RESEARCH

INTELLIGENT INVESTMENT PORTFOLIO GENERATOR

2022

Andreas Katsoulieris

Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Intelligent Investment Portfolio Generator

Andreas Katsoulieris

May 2022

Abstract

Choosing the appropriate assets to include in an investment portfolio is a complex and tedious task that traditionally uses outdated assumptions and mathematical methods; as a result, many investors currently use the power of Artificial Intelligence to reinforce their investment strategy. This project introduces an Intelligent Investment Portfolio Generation, a platform that combines the power of Machine Learning with Finance for creating optimised investment portfolios. The platform is designed to accept the investor's risk tolerance, the maximum investment amount, and the investor's preferred Stock Market Sectors. The provided Stock Market Sectors in this project are composed of the Sectors classified by the Global Industry Classification Standard in addition to the Sustainable Sector, which includes only Sustainable companies based on their Environmental, Social, and corporate Governance scores. In addition to creating optimised investment portfolios, a feature for portfolio rebalancing was also implemented in the platform, which can be either performed on demand by the investor or based on a notification provided by the platform. A fully functional web-based User Interface is also designed to provide the investor with a simple and intuitive way to create and manage a Portfolio; the User Interface includes useful statistical metrics and graphs for assisting the investor's decisions.

Contents

1 Project Description

Investing can be described as the process of resource allocation or asset buying, most commonly using money, with the expectation that it will increase in value over time and provide profitable returns either in terms of money or capital gains. A successful investing strategy can provide both present and future financial security and ensure financial independence [1].

Building an investment portfolio which is a collection of financial investments, such as stocks, bonds, and index funds, requires choosing the assets to invest in. As an investor choosing the "right" assets, the ones with good returns, can be a pretty complex task.

1.1 Problem and Motivation

The current way of creating an investment portfolio is based on assumptions such as all investors have access to perfect information, and all investors have unlimited access to capital. Moreover, it can be argued that the employed mathematical methods do not necessarily represent the volatility of the 21st- century stock market [2]. The power of the Machine Learning algorithms can be utilised in the field of Finance as they can be used to make more informed investment decisions by exploiting patterns in the data that may not be recognisable by traditional financial methods.

1.2 Goals and Scope

The end goal of this project is to create an Intelligent Investment Portfolio Generator, a platform for managing and creating optimised investment portfolios. The main parameters of the platform include the investor's preferred Stock Market Sectors, the investor's risk tolerance, and the maximum amount of the investment.

The Stock Market Sectors in the platform consists of those classified by the Global Industry Classification Standard and the Sustainable Sector, which incorporates companies classified in terms of their Environmental, Social, and corporate Governance (ESG) scores. The power of Long Short-Term Memory (LSTMs) is utilised in this project, given their advantage on time-series data, to train the models based on each stock's daily return and make predictions based on them.

The parameters that the investor specified, together with the LSTM model's predictions, are used to calculate the weight of each stock, which is the proportion of the overall investment a specific stock comprises; using the calculated weights, a Portfolio allocation is created. A web-based User Interface (UI) is built to provide an intuitive and simple way for an investor to create and manage its Portfolio by providing useful statistics and graphs about the Portfolio and its performance. An additional benefit of the UI is that it allows all types of investors to create and manage their portfolios with only basic computer skills.

For this project's scope, all the assets in the generated Portfolio were obtained from the USA; the generated Portfolio did not take the asset's current price into account; the previous closing price was used instead. Additionally, all the models were trained with Technical Analysis, which means that irregular events like the COVID-19 pandemic that severely impacted the stock market were not considered.

2 Background and Report of Literature Research

2.1 Machine Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI), which is a branch of computer science with a focus on building systems that perform tasks that usually require human intelligence, the replication of the human thought process by a machine [3]. Turing Award winner John McCarthy first introduced the term Artificial Intelligence at a conference at Dartmouth College in 1956 [4].

ML was first introduced by the computer scientist Arthur Samuel in a journal called "Some studies in machine learning using the game of checkers" in 1959 while working at IBM [5]. It uses statistical methods to learn from a dataset by uncovering patterns in the data and performing specific tasks without being explicitly programmed to do so [6]. While ML is undoubtedly not a new concept, the development of frameworks, easy-to-use programming languages, and improvements in computing power have allowed for more profound research into the topic and extensive commercial use.

2.1.1 Training a Machine Learning Model

There are three main approaches for training an ML model: Supervised, Unsupervised, and Reinforcement Learning [7].

2.1.1.1 Supervised Learning

Supervised Learning uses labelled training data, which includes inputs and their correct outputs. The goal is to create a function that maps inputs to outputs to correctly predict the outputs of new input data, in the future. Supervised learning is performed by using the Classification and Regression algorithms.

In Classification, the algorithm categorises the data into specific classes; its objective is to find a function that can classify new data into the correct class. It is first trained using the training set, and based on recognised patterns, classifies unseen data by predicting the most likely class. Common classification algorithms are linear classifiers, Support Vector Machines (SVMs), k-nearest neighbour, Random Forest, and Naïve Bayes [8].

In Regression, the algorithm tries to understand the relationship between the dependent and independent variables; its objective is to find a function that can predict the dependent variable using the independent one. It aims to find the correlation and understand the linking relationship between the dependent and independent variables [9]. Common regression algorithms are Linear, Polynomial and Logistic Regression.

The model's accuracy is measured via the loss function, which quantifies the difference between the predicted and actual outputs; adjustments are then performed in order to adequately minimise the loss function [10].

2.1.1.2 Unsupervised Learning

Unsupervised Learning, unlike Supervised, uses unlabelled data, whereby only the inputs are given without the correct outputs; its goal is to find any hidden patterns in the training data. This technique can be particularly useful in real life, in which data on the outcomes are not always available; as Unsupervised Learning does not require labels, it may be more attractive to use as it would be both less costly and time-consuming than Unsupervised Learning [7]. One of the most common applications of Unsupervised Learning is clustering.

In Clustering, the algorithm looks for similar patterns in the data, for example, colour, shape and size, and tries to divide the data into groups based on the existence or inexistence of those patterns. While clustering can frequently overestimate the similarity among the groups, it helps with getting an insight into the natural grouping of the dataset. Common clustering algorithms include the K-Means, Centroid-based and Hierarchical-based clustering.

2.1.1.3 Reinforcement Learning

Reinforcement Learning (RL) uses an agent, an entity which by using sensors and actuators, can act upon an environment that "learns" by using trial and error and feedback so to achieve a goal [11]. The feedback is either positive, considered a reward, or negative, considered a punishment. RL aims to discover the optimised action model to maximise the agent's total cumulative reward. In view of finding this optimised action model, the agent has to choose between exploring new states and maximising its reward. This dilemma is known as the exploration-exploitation trade-off; the objective is not to immediately look for a reward but rather to maximise the cumulative reward throughout the duration of the training [11]. As the reward agent depends on all the past states and not just the current state, the time period is also very important. RL has virtually unlimited use cases; notably, they include self-driving cars and playing complex board games like Go. It is currently considered as the branch of AI that can closest mimic the intelligence of a human. Common RL algorithms include the Markov Decision Process and Q learning.

2.1.2 Overfitting and Underfitting

Two of the major problems that may occur during training a model are Overfitting and Underfitting.

2.1.2.1 Overfitting

Overfitting is a statistical term denoting the error in which the model closely fits a set of limited data points, resulting in the model being only useful in the initial dataset; in ML, this occurs once the model closely fits the training dataset.

When the model is either too complex or trains for an unreasonable amount of time, it may start to pick up noise or details that are only relevant to the training dataset. An overfitted model is unable to effectively generalise on unseen data, which means that it may not be able to make accurate predictions. Overfitting can be detected by calculating the error rate in both the training and testing dataset, which is a dataset of unseen data; if the error is low on the training dataset but high on the testing dataset, it is an indicator of overfitting.

Overfitting can be prevented using a number of techniques: Early stopping, in which the training stops early with the view of preventing picking up noise; Feature selection, in which the most important parameters or features are identified in the training data with the rest being discarded; Data augmentation, in which some noisy data may be added for the sake of making the model steadier, and Regularisation, in which a penalty is added to the parameters with the larger coefficients so to limit the model's variance [12].

2.1.2.2 Underfitting

Underfitting can be defined as the error whereby the model cannot capture with accuracy the input-output relationship; in ML, this occurs when the model is not complex enough.

Underfitting is the result of inadequate training time, training with an insufficient number of features, or too much regularisation. As an underfitted model is incapable of capturing the relation between the training dataset's features and target variables, the model, similarly with overfitting, cannot generalise and make accurate predictions. Detecting underfitting is considered easier than detecting overfitting as the error rate is high in both the testing dataset and the training set. Low variance and high bias are also treated as underfitting indicators[13].

Underfitting can be avoided by building a sufficiently complex model. This can be achieved by using more time for training, adding more features, and decreasing regularisation, which increases the variance in the model. When performing these techniques, special care should be taken to ensure that underfitting is not turned into overfitting.

2.1.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a subfield of Machine Learning inspired by the architecture of the human brain; they can be constructed by simulating how a network of biological neurons interact with each other [14]. By using this architecture, ANNs are adaptable and can be used to solve a huge variety of problems. The first ANN was invented by the American Psychologist and distinguished in the field of AI Frank Rosenblatt , while working at the Cornell Aeronautical Laboratory in 1958. Rosenblatt's ANN, which is considered the simplest form of an ANN, contains a single neuron and can only be used for linear separable problems [15]. As most problems are not linearly separable, the currently used ANNs are Multilayer Perceptrons (MLPs) [16].

MLPs are made up of layers, each comprising a collection of neurons. There are three types of layers, the input, the output and the hidden layer; for MPLs to be functional, they should contain at least one layer of each type. Each layer and neuron have a weight and a threshold value linked to them. A neuron is a place where computation happens; input neurons are activated by sensors that perceive the environment; other neurons are activated by weighted connections from previously connected active neurons; some neurons can also influence the environment by triggering actions [17].

The activation function determines whether a neuron should be fired; inputs are multiplied by their weights, weights control the importance of a variable, then the total is calculated, and bias is added [18]. Assuming that the activation function is a step function, the output is compared with the threshold; if the output is above or equal to the threshold value, then the neuron is "fired" it outputs 1; else, if the output is below the threshold, the neuron is not "fired" and it outputs 0. The unidirectional passing of data from one layer to the next one is what makes an ANN a feedforward network.

When using a step function as an activation, the network is linear, whereby non-linear problems cannot be addressed, signifying the need for non-linear activation functions. Notable examples of those types of functions include the Sigmoid, Tanh and the ReLU functions.

The Sigmoid function is an S-shaped function with Domain (-∞, +∞) and Range (0,1); it is defined as $A(x) = 1/(1 + e^{-x})$ [18] which means that even with either very big or very small inputs, its output will always be between 0 and 1. Calculating the Sigmoid function is considered computationally expensive and suffers from the problem of vanishing gradients; gradients can quickly get very close to 0 or 1, making minor updates on the weights during backpropagation as the derivative becomes very small. Softmax is a generalised version of the Sigmoid function, and it is used for multi-class Classification, while Sigmoid is used for binary.

Tanh function, which is defined as $A(x) = 2\,/(1 + e^{-2x}) - 1$ is also S-shaped like the Sigmoid function, with Domain (-∞, +∞) and Range (-1,1) [18]. Similar to Sigmoid it also suffers from the problem of vanishing gradients. Instead of gradients becoming very close to 0 and 1, they become very close to -1 and 1. However, it is Zero-Centered as it is symmetric at 0 [19], which helps with backpropagation.

ReLU function, which is short for Rectified Linear Unit, is currently the most used activation function. Similarly to the Sigmoid and Tahn function, it is not linear, but it consists of two linear functions, piecewise linear, with Domain (-∞, +∞) and Range [0, +∞). It defined as $A(x) = \max{(0, x)}$ [18], which denotes that all negative values are mapped to 0 while all positive to x. ReLU is considered less computationally expensive compared to the Sigmoid and Tanh function as it includes simpler mathematical calculations. ReLU allows for sparsity in the model; given that it allows mappings to 0, not all neurons are activated simultaneously, resulting in the model being less dense, which can step up the learning and make the model less costly.

