance*, vol. 47, no. 2, pp. 427–465, 1992. DOI: https://doi.org/10.1111/j.1540-6261.1992.tb04398.x.

[27] E. F. Fama and K. R. French, ‘The capital asset pricing model: Theory and evidence,’ *Journal of Economic Perspectives*, vol. 18, no. 3, pp. 25–46, Sep. 2004. DOI: 10.1257/0895330042162430.

[28] F. Black and R. Litterman, ‘Global portfolio optimisation,’ *Financial Analysts Journal*, vol. 48, pp. 28–43, Sep. 1992. [Online]. Available: http://www.jstor.org/stable/4479577.

[29] P. Xu, A. Chen and P. Tsui, ‘Black-litterman model,’ *Statistics*, vol. 157, no. 4, p. 29, 2008. [Online]. Available: https://www.stat.berkeley.edu/~nolan/vigre/reports/Black-Litterman.pdf.

[30] G. J. Alexander and A. M. Baptista, ‘A comparison of var and cvar constraints on portfolio selection with the mean-variance model,’ *Management Science*, vol. 50, no. 9, pp. 1261–1273, 2004. [Online]. Available: http://www.jstor.org/stable/30046232.

[31] R. Cont and D. Nitions, *Statistical properties of financial time series*, Unpublished.

[32] wikipedia,(Jan 29, 2022), *Extreme value theory*, en.wikipedia.org, Accessed: May 9, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Extreme_value_theory.

[33] J. Tamošaitienė, V. Yousefi and H. Tabasi, ‘Project portfolio construction using extreme value theory,’ *Sustainability*, vol. 13, no. 2, 2021. DOI: 10.3390/su13020855.

[34] B. M. Rom and K. W. Ferguson, ‘Post-modern portfolio theory comes of age,’ *Journal of investing*, vol. 3, no. 3, pp. 11–17, 1994. DOI: https://doi.org/10.3905/joi.2.4.27.

[35] K. Hodnett and H.-H. Hsieh, ‘Capital market theories: Market efficiency versus investor prospects,’ *International Business Economics Research Journal (IBER)*, vol. 11, no. 8, pp. 849–862, Aug. 2012. DOI: https://doi.org/10.19030/iber.v11i8.7163.

[36] D. Snow, ‘Machine learning in asset management—part 2: Portfolio construction—weight optimization,’ *The Journal of Financial Data Science*, vol. 2, no. 2, pp. 17–24, 2020. DOI: 10.3905/jfds.2020.1.029.

[37] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. de Freitas, 'Taking the human out of the loop: A review of bayesian optimization,' *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016. DOI: 10.1109/JPROC.2015.2494218.

[38] J. D. Fisher, D. Pettenuzzo and C. M. Carvalho, 'Optimal asset allocation with multivariate bayesian dynamic linear models,' *The Annals of Applied Statistics*, vol. 14, no. 1, pp. 299–338, 2020. [Online]. Available: https://www.brandeis.edu/economics/RePEc/brd/doc/Brandeis_WP123.pdf.

[39] D. Bauder, T. Bodnar, N. Parolya and W. Schmid, 'Bayesian mean–variance analysis: Optimal portfolio selection under parameter uncertainty,' *Quantitative Finance*, vol. 21, no. 2, pp. 221–242, 2021. DOI: 10.1080/14697688.2020.1748214.

[40] N. Nguyen, 'An analysis and implementation of the hidden markov model to technology stock prediction,' *Risks*, vol. 5, no. 4, 2017. DOI: 10.3390/risks5040062.

[41] R. H. Tarik Al-ani Guy Leonard Kouemou, and etc. in *Hidden Markov Models, Theory and Applications*, D. Przemyslaw, IntechOpen, 2011.

[42] L. Breuer, (2014), *Introduction to stochastic processes*, kent.ac.uk, Accessed: May 9, 2022. [Online]. Available: https://www.kent.ac.uk/smsas/personal/lb209/files/notes1.pdf.

[43] S. Frühwirth-Schnatter, *Finite Mixture and Markov Switching Models*, eng, ser. Springer Series in Statistics. New York, NY: Springer Science + Business Media, LLC, 2006.

[44] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, 'Stock price prediction using the arima model,' in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 106–112. DOI: 10.1109/UKSim.2014.67.

[45] Z. Hu, J. Zhu and K. Tse, 'Stocks market prediction using support vector machine,' in *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 2, 2013, pp. 115–118. DOI: 10.1109/ICIII.2013.6703096.

[46] V. Kecman, 'Support vector machines – an introduction,' in May 2005, vol. 177, pp. 605–605. DOI: 10.1007/10984697_1.

[47] S. Mehtab and J. Sen, 'Stock price prediction using convolutional neural networks on a multivariate time series,' Aug. 2021. DOI: 10.36227/techrxiv.15088734.v1.

[48] J. Sen, A. Dutta and S. Mehtab, 'Stock portfolio optimization using a deep learning lstm model,' in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, 2021, pp. 263–271. DOI: 10.1109/MysuruCon52639.2021.9641662.