While ReLU does not suffer from the problem of vanishing gradients as the derivative will always be 1 for positive numbers and 0 otherwise; it encounters another problem called "dying ReLU". While sparsity in the model is a desirable feature of ReLU, the problem occurs when many of the neurons start to become inactive during the training process. A variant of ReLU called Leaky ReLU can be used for solving, to some extent, this problem. Leaky ReLU can be defined as $A(x) = \max{(ax, x)}$ [18]; a is typically set to a value close to 0, such as 0.01 and never close to 1, as when $a = 1$, the activation function becomes $A(x) = x$, a linear function. Instead of mapping all negative values to 0, Leaky ReLU allows for small negative gradients when a unit is not active and saturated [20]. When the parameter a is adjusted adaptively during training, the activation function becomes parametric ReLU [21].

Another shortfall of ReLU is the problem of exploding gradients. Given that there is no bound at its output, large updates in the weights may occur during training, which results in the model becoming unstable and unable to learn. One of the techniques that can be used for addressing to a certain degree the exploding gradients problem is the Gradient

Clipping, in which a threshold value is introduced; if the norm of the gradient surpasses the threshold, rescaling is performed to keep the gradient small [22].

Backpropagation, which is considered a generalisation of the Least Mean Squared (LMSE) [23] and is short for backward propagation of errors, is a technique used to calibrate a Neural Network's weights by computing the gradient of an objective function with respect to the weights of the network. At first, it calculates the derivate of the objective function; next, it calculates the derivative of the objective function with respect to the output layers and then traverses backwards through the network by using the chain rule, calculating all the derivatives until it reaches the input layer. Once all the derivates have been computed, the gradients with respect to the weights are calculated for each layer [24]. Updates in the weights are then performed so as to minimise the objective function. Backpropagation is repeated until the error of the network has been adequately minimised [25].

To train an ANN, all the weights are initially set to small random numbers. The outputs of the networks are random; during training, the goal is to minimise the objective function, which is a function that calculates the error of the model. Backpropagation is used for fine-tuning the weights [26]. After the model reaches a solution, the model is validated using the validation dataset, a dataset that the model has never seen before, to tune the model's parameters further. Finally, the testing dataset is used to get the model's accuracy.

2.1.4 Deep Learning

Deep Learning (DL) is a subset of ML; it is a Deep Neural Network composed of multiple hidden layers. A Deep Neural Network can be defined as a Neural Network with more than three layers. The term Deep Learning was first introduced by the computer scientist Rina Dechter in a paper in 1986 [27].

Learning refers to the discovery of the weights that make the Neural Network manifest the desired behaviour; DL is about accurately assigning those weights across many computational stages [26]. DP aims to simulate the human brain by using those assigned weights, inputs from the dataset and biases.

While ML uses either data that are already structured or data that have been pre-processed to make them structured for making predictions, DL can easily take in and process unstructured data. The feature extraction in DL is performed automatically as the algorithms are capable of determining which features are going to be the most useful given a specific task. In ML, human intervention is needed to create this ordered list of features. Another key characteristic of DL is that, most of the time, it continues to improve with the addition of more data.

Using forward propagation, moving from one layer of the network to the next, and backpropagation, adjusting the weights of the model by moving backwards over the network layers, the DP algorithms become bit by bit more accurate.

DL and its applications are predicted to become more significant in the future [25] as recognising and discovering patterns is regarded as a vital part of progress and innovation.

2.1.5 Recurrent Neural Networks

One of the main problems of traditional ANNs is their inability to remember things about previous inputs they have received; they do not have persistence. One way of fixing that problem is to use Recurrent Neural Networks (RNNs), which can be defined as an ANN with loops inside it that allow information to persist [28].

Feedforward ANNs work on the assumption that a dataset with independent data is used for training. When the training dataset includes sequential data such as time series, RNNs are utilised, as they have the concept of memory which helps them store information from previous inputs used for generating future outputs [29].

The RNN layer has two weights, one for the input and one for the hidden layers; weights in an RNN model are shared across time [30]. Similar to ANNs, a specialised to sequential data, type of backpropagation called, Backpropagation Through Time (BPTT) is used for fine-tuning the weights. BPTT operates by unfolding all the input timesteps of the model. An unfolded RNN is considered very similar to Feedforward ANN; each timestep contains an input timestep, an output timestep and a copy of the network. The BPTT traverses through the RNN, using the chain rule, and calculates the derivative of the error with respect to the weights. Weights adjustments are then performed, informed by the calculated derivatives [31], to reduce the network's error. The model is folded again, and the weights are updated; the process is repeated until the network error has been minimised to satisfactory levels. BPTT can become very computationally expensive as the number of timesteps grows. During BPTT, problems such as vanishing and exploding gradients are possible to occur.

RNNs have different types and architectures; the main ones include, One to One, One to Many, Many to One and Many to Many.

One to One, also known as vanilla RNN, is considered the most basic RNN architecture. It only has a single input and output, and it can be used for general ML problems. It often suffers from the problem of exploding gradients.

One to Many has a single input but multiple outputs; it can be used in music generation where a music track is generated from a single note.

Many to One has multiple inputs but a single output; it can be used in sentiment analysis where the content of a text can be determined as positive, negative or neutral.

Many to Many has multiple inputs and outputs; it can be used in language translation where a text is transformed into another text.

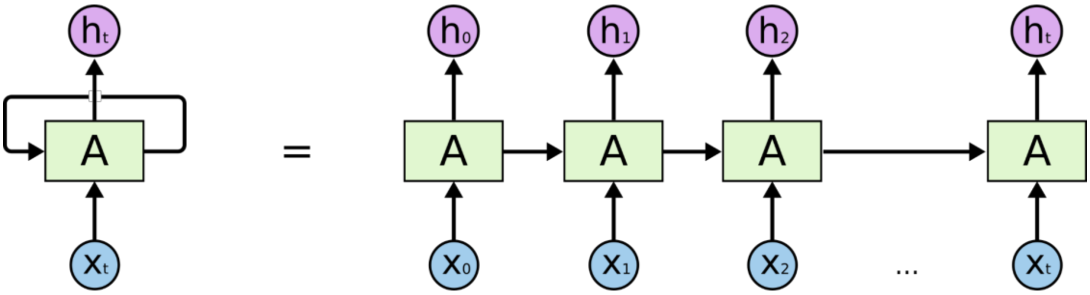


Figure 1 Unfolded RNN [32]

2.1.6 Long Short Term Memory

Long Short Term Memory (LSTMs) are a special kind of RNN proposed by the two computer scientists, Sepp Hochreiter and Juergen Schmidhuber, in 1997 as a solution to the problem of vanishing and exploding gradients [33]. LSTMs include a memory cell that can keep information for longer periods of time compared to RNNs. In a LSTM memory cell, the length of time that information persists is not fixed in advance but, instead, depends on the input data and the weights. It is also considered that traditional RNNs cannot learn when the time lag between the input and the output is greater than 5-10 time steps, while LSTMs can learn even with time lags that exceed 1000 timesteps by applying a constant error flow using the Constant Error Carousel (CEC), a unit inside the memory cell [34].

The key purpose of LSTM is to control the deletion or the addition of information via three gates: forget, input and output gate [34].

The forget gate decides whether to keep or discard the information. A Sigmoid function is applied to the current timestep and the previous hidden state; if the output of the Sigmoid is closer to 0, information is ignored, and if it is closer to 1, information is kept.

The input gate decides the information to update the current cell state with. A Sigmoid function is applied to the current timestep and the previous hidden state; the output decides the values to be updated. The Tanh function is applied to the same inputs with the Sigmoid to regulate the network. The outputs of both the Sigmoid and the Tanh functions are then multiplied to determine the information to update the current cell state.

The output gate decides what the output of each cell will be; a Sigmoid function is applied to the current timestep and the previous hidden state, a Tanh function is then applied to the output, and both outputs are multiplied to determine the final output.

As LSTM has the ability to memorise sequences through gates, and has an advantage in evaluating relationships in time series data, such as stock market data [35].

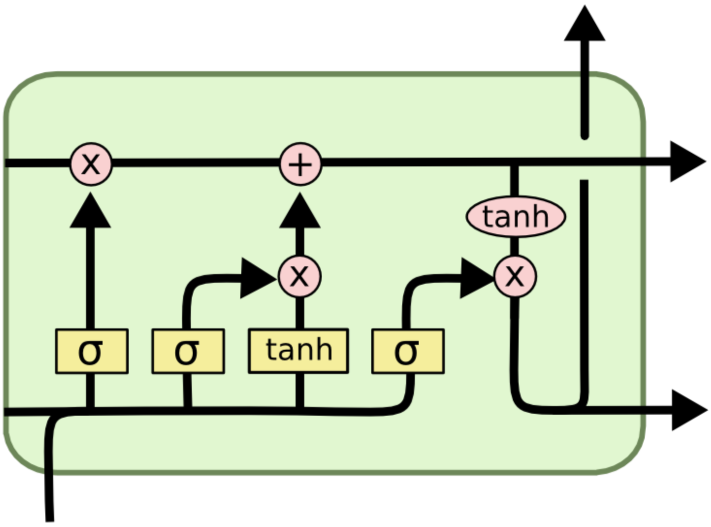


Figure 2 LSTM Architecture [32]

2.2 Finance

2.2.1 Stocks and the Stock Market

Predicting the trajectory of a stock is quite fascinating as, by making the correct predictions, one can potentially end up making a fortune. A stock can be defined as fractional ownership or equity at a company, typically issued in the form of shares [36]. The stock owner is qualified for access to the company's profits and assets proportionate to the amount of stocks he/she owns.

Stocks from publicly traded listed companies can be bought and sold at either an institutionalised stock exchange which is part of a country's stock market, or most commonly via a broker, a mediator between the stock exchange and the investor. USA's major stock exchanges consist of the New York Stock Exchange (NYSE) and the National Association of Securities Dealers Automated Quotations (NASDAQ). The stock market is regarded as a vital part of a country's economy as it allows access to trading to all types of investors and helps companies grow by quickly raising funds from the public.

2.2.2 Investing

Investing can be described as the action of buying assets which are expected to increase in value and generate a profit [1]. In order to find the right stocks to invest in, an investor has to perform an investment analysis which involves both researching and evaluating a stock. Investment analysis can be partitioned into different categories, Bottom-Up, Top-Down, Fundamental, Technical and Sentiment Analysis [37].

In Bottom-Up Analysis the investor focuses on evaluating a specific stock rather than the sector it belongs to or the whole economy. The objective is to find the best stock to invest in, disregarding the underlying trends in the economy. The investor takes a microeconomic approach.

In Top-Down Analysis, the investor focuses on examining the market, industry and economic trends rather than a specific company; the objective is to examine the global markets, at first sight, then the industries and sectors and in the end, a particular company, the investor takes a macroeconomic approach.

In Fundamental Analysis the investor focuses on finding mispriced stocks, and calculating their "fair market" value, most frequently by using a Bottom-Up Analysis. The objective is firstly to study the general state of the economy, secondly to study the stock industry and finally the particular stock to determine its "fair market" value.

In Technical Analysis, the investor focuses on examining the price and volume of a specific stock by finding patterns and statistical trends from the stock's trading activity to evaluate its strengths or weaknesses [38]. Technical analysis operates under the assumption that past trading activity can be a valuable indicator for predicting the stock's price movement when combined with suitable trading rules.

In Sentimental Analysis, the investor focuses on finding the stock's sentiment, the general point of view of investors towards the particular stock by applying Natural Language Processing (NLP) on news and articles related to the stock [38].

2.2.3 Investement Portfolio

Since an investor usually does not just hold an individual stock, an investment portfolio is required. An investment portfolio is a collection of financial investments, such as stocks, bonds, and index funds [39]. Although stocks, cash and bonds are considered the fundamental components of an investment portfolio, different types of assets can also be included, for example, private investments, real estate properties, collectable art and gold. The types of assets included in the portfolio are based on various factors, including the investor's preferences, investment horizon, and risk tolerance. An optimised portfolio can be described as the best portfolio out of a selection of portfolios according to the parameters specified by the investor.