[49] J. Heaton, N. Polson and J. Witte, 'Deep learning for finance: Deep portfolios,' *Applied Stochastic Models in Business and Industry*, vol. 33, Oct. 2016. DOI: 10.1002/asmb.2209.

[50] I. Goodfellow, Y. Bengio and A. Courville, "Autoencoders," in *Deep learning*, MIT Press, 2016, pp. 499-523.

[51] B. M. Henrique, V. A. Sobreiro and H. Kimura, 'Stock price prediction using support vector regression on daily and up to the minute prices,' *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018. DOI: https://doi.org/10.1016/j.jfds.2018.04.003.

[52] S. Wadi, M. Almasarweh, A. A. Alsaraireh and J. Aqaba, 'Predicting closed price time series data using arima model,' *Modern Applied Science*, vol. 12, no. 11, pp. 181–5, 2018. DOI: doi:10.5539/mas.v12n11p181.

[53] B. de Wit, 'Stock prediction using a hidden markov model versus a long short-term memory,' Bachelor's Thesis, University of Groningen, Groningen, Netherlands, Ph.D. dissertation, 2019. [Online]. Available: https://fse.studenttheses.ub.rug.nl/20683/1/AI_BA_2019_Bram.pdf.

[54] G. K. Kanji, *100 Statistical Tests*, 3rd ed. London: SAGE Publications Ltd, 2006.

[55] P. Samuels and M. Gilchrist, 'Statistical hypothesis testing,' Birmingham City University, Birmingham, United Kingdom, Tech. Rep. stcp-gilchristsamuels-1, 2014, Accessed: May 9, 2022. [Online]. Available: https://www.researchgate.net/publication/275018715_Statistical_Hypothesis_Testing.

[56] A. Hayes, (Dec 01, 2021), *Wilcoxon test*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/w/wilcoxon-test.asp.

[57] wikipedia, (Mar 25, 2022), *Friedman Test*, en.wikipedia.org, Accessed: May 9, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Friedman_test.

[58] B. B. Gerstman, (May 10, 2016), *Paired samples*, sjsu.edu, Accessed: May 9, 2022. [Online]. Available: https://www.sjsu.edu/faculty/gerstman/StatPrimer/paired.pdf.

[59] B. B. Gerstman, (Aug 9, 2006), *Paired Samples*, sjsu.edu, Accessed: May 9, 2022. [Online]. Available: https://www.sjsu.edu/faculty/gerstman/StatPrimer/paired.pdf.

[60] N. Lioudis, (Apr 30, 2021), *Understanding the sharpe ratio*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/articles/07/sharpe_ratio.asp.

[61] J. Chen, (Mar 07, 2022), *Bear market guide*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/b/bearmarket.asp.

[62] J. Chen, (June 25, 2019), *Equity curve*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/e/equity-curve.asp.

[63] W. Kenton, (Mar 22, 2022), *Rate of return (RoR)*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/r/rateofreturn.asp.

[64] J. Chen, (Apr 26, 2020), *Rebalancing*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/r/rebalancing.asp.

[65] J. Chen, (Feb 26, 2022), *Exchange-traded fund (ETF) guide*, investopedia.com, Accessed: May 9, 2022. [Online]. Available: https://www.investopedia.com/terms/e/etf.asp.

[66] S. Prabhakaran, (Nov 2, 2019), *Augmented dickey fuller test (ADF test) – must read guide*, machinelearningplus.com, Accessed: May 9, 2022. [Online]. Available: https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/.

[67] Duke University, *Stationarity and differencing*, duke.edu, Accessed: May 9, 2022. [Online]. Available: https://people.duke.edu/~rnau/411diff.htm.

[68] M. Halford, (Feb 26, 2020), *Bayesian linear regression for practitioners*, maxhalford.github.io, Accessed: May 9, 2022. [Online]. Available: https://maxhalford.github.io/blog/bayesian-linear-regression/.

[69] B. S. B. Svetlozar T. Rachev John S. J. Hsu, and F. J. Fabozzi *Bayesian methods in finance*, Hoboken: John Wiley Sons, Incorporated, 2008.

[70] PyTorch, (2022), *LSTM*, pytorch.org, Accessed: May 9, 2022. [Online]. Available: https://pytorch.org/docs/master/generated/torch.nn.LSTM.html?highlight=lstm#torch.nn.LSTM.

[71] A. Rezaeian, *Time-series data analysis using LSTM (Tutorial)*, kaggle.com, Accessed: May 9, 2022. [Online]. Available: https://www.kaggle.com/code/amirrezaeian/time-series-data-analysis-using-lstm-tutorial/notebook.

[72] T. Y. Jonathon Price Alfred Wong, J. Mathews, T. Olorunniwo, (Dec 21, 2020), *Stochastic gradient descent)*, optimization.cbe. cornell.edu, Accessed: May 9, 2022. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=Stochastic_gradient_descent.