One way of performing risk minimisation is diversification, which is the strategy of spreading an investment among various sectors to limit the investor's exposure to any particular risk or asset [40]. Its objective is to invest in contrasting areas that would behave oppositely to a specific event. This involves evening out the portfolio's unsystematic risk, a risk specific to a single company and not the whole market, as the negative performance of some assets will be counterbalanced by the positive performance of others. On average, diversification will reduce the Portfolio's volatility and risk and yield higher long-term returns.

Diversification can be achieved by leveraging the correlation of the assets, how one asset's price moves with regard to another. Asset prices with positive correlation (closer to +1) would tend to move in the same direction, while asset prices with negative correlation (closer to -1) would tend to move in the opposite direction. Diversification is effective if the assets in the portfolio are not perfectly correlated [41]; as they must react differently, move in opposite ways to market events.

Another practice of performing risk minimisation is rebalancing, which is the process of selling or buying assets periodically in a portfolio. The rebalancing period is decided based on different factors such as investment strategy, time constraints and transaction costs. Risk can easily tilt as a portfolio tends to increase its exposure to its best performing area, leaving the investor with more or less exposure to market areas than initially set. Rebalancing can mitigate this by realigning the weights, the proportion that each asset holds in the Portfolio [42].

2.2.4 Modern Portfolio Theory

One of the most popular techniques of asset selection is the Modern Portfolio Theory (MPT) [43], written by the Economic Science Nobel winner Harry Markowitz. Markowitz, described the impact the number of assets and their covariance relationship have on the diversification of a portfolio [44]. MPT works under a few assumptions, including that risk and return are directly linked, all investors are risk-averse, have access to perfect information, and have unlimited capital. Critics of Markowitz's MPT argue that those fundamental assumptions do not align with the conditions of the real world [2].

Another flaw of MPT is that it treats two portfolios that have similar returns and variances the same. Even though the first portfolio could have the same variance as the second one due to significant infrequent declines, the second portfolio's declines may be small but frequent. Most investors would choose the second portfolio as frequent, but small losses are easier to withstand.

2.2.5 ESG Investing

As MPT only aims to maximise risk-adjusted returns, other factors, for example, environmental ones, are ignored. Nevertheless, investing in companies based on their environmental impact has gained a great deal of momentum.

ESG is one of the most popular sustainability metrics. It stands for Environmental, Social and corporate Governance; Environmental examines how a company deals with environmental challenges such as climate change and waste; Social examines how a company deals with people, such as diversity and working conditions; and Governance examines how a company is being directed. It has been concluded that strengths in ESG increase the value of a firm and that concerns decrease it [45].

In Morgan Stanley [46], it is believed that the ESG factors are important during an investment process as they are an integral part of assessing a company's quality. They also support the idea that ESG directly impacts a company's competitive position and that events that negatively affect the company's ESG come along with significant losses for the company and, subsequently, for its investors. Moreover, they suggest that poor Governance can sabotage the success even of companies with long-term growth prospects and competitive advantages.

According to J.P. Morgan [47], the current pandemic could be a major turning point for ESG investing, as pandemics and environmental issues are viewed similarly impact-wise. They propose that the ESG market is a fast-growing one in every continent and that soon assets that follow the ESG principles will represent 44% of the total Assets Under Management (AUM). They also argue that ESG related funds more than doubled throughout 2019 and that during 2020 around 30 ESG-linked funds were launched globally.

In April of 2018, J.P. Morgan launched the ESG Index Suite, which currently has more than $13 billion in assets benchmarked into it. They suggest that ESG indices from JP Morgan outperformed their baselines in 2020 as they had reduced exposure to the 2020 stock market crash, specifically to the collapse of commodity-heavy sectors such as the energy sector, which includes fossil fuels like oil.

2.3 Machine Learning and Finance

ML has been applied, with positive results, into various quantitative Finance areas, including algorithmic trading, fraud detection, risk management and portfolio optimisation [48]. One of the key arguments for using ML in quantitative Finance is that it can spot non-linear relationships, which is required whenever the inputs from the data are not directly proportional to the outputs. Many traditional analysis techniques either assume a linear relationship or that transformation of a non-linear to a linear model is possible [49]. It is also considered that ML can identify and exploit patterns in the data that may not be recognisable by traditional financial methods. Another benefit of the ML that applies to Finance is the automation of repetitive tasks that traditionally require a human.

Applying ML to the stock market can help with making better and more informed trading decisions, as ML can process thousands of stock market data points simultaneously, something that is impossible to achieve by a human [50]. Moreover, its decision are not based on emotions that may affect the judgement of the trader.

During the application of ML into the stock market, the main interest lies in performing technical analysis to see if the algorithm can learn the underlying patterns with accuracy [51].

The integration of ML into the Financial world has helped the rise of the Fintech industry, a combination of the two words Finance and Technology. Fintech is defined as the application of innovative technological solutions into financial products [52]. Popular fintech products include Cryptocurrencies, Smart Contracts and mobile payment platforms. According to McKinsey [53] the use of digital banking products rose by 20 to 50 per cent in the first few months of 2020, and it is expected to rise even more.

JP Morgan has created a platform called Contract Intelligence (COIN). By using the power of ML, it is capable of interpreting legal documents and extracting important information from them. It has been reported that this task would typically require 360,000 hours per year from lawyers [54].

Robo-advisors are considered digital financial advisors that can help investors make smarter investment decisions. They are primarily used to manage an investment portfolio based on the investor's preferences, such as investment horizon and risk tolerance. Robo-advisors can also be used to recommend financial products such as an insurance plan or a loan, which may be personalised to a particular user.

Traditionally, fraud detection systems were based on a predefined set of rules that fraudsters could quickly bypass. Fraud is treated as a major problem in financial institutions, given that it costs the industry billions of dollars in losses per year. Fraud detection is one of the oldest successful applications of ML in the financial industry; ML can quickly detect anomalies in real-time by analysing a number of parameters and patterns such as the user's IP address, location and transaction history. Some exchanges also consider the utilisation of DL to prevent spoofing [55].

3 Methodology

The end goal of this project is to create an "Intelligent Investment Portfolio Generator". Two main components had to be built: the LSTM model and the optimised Portfolio.

3.1 Recurrent Neural Network Model

The proposed approach implemented an (LSTM) model. In order to train the LSTM, a number of steps were required; firstly, historical stock market data were collected from the web, and, secondly, pre-processing of the data took place. After having cleaned the dataset, it was split into three different datasets the training, testing and validation dataset. This was followed by feature extraction, in which the features to be fed into the LSTM were chosen. At this stage, the data were fed into the initially created LSTM, and training started. Hyperparameter tuning was then performed to optimise the LSTM model's parameters. The model was trained on previous returns of a specific stock, and it predicts future returns based on them.

The performance of the model was evaluated using two metrics, the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE). The former is defined as the squared root of the average squared difference between predicted and actual values, which measures the average error magnitude; the latter is defined as the average sum of the absolute difference between predicted and actual values divided by the actual values, which measures the average difference between the predicted and the actual values. RMSE is more sensitive to outliers than MAPE; both metrics are considered two of the most popular for measuring the accuracy of forecasting models.

3.2 Optimised Portfolio

The optimised Portfolio in each case was created by utilising the parameters asked from the investor and the predicted returns from the LSTM model. The platform asks the investor which sector he/she would like to invest in, the maximum amount of the investment and finally, the way the Portfolio is going to be optimised, and its point of view towards risk. After that, the program calculates the sample covariance of all the stocks in the sector the investor chose, the expected returns of those stocks based on the LSTM's predictions, and the weight that each stock will have in the Portfolio based on the investor's attitude towards risk. The weights are then cleaned, and an optimised portfolio is generated for the investor; the output of the platform includes the ticker name, an abbreviation used to identify a particular stock, and the number of stocks to buy. Finally, the remaining funds' amount is outputted to the investor.

The investment portfolio's performance was evaluated against the performance of the S&P 500 index, an index that tracks the performance of the 500 largest, in market capitalisation, companies traded in the US stock market. S&P 500 is regarded as an essential benchmark index for the overall US stock market.

4 Implementation

4.1 Tools

For the implementation of the LSTM, the Python programming language was utilised together with TensorFlow, an end-to-end open-source Machine Learning Library and Keras, an open-source ANN library. Data analysis and manipulation were conducted using the open-source library written for python pandas, and the UI was implemented using the Bootstrap front-end framework. The whole platform runs on the Python micro web framework Flask, with its data being stored in an SQL database.

4.2 Data Gathering

Pricing data were downloaded using the download() method of the yfinance library, which accepts the array of tickers together with the start and end date. The tickers array is obtained each time using a custom method that returns a dictionary with the sector's name as a key and its respective tickers stored as a value. The tickers in the Sustainable sectors are collected based on their ESG score acquired from the get_esg() method of the yesg library.

4.3 Data Pre-processing

After downloading the appropriate data, the null values were removed from the ticker's DataFrame, the daily percentage change of the prices was calculated and stored in a new column in the DataFrame the date was set as an index, and the rest of the DataFrame columns were removed. The DataFrame for each ticker was stored in a dictionary with the key set as the name of the ticker. Data were then passed to a custom function and were scaled between 0 and 1 using the MinMaxScaler() method of the scikit-learn library in order to ensure that updates in weights occur at the same rate during backpropagation. The dataset is then split with a ratio of 80:10:10 into three datasets the training, validation and testing dataset. The three datasets were further split into six arrays, x_train, y_train, x_val, y_val, x_test, and y_test, respectively. The number of timesteps was set to 60, which in this case is the number of previous dates. Finally, the custom method appends values, reshapes the six arrays and returns the x_train, y_train, x_val, y_val, x_test, and y_test, in order to be used by the LSTM model.

4.4 LSTM Model

The initial LSTM model was implemented following the instructions from the TensorFlow's website [56], using the RMSE and MAPE as guides hyperparameter tuning was performed in order to find the optimised model for the dataset, as described in the Evaluation and Testing Section.

A different model is created for each ticker and is trained using the x_train and y_train arrays; the accuracy and loss of each epoch are calculated using the x_val and y_val arrays. After the model finished training, it is saved in a specified directory in a .h5 file using the save() method of the Keras library. Using the trained model and the previous 60 timesteps for each ticker, the model predicts the 61^{st} value, the ticker's price movement at date 61. Finally, all the predicted values are stored in an array to be used as the portfolio generator.

4.5 Portfolio Generation

The optimised Portfolio was generated by a custom method. At first, the sample covariance between the tickers was calculated using the downloaded data from yfinance, and an Efficient Frontier was created based on the predictions from the LSTM model and the calculated sample covariance. An Efficient frontier aims to minimise the risk for a given return or maximise the return for a defined risk. The weights of each ticker in the portfolios were determined based on the optimisation technique specified by the investor; the optimisation techniques include low return, medium return, high return, and maximum Sharpe ratio. The weights were then "cleaned", which included rounding and zeroing tiny weights, and the latest prices of all the tickers were downloaded. Finally, an optimised portfolio allocation was generated based on the "cleaned" weights, the latest prices and the maximum investment value specified by the investor. Special care had to be given to creating a custom function for days the stock market was closed, as yfinance would fail to download any data. Finally, the method returned the optimised portfolio allocation together with expected yearly return and the leftover amount.

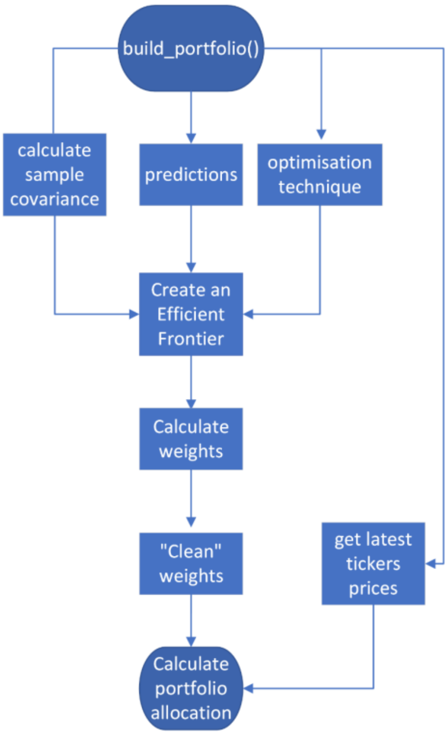

build_portfolio()
calculate sample covariance
predictions
optimisation technique
Create an Efficient Frontier
Calculate weights
"Clean" weights
get latest tickers prices
Calculate portfolio allocation

Figure 3 Flowchart of the portfolio creation process

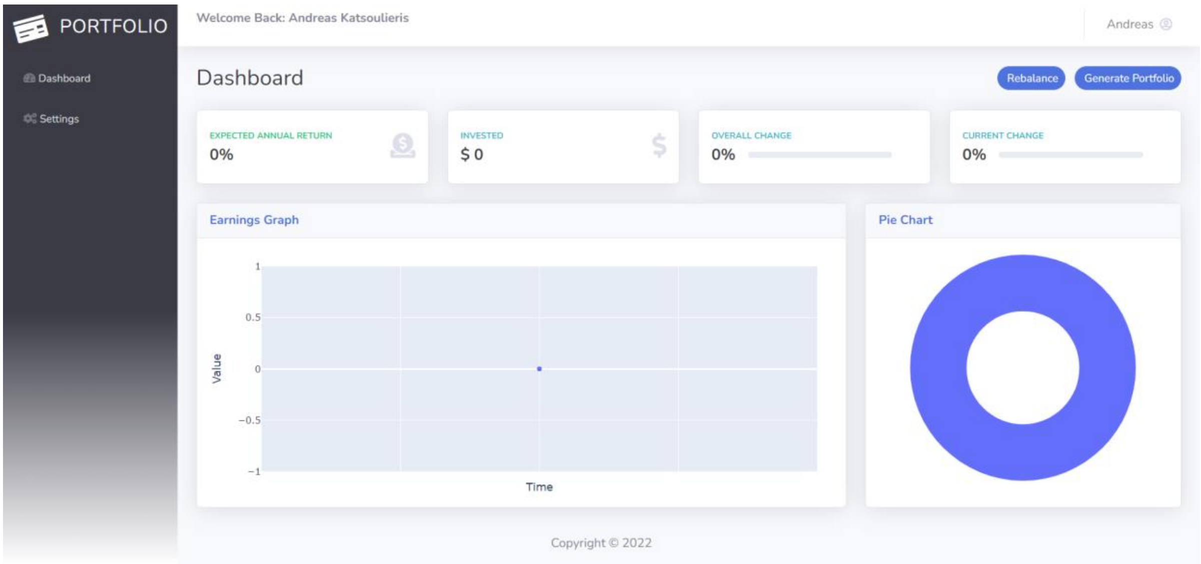
4.6 Rebalancing

For this project's scope, rebalancing was set to be performed every six months, which is when the investor receives a notification to rebalance. Rebalancing was performed by a custom method which accepts the allocation of the initial Portfolio, its optimisation technique, and its current total value. Similar to the optimised portfolio generation the sample covariance between the tickers in the initial allocation was calculated, new models were created to make predictions, and an Efficient frontier was generated. Weights were determined based on the optimisation technique of the initial Portfolio and were "cleaned". A new allocation was then created based on the weights, the latest prices of the tickers, and the current value of the Portfolio. The difference between the amount of each ticker in the new allocation and the initial allocation is calculated and stored in a dictionary as a value with a key being the ticker. Finally, the dictionary was returned by the method.

4.7 Platform

The platform has been set to open to the login page when it first starts up; if the investor does not have an account, it can create one by clicking the create an account button. The sign-up page asks for the investor's First and Last Name, his/her email address which the platform checks that it does not already exist, and two times for the password to verify that the investor typed the correct password. Assuming that information given by the investor passed all the safety checks, a new user object is created, which stores the hashed version of the password together with the First Name, Last Name and the email address at the Users table of the SQL database, and the investor is logged to the platform.

The login page works by checking the email and password the investor provides against the database and logs the investor in if a match is found. When the investor logs in for the first time he/she presented with the simplified version of the main page as a portfolio has not been generated yet, such as "figure 4". The overall change represents the percentage change since the creation date, while the current change represents the change since the investor last logged in.



Welcome Back: Andreas Katsoulieris

Dashboard

EXPECTED ANNUAL RETURN 0%

INVESTED $ 0

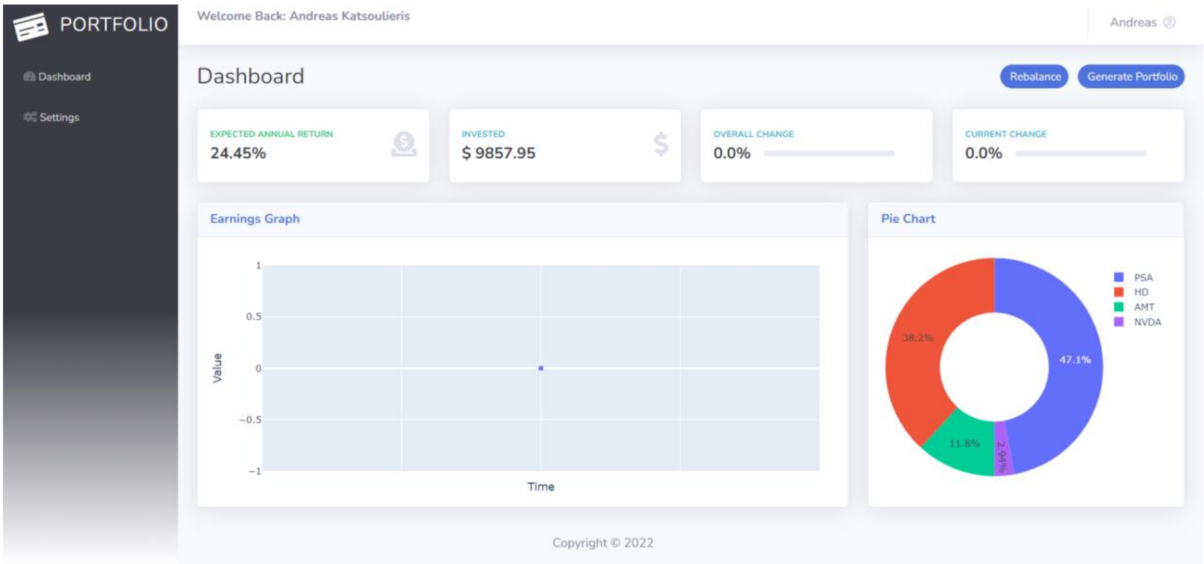
OVERALL CHANGE 0%

CURRENT CHANGE 0%

Copyright © 2022

Figure 4 Main page without generated Portfolio

For creating a portfolio, the investor must click the Generate Portfolio button, which at first will be present the Sectors the investor can invest in, then with its preferred optimisation technique and finally, the maximum investment amount of the Portfolio. Those parameters are then fed to the method that generates the Portfolio. Its output is presented to the investor in a table like format together with the leftover amount; ("see Appendix A"). All the parameters, outputs of the portfolio generation date of creation and graphs are stored at a new portfolio object in the Portfolio table of the database, converting when necessary to JSON strings; if the investor had created a portfolio before it is first deleted from the Portfolio table before creating a new one. When the investor clicks the finished button, he/she is presented with an updated main page, such as "figure 5".



Welcome Back: Andreas Katsoulieris

EXPECTED ANNUAL RETURN 24.45%

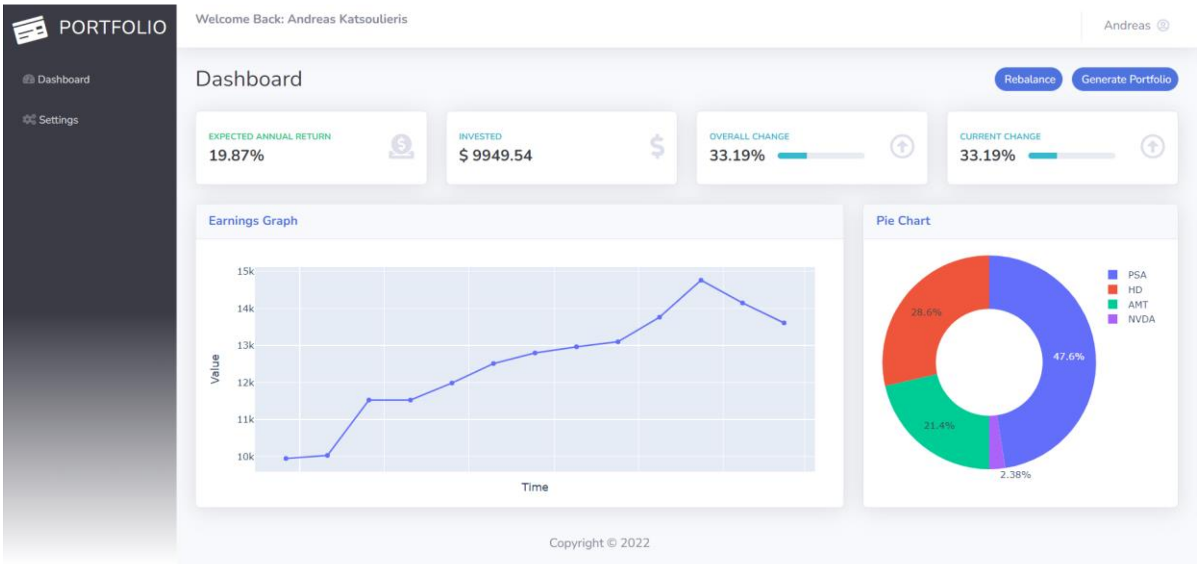
INVESTED $ 9857.95

OVERALL CHANGE 0.0%

CURRENT CHANGE 0.0%

Figure 5 Main page After Creating a Portfolio

As the Portfolio was just created, the Earnings graph has not yet been created, and both overall and current change is 0; for a portfolio that was created a year ago ("see figure 6").



Welcome Back: Andreas Katsoulieris

EXPECTED ANNUAL RETURN 19.87%

INVESTED $ 9949.54

OVERALL CHANGE 33.19%

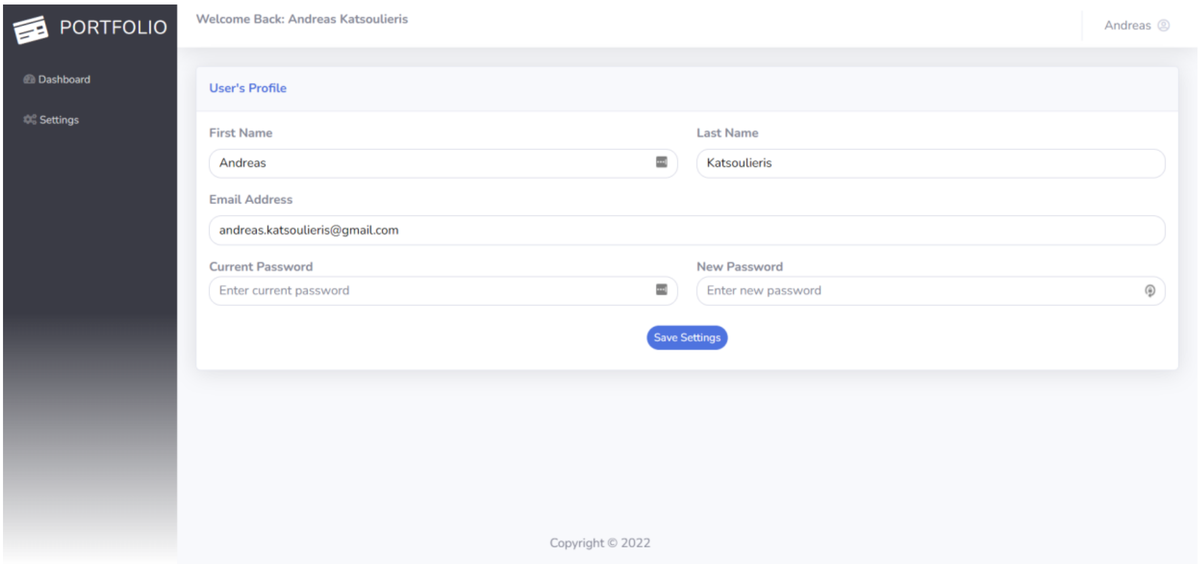
CURRENT CHANGE 33.19%

Figure 6 Main page 1-year after creating Portfolio

All graphs in this project were generated using the Plotly library. The data in the Earnings Graph was obtained using a custom function that calculates the total portfolio value from the date of creation to the current date.

Assuming that a portfolio has been generated, rebalancing can be performed at the Portfolio at any time by clicking the "Rebalance" button. The custom method for rebalancing is called, and its output will be presented to the investor in a similar format as was presented when creating the initial Portfolio ("see Appendix A"). The new output and the date of rebalancing will replace the initial Portfolio in the database.

The secondary page in the platform is the Settings ("see figure 7"), in which the investor can change his/her profile information and its password, assuming that correct values are given to each field; for example, the current password should match with the one stored in the database, the new password cannot be empty, and the email cannot be same as one already existing in the database. The name in the navigation bar is updated automatically as soon as the investor clicks the "Save Settings" button.



Welcome Back: Andreas Katsoulieris

User's Profile

First Name: Andreas

Last Name: Katsoulieris

Email Address: andreas.katsoulieris@gmail.com

Current Password: Enter current password

New Password: Enter new password

Copyright © 2022

Figure 7 Settings Page

5 Evaluation and Testing

Once the initial training was completed, the LSTM model's efficiency was analysed by calculating for selected stocks the RMSE, and MAPE scores, using the test dataset, and by creating the appropriate graphs; adjustments were made to the model until both RMSE and MAPE were at satisfactory levels. For the evaluation of the optimised Portfolio, multiple portfolios were created with different investment parameters to simulate the 1-year of investment. All the portfolios were created on 19-02-21; the performance of the Portfolio was evaluated by creating historical price graphs and by using the S&P 500 as a benchmark. During development, additional tests were conducted to ensure that this project works as expected, even in edge cases, such as when the stock market is closed.

5.1 LSTM Model

After implementing the first LSTM model, both RMSE and MAPE were calculated for a number of selected stocks. Given that the model was preliminary and relatively simple the metrics were much higher compared to the acceptable range for the dataset used. The results are displayed in "Table 1". A MAPE score close to 100 implies that errors are greater compared to the actual values, while a MAPE score close to 10 indicates that the model has very good accuracy and its results can be used with relatively high confidence.

Stock	RMSE	MAPE
APPL	53.97	100.00
MSFT	51.57	100.00
NVDA	39.84	100.00
JPM	38.73	100.00
EQIX	49.58	100.00
HD	59.16	100.00

Table 1 Metrics of the Initial Model

An additional layer was added to the LSTM network without much of an improvement in the results of the two metrics, results are displayed in "Table 2".

Stock	RMSE	MAPE
APPL	53.27	100.00
MSFT	51.44	100.00
NVDA	39.56	100.00
JPM	38.62	100.00
EQIX	49.14	100.00
HD	58.87	100.00

Table 2 Metrics of the Initial Model with an additional layer

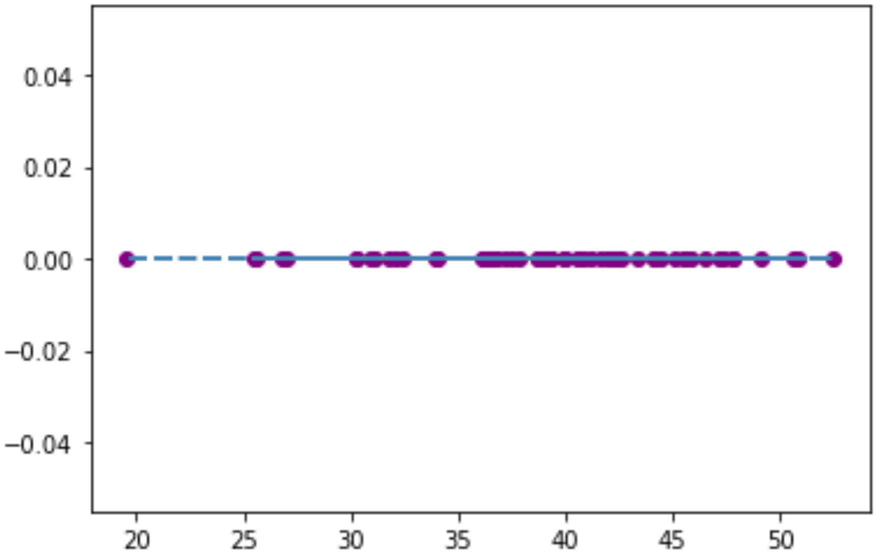


Figure 8 Line of Best Fit of the Initial Model with an additional layer

As all the points lie in a straight line ("see figure 8"), it can be deduced that the current model is unable to capture the underlying patterns of the training data and make predictions; the model underfits.

Different configurations of the activation function were also tried, and the results improved significantly("see Table 3") when the Tanh activation function was used at the output layer.

Stock	**RMSE**	**MAPE**
APPL	10.19	12.47
MSFT	9.62	12.68
NVDA	10.47	23.63
JPM	9.13	11.32
EQIX	9.55	12.16
HD	12.27	11.66

Table 3 Metrics of Initial Model with Tanh activation function

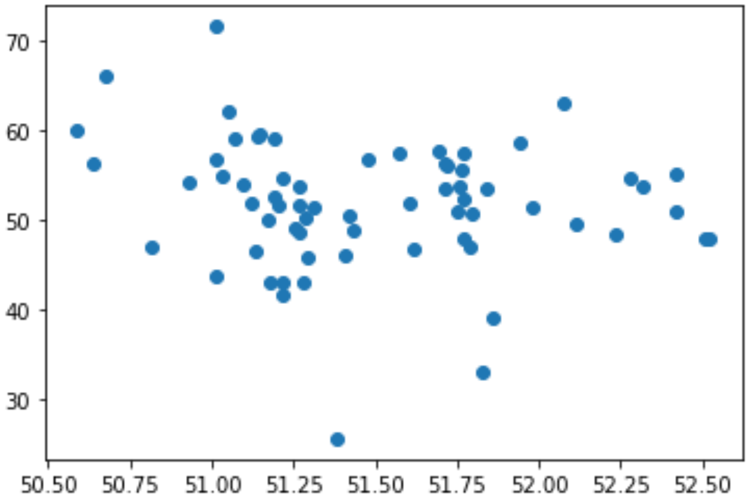


Figure 9 Scatter Graph of the Initial Model with Tanh activation function

Hyperparameter tuning was then performed on the model using Exhaustive Grid Search to further optimise the LSTM model; the grid of parameter values is displayed in "Table 4".

parameters	**values**
learning_rate	[0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4]
optimiser	['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']
batch_size	[10, 20, 40, 60, 80, 100]
epochs	[10, 50, 100]
activation	['softmax', 'softplus', 'softsign', 'relu', 'tanh', 'sigmoid', 'hard_sigmoid']
dropout_rate	[0.0, 0.1, 0.2, 0.3, 0.4]
neurons	[1, 10, 20, 30, 50, 70, 90, 100, 150]

Table 4 Hyperparameter Tuning

Optimised parameters according to the Grid Search are displayed "Table 5".

parameters	**values**
learning_rate	0.001
optimiser	Adam
batch_size	10
epochs	10
activation	relu
dropout_rate	0.2
neurons	50

Table 5 Hyperparameter Tuning Results

The results of RMSE and MAPE for this configuration of the RNN are displayed in "Table 6".

Stock	**RMSE**	**MAPE**
APPL	8.10	10.11
MSFT	7.52	10.57
NVDA	6.60	9.89
JPM	7.43	10.28
EQIX	7.07	9.52
HD	4.6	6.34

Table 6 Metrics of the Optimised Model

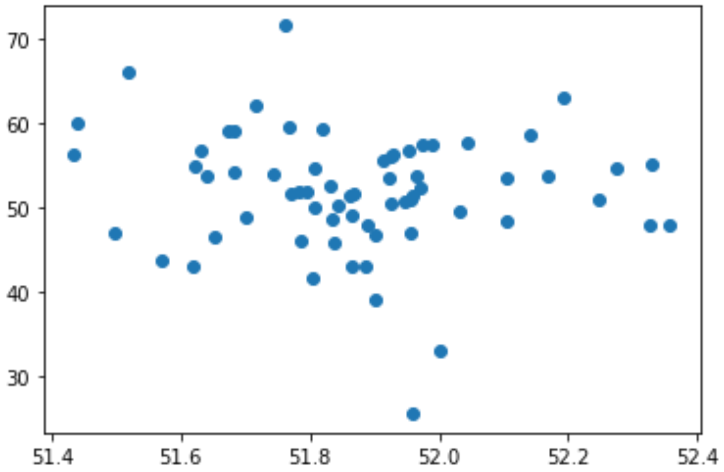


Figure 10 Scatter Graph of the Optimised Model

5.2 Optimised Portfolio

After generating multiple investment portfolios from different sectors with different risk exposures, graphs with the historical price change of the Portfolio and the S&P 500 at the same period were created. The amount of the S&P 500 was set as close as possible, rounded to 2 decimal places, to the investment amount. As an abstraction, all portfolios were built, with the maximum investment amount being set at $10,000.

5.2.1 Low-Risk Portfolios

5.2.1.1 Long Term Performance Evaluation

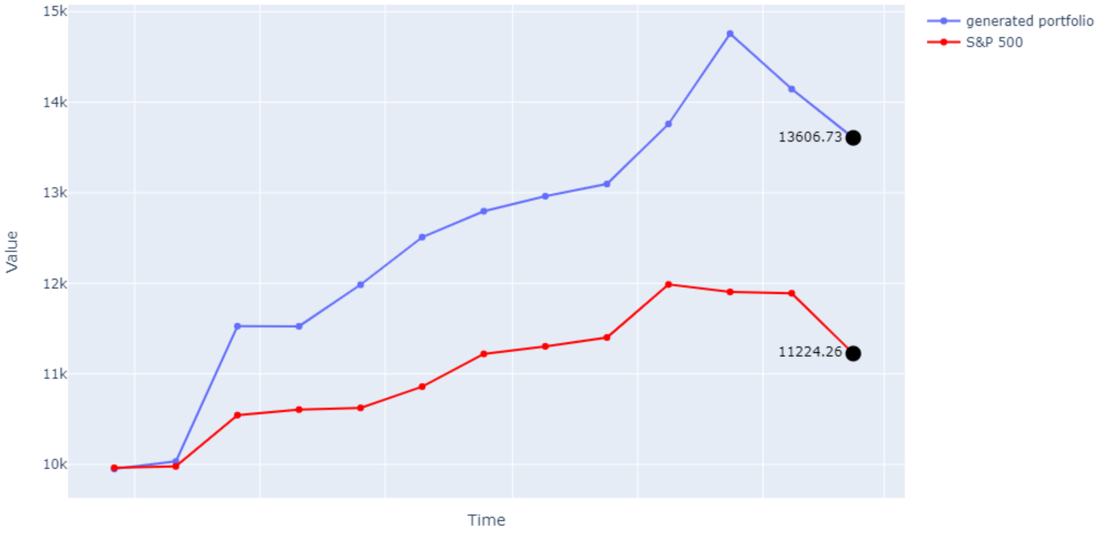


Figure 11 Twelve-Month Sustainable Low-Risk Portfolio Graph

ticker	amount
AMT	9
HD	12
PSA	20
NVDA	1
Invested	**9949.54**

Table 7 Twelve-Month Sustainable Low-Risk Portfolio Allocation

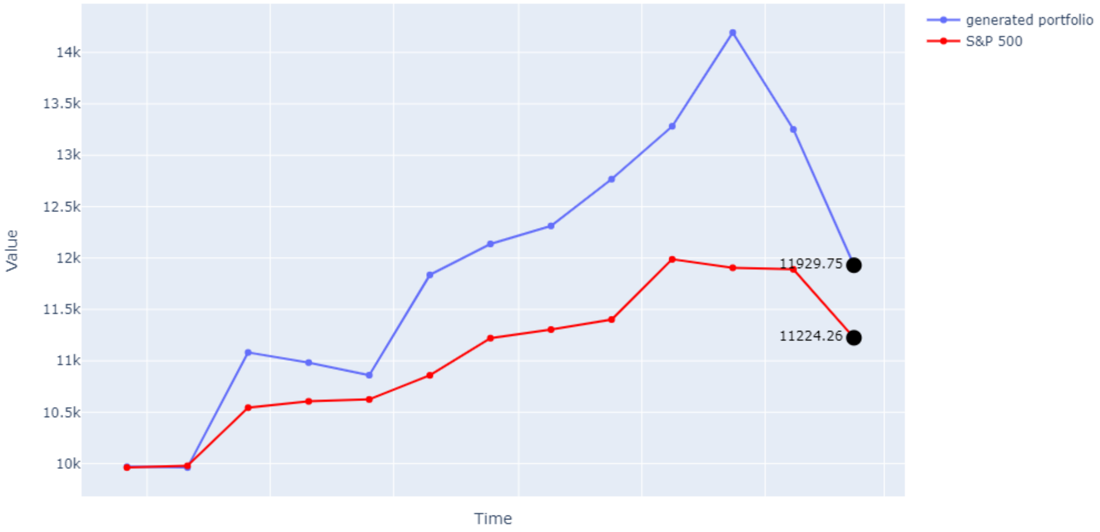


Figure 12 Twelve-Month Technology Low-Risk Portfolio Graph

ticker	amount
AAPL	4
TSM	14
CSCO	30
ORCL	48
ACN	13
Invested	**9970**

Table 8 Twelve-Month Technology Low-Risk Portfolio Allocation

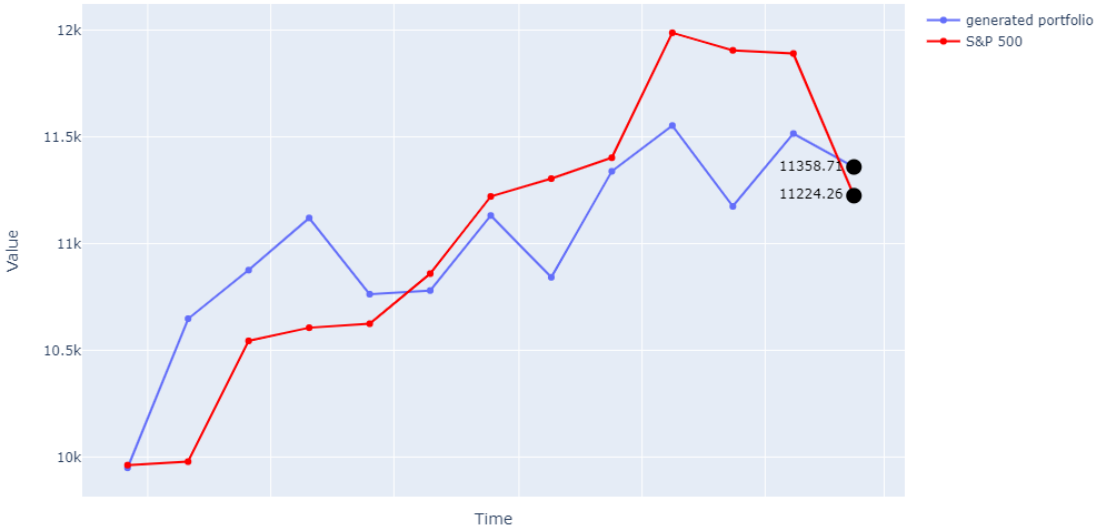


Figure 13 Twelve-Month Financial Services Low-Risk Portfolio Graph

ticker	amount
JPM	20
MA	16
GS	4
WFC	16
Invested	**9972.55**

Table 9 Twelve-Month Financial Services Low-Risk Portfolio Allocation

From the three generated graphs ("see figures 11,12 and 13"), it can be deduced that all sectors outperformed the S&P500 in the twelve-month evaluation period. The Sustainable sector was found to be the most profitable with +36.76%, and the Financial Services was the least with +13.90%.

5.2.1.2 Short Term Performance Evaluation

The three previous generated optimised portfolios were utilised and evaluated one week after the creation date to evaluate the short-term trading performance of the optimised portfolios.

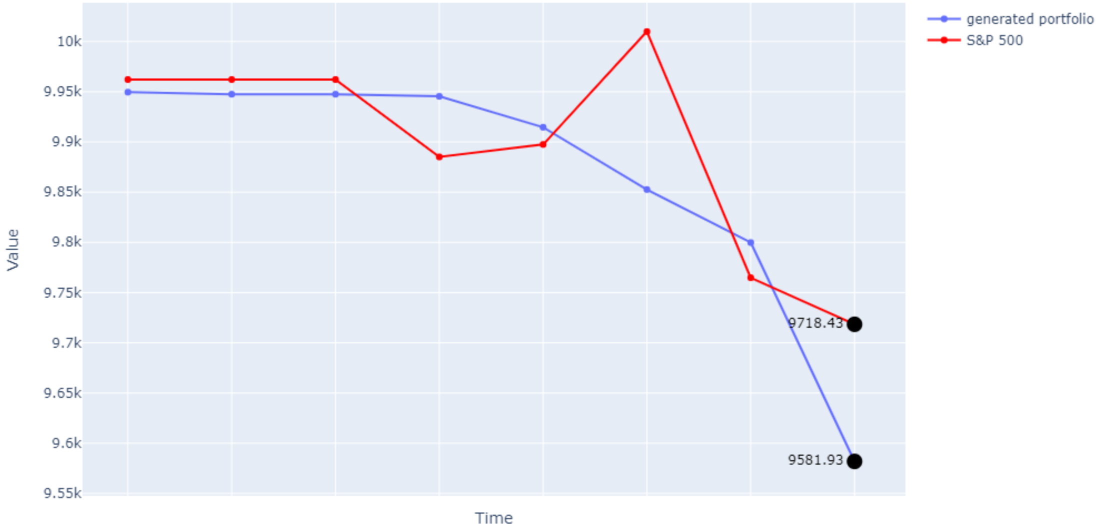


Figure 14   One-Week Sustainable Low-Risk Portfolio Graph

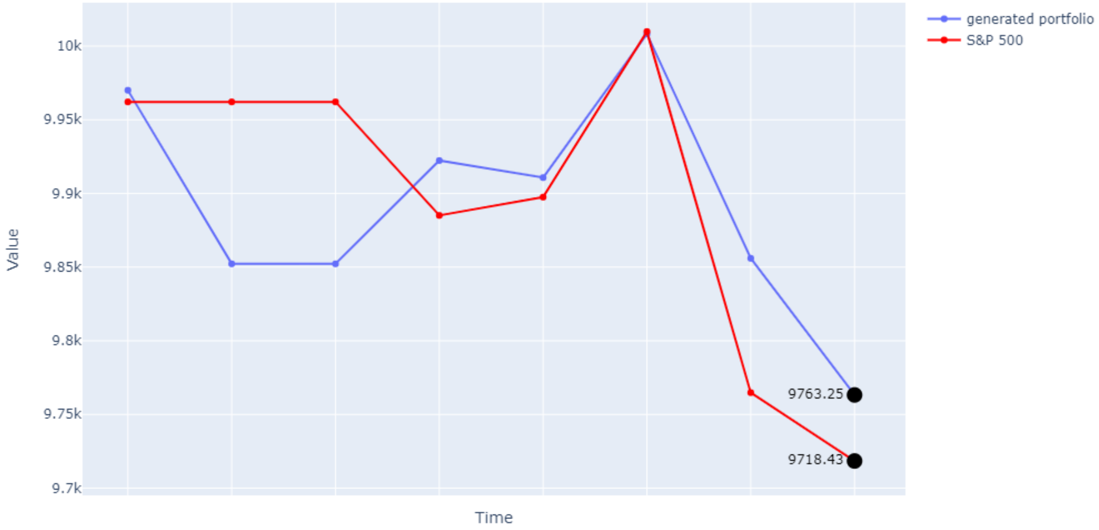


Figure 15 One-Week Technology Low-Risk Portfolio Graph

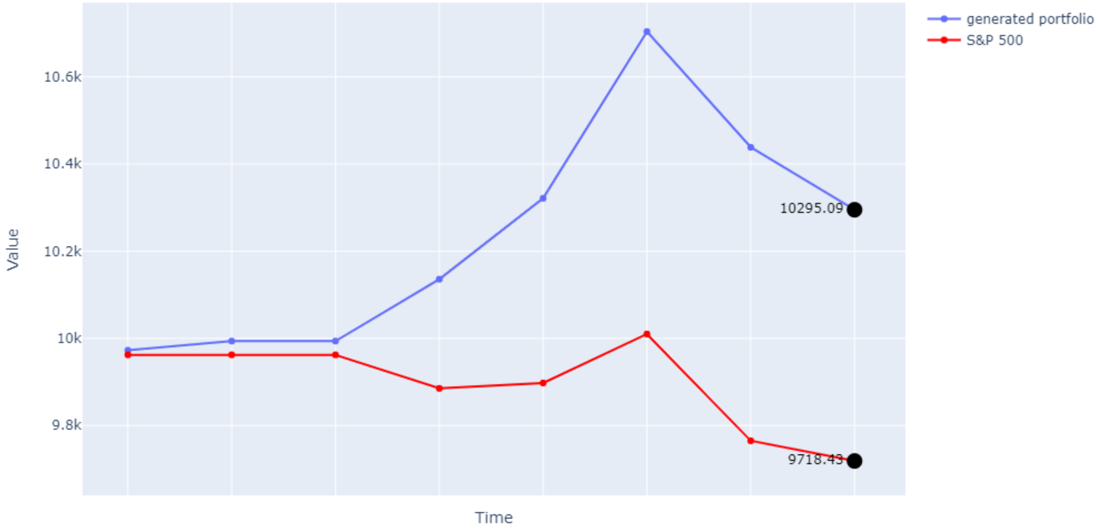


Figure 16 One-Week Financial Services Low-Risk Portfolio Graph

After analysing the three new evaluation graphs, ("see figures 11,12 and 13"), it can be derived that only the portfolios with Technology and Financial Services sectors managed to beat the S&P500 at this one week, with only the Financial Services portfolio being profitable with +3.23%.

5.2.2 High-Risk Portfolios

Portfolios with high risk were also created by adopting the same process of creating low-risk portfolios.

5.2.2.1 Long Term Performance Evaluation

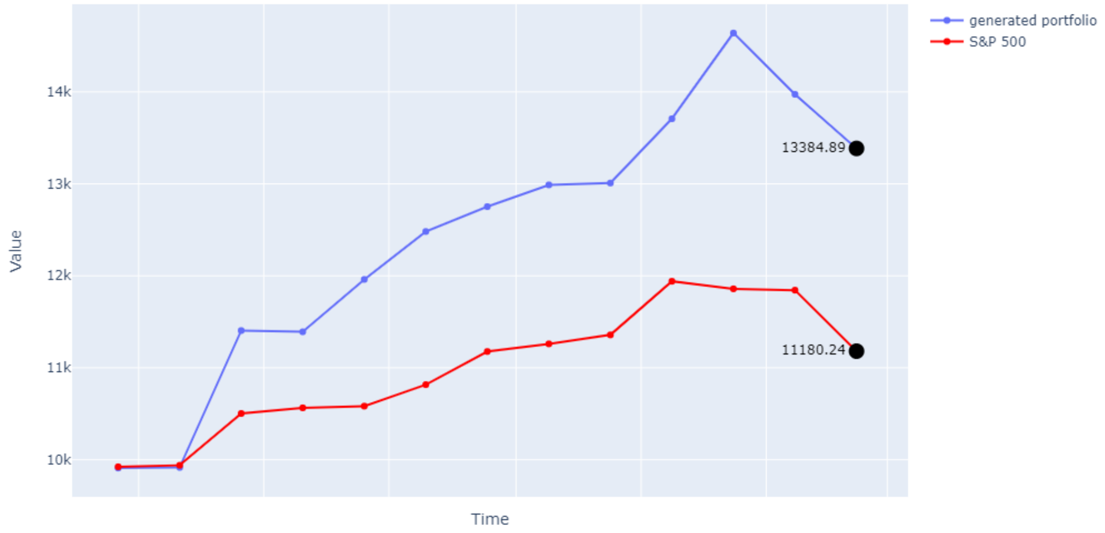


Figure 17 Twelve-Month Sustainable High-Risk Portfolio Graph

ticker	amount
AMT	12
HD	10
PSA	18
NVDA	3
Invested	**9909.38**

Table 10 Twelve-Month Sustainable High-Risk Portfolio Allocation

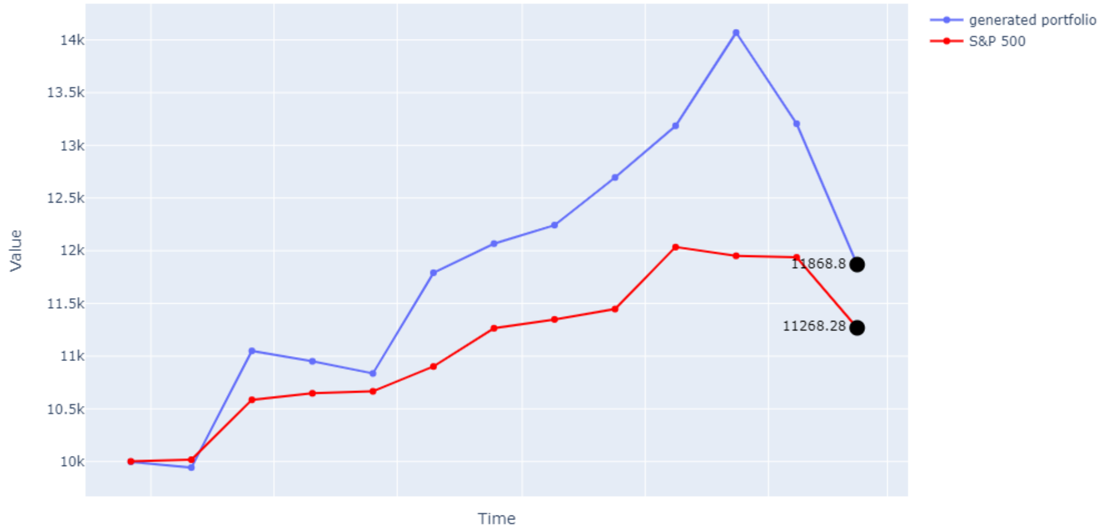


Figure 18 Twelve-Month Technology High-Risk Portfolio Graph

ticker	amount
AAPL	4
TSM	16
CSCO	29
ORCL	49
ACN	12
Invested	**9998.09**

Table 11 Twelve-Month Technology High-Risk Portfolio Allocation

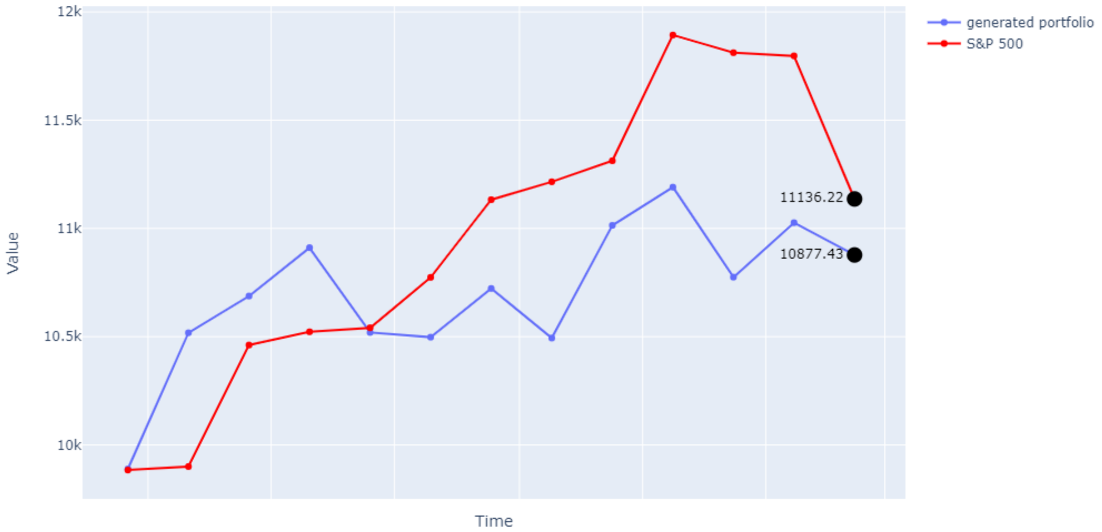


Figure 19 Twelve-Month Financial Services High-Risk Portfolio Graph

ticker	amount
JPM	30
MA	16
GS	1
Invested	**9888.97**

Table 12 Twelve-Month Financial Services High-Risk Portfolio Allocation

From the three graphs generated ("see figures 17,18 and 19"), it can be deduced that the Portfolio with Financial Services failed to beat the S&P500 while both portfolios in the Sustainable and Technology sector managed to surpass it; the most profitable Portfolio was in the Sustainable sector, with a profit of +35.07%.

5.2.2.2 Short Term Performance Evaluation

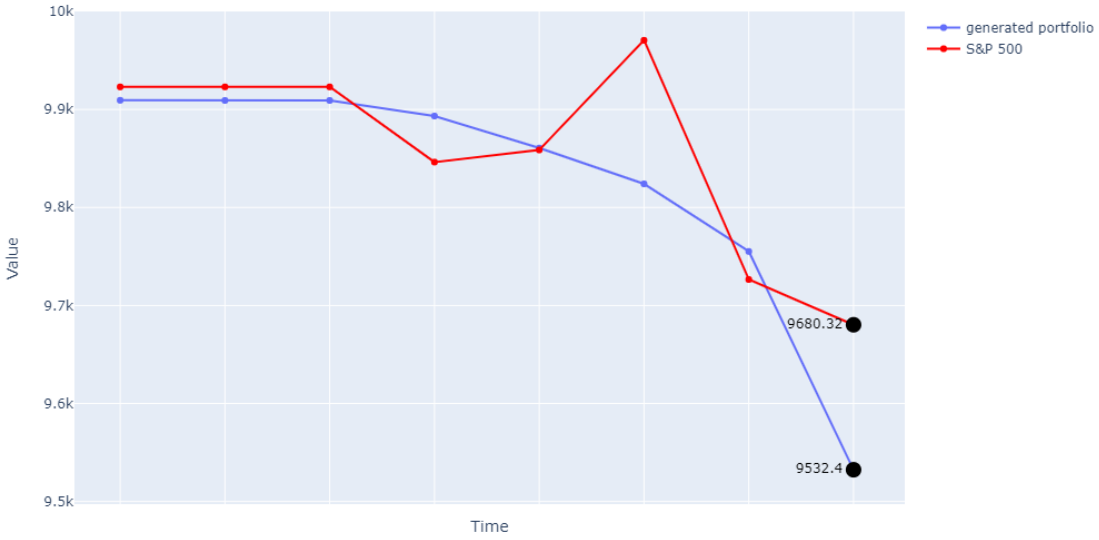


Figure 20 One-Week Sustainable High-Risk Portfolio Graph

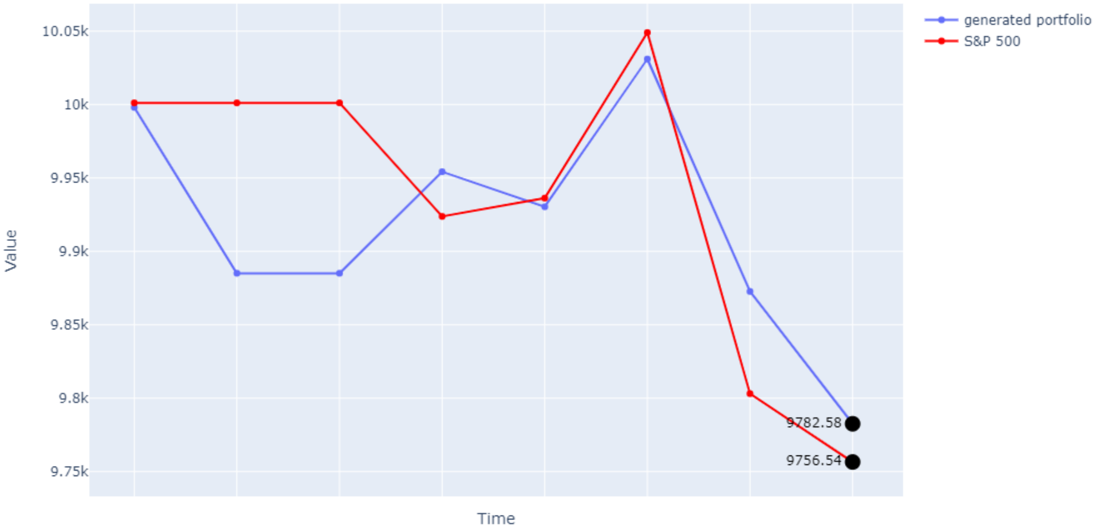


Figure 21 One-Week Technology High-Risk Portfolio Graph

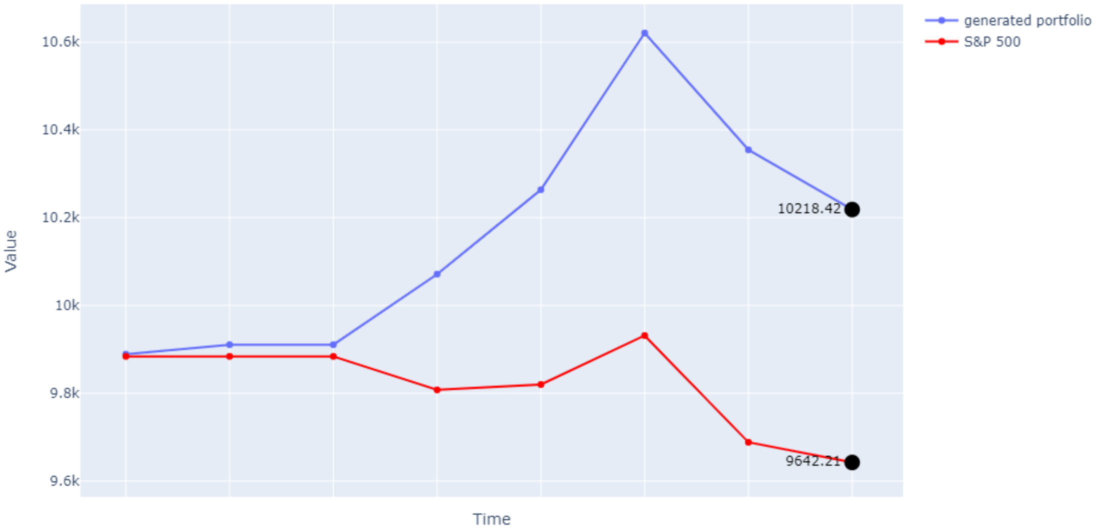


Figure 22 One-Week Financial Services High-Risk Portfolio Graph

The outcome of the three graphs ("see figures 20, 21 and 22") can be considered similar to the low-risk short term performance graphs, as again, only the portfolios in the Technology and Financial Services Sectors managed to beat the S&P 500, and the only profitable portfolio was in the Financial Services sector with + 3.33 %.

5.2.3 Further Analysis

The results of the above analysis can be used to extract a variety of information about the project and its scope.

Firstly the short-term performance of both low and high risk optimised portfolios is an indication that the generated portfolios are limited to long term investing; given that, for the scope of this project, only Technical Analysis has been performed, the model is unable to capture day-to-day events that may affect the stock market.

Comparing the low-risk with the high-risk portfolios, it can be deduced that all high-risk portfolios were more volatile, with the Portfolio in the Technology sector performing worst by -0.51% compared to the lower risk portfolio. The increases in the total portfolio value of the portfolios in the Sustainable and Financial Services sector were almost negligible, with +1.63% and +4.24%, respectively.

In addition to that, portfolios evaluated in the long-term with investments in the Sustainable sector had significantly higher returns than portfolios in the Technological and Financial sector, reflecting the current stock market trends.

5.3 Platform

The evaluation of the overall platform was conducted by testing different use cases. More specifically, the platform was tested on different browsers and operating systems to ensure functionality across different devices. In addition, the platform was tested on days that the stock market is closed, such as weekends and public holidays. Wrong inputs were also given to the platform, including negative numbers for the investment amount and null passwords, to test how it deals with them. Database testing was conducted on top of that to ensure that values were stored as they should and that passwords remained hashed. Usability testing was also conducted to ensure that the platform is simple to navigate and learn, even for users who have not used similar software before.

6 Project Management

6.1 Risk Assessment

The risk assessment of this project was conducted using a Risk Assessment Matrix in which each risk was assigned a probability from 1 to 5, with 1 being unlikely and 5 being most likely and an impact value again from 1 to 5, with 1 being insignificant and 5 being disastrous.

The final risk rating is calculated using the $risk\,rating = Probability * Impact$ formula.

Risk	Probability	Impact	Risk Rating	Action
Loss of Code	2	5	10	Regular commits in GitLab and backups on external hard drives
Loss of Writing Report	2	5	10	Regular backups in the cloud and backups on external hard drives
Bugs in the Code	3	4	12	Allocation of extra time
Compatibility issues	3	3	9	Reading documentation and online research
Hardware Limitations for training the models	3	3	9	Utilising the computers of ECS either virtually or in-person
Insufficient knowledge of Financial Concepts	3	4	12	Reading academic papers and seeking the advice of experts in the field
Contracting COVID-19	4	3	12	Applying for an Extension and allocating extra time

Table 13 Risk Assessment Table

6.2 Task Management

For the efficient management of the tasks, the initial Gant Chart was followed closely, together with daily to-do lists and weekly scrum meetings with the Supervisor of this project. The initial Gannt chart had to be revised to accommodate the period of illness due to COVID-19 and the extra time spent on code debugging.

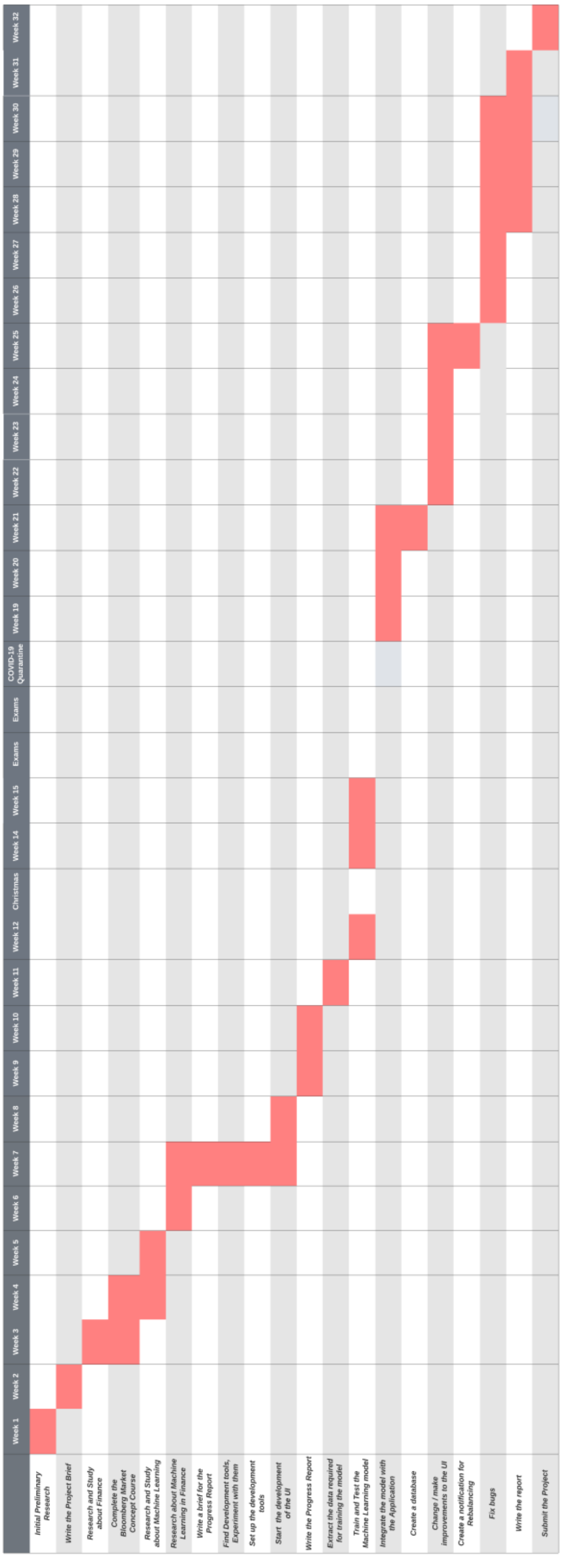


Figure 23 Revised Gannt Chart

7 Conclusions

This project successfully met the initial set objective of providing a platform for managing and creating optimised investment portfolios using the power of Machine Learning. The Evaluation found that long term investment portfolios created by the LSTM model with an average MAPE score of 9.45 and an average RMSE score of 6.88, successfully generated profit and beat the set benchmark S&P 500. In addition it was found that investing in Sustainable companies can lead to lucrative returns for the investor.

One of the project limitations is that only Technical Analysis is performed, meaning that events such as pandemics which may affect the stock market are not considered by the application. This adversely affects the performance of short term portfolios.

The main component of this project was its web-based UI, as it connects all the components of this project together and has the ability to run on a plethora of different operating systems. In addition, the UI provides useful statistics and graphs about the Portfolio and its performance to aid the investor's financial decision. An additional benefit of the UI is that it allows all types of investors to create and manage their portfolios with only basic computer skills required.

In conclusion, this project demonstrates how the field of Computer Science can be combined with the field of Finance to create a fully functional platform for managing and creating optimised investment portfolios.

7.1 Future work

Even though the main objective of the projects was successfully met, through future work, the model's accuracy could be improved by utilising Sentiment Analysis and by further tuning the model.

In addition to improvements in the prediction model, a few features could be enhanced in the platform, starting with the option of creating and managing multiple portfolios and adding more statistical information to assist the investor. The forget password capability, weekly emails about the performance of the Portfolio and the option of choosing a dark theme for the UI could also become a part of the platform.

Another component that could be added is the use of live stock prices instead of the previous day's closing price in the creation of the portfolios and their performance, together with the more stock markets, instead of only the US. As a final last step, the application could be deployed on a server instead of running it locally.

8 References

[1] Elvis Picardo, “Investing,” Apr. 30, 2021. https://www.investopedia.com/terms/i/investing.asp (accessed Nov. 28, 2021).

[2] M. E. Mangram, "A simplified perspective of the Markowitz portfolio theory," *Global journal of business research*, vol. 7, no. 1, pp. 59–70, 2013.

[3] A. Sabharwal and B. Selman, "S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition.," *Artif. Intell.*, vol. 175, pp. 935–937, Apr. 2011, doi: 10.1016/j.artint.2011.01.005.

[4] S. L. Andresen, "John McCarthy: father of AI," *IEEE Intelligent Systems*, vol. 17, no. 5, pp. 84–85, 2002, doi: 10.1109/MIS.2002.1039837.

[5] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 44, no. 1.2, pp. 206–226, 2000, doi: 10.1147/rd.441.0206.

[6] S. Emerson, R. Kennedy, L. O'Shea, and J. O'Brien, "Trends and applications of machine learning in quantitative finance," 2019.

[7] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[8] A. A. Soofi and A. Awan, "Classification techniques in machine learning: applications and issues," *Journal of Basic and Applied Sciences*, vol. 13, pp. 459–465, 2017.

[9] G. K. Uyanık and N. Güler, "A Study on Multiple Linear Regression Analysis," *Procedia - Social and Behavioral Sciences*, vol. 106, pp. 234–240, 2013, doi: https://doi.org/10.1016/j.sbspro.2013.12.027.

[10] F. Nie, H. Zhanxuan, and X. Li, "An investigation for loss functions widely used in machine learning," *Communications in Information and Systems*, vol. 18, pp. 37–52, Jan. 2018, doi: 10.4310/CIS.2018.v18.n1.a2.

[11] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.

[12] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, 2019, vol. 1168, no. 2, p. 022022.

[13] P. Cunningham and S. J. Delany, "Underestimation Bias and Underfitting in Machine Learning," *arXiv preprint arXiv:2005.09052*, 2020.

[14] A. Krogh, "What are artificial neural networks?," *Nature Biotechnology*, vol. 26, no. 2, pp. 195–197, 2008, doi: 10.1038/nbt1386.

[15] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organisation in the brain.," *Psychol Rev*, vol. 65, no. 6, p. 386, 1958.

[16] H. Taud and J. F. Mas, "Multilayer perceptron (MLP)," in *Geomatic approaches for modeling land change scenarios*, Springer, 2018, pp. 451–455.

[17] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[18] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A Review of Activation Function for Artificial Neural Network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2020, pp. 281–286. doi: 10.1109/SAMI48414.2020.9108717.

[19] L. Datta, "A survey on activation functions and their relation with xavier and he normal initialisation," *arXiv preprint arXiv:2004.06632*, 2020.

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, 2013, vol. 30, no. 1, p. 3.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[22] Y. Goldberg, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7909255

[23] K.-L. Du and M. N. S. Swamy, *Neural networks and statistical learning*. Springer Science & Business Media, 2013.

[24] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks: A Systematic Introduction*, R. Rojas, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 149–182. doi: 10.1007/978-3-642-61068-4_7.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[27] R. Dechter, *Learning While Searching in Constraint-Satisfaction-Problems.* 1986.

[28] S. Grossberg, "Recurrent neural networks," *Scholarpedia*, vol. 8, no. 2, p. 1888, 2013.

[29] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, doi: 10.1109/72.279181.

[30] R. DiPietro *et al.*, "Recognising Surgical Activities with Recurrent Neural Networks," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, 2016, pp. 551–558.

[31] T. P. Lillicrap and A. Santoro, "Backpropagation through time and the brain," *Current Opinion in Neurobiology*, vol. 55, pp. 82–89, 2019, doi: https://doi.org/10.1016/j.conb.2019.01.011.

[32] "Understanding LSTM Networks," Aug. 27, 2015. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Nov. 14, 2021).

[33] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[35] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[36] R. J. Teweles and E. S. Bradley, *The stock market*, vol. 64. John Wiley & Sons, 1998.

[37] Investment Analysis, "Investment Analysis," Aug. 25, 2021. https://www.investopedia.com/terms/i/investment-analysis.asp (accessed Apr. 14, 2022).

[38] A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," *Expert Systems with Applications*, vol. 135, pp. 60–70, 2019, doi: https://doi.org/10.1016/j.eswa.2019.06.014.

[39] Carla Tardi, "What Is a Portfolio?," Mar. 03, 2021. https://www.investopedia.com/terms/p/portfolio.asp (accessed Nov. 10, 2021).

[40] W. N. Goetzmann and A. Kumar, "Equity Portfolio Diversification," *Review of Finance*, vol. 12, no. 3, pp. 433–463, Jan. 2008, doi: 10.1093/rof/rfn005.

[41] Diversification, “Diversification,” Apr. 21, 2021. https://www.investopedia.com/terms/d/diversification.asp (accessed Apr. 14, 2022).

[42] A. Ang, *Asset management: A systematic approach to factor investing*. Oxford University Press, 2014.

[43] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952, doi: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x.

[44] W. Megginson, "A historical overview of research in finance," *Journal of finance*, vol. 39, no. 2, pp. 323–346, 1996.

[45] A. Fatemi, M. Glaum, and S. Kaiser, "ESG performance and firm value: The moderating role of disclosure," *Global Finance Journal*, vol. 38, pp. 45–64, Nov. 2018, doi: 10.1016/j.gfj.2017.03.001.

[46] K. Heugh and M. Fox, "ESG and the sustainability of competitive advantage," *Investment Insight*, 2017.

[47] J. P. Morgan, "Why COVID-19 could prove to be a major turning point for ESG investing," JP Morgan, 2020. Accessed: Nov. 15, 2021. [Online]. Available: https://www.jpmorgan.com/insights/research/covid-19-esg-investing

[48] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[49] M. López de Prado, "Beyond econometrics: A roadmap towards financial machine learning," *Available at SSRN 3365282*, 2019.

[50] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance," *arXiv preprint arXiv:1602.06561*, 2016.

[51] V. H. Shah, "Machine learning techniques for stock prediction," *Foundations of Machine Learning| Spring*, vol. 1, no. 1, pp. 6–12, 2007.

[52] T. Puschmann, “Fintech,” *Business & Information Systems Engineering*, vol. 59, no. 1, pp. 69–76, 2017, doi: 10.1007/s12599-017-0464-6.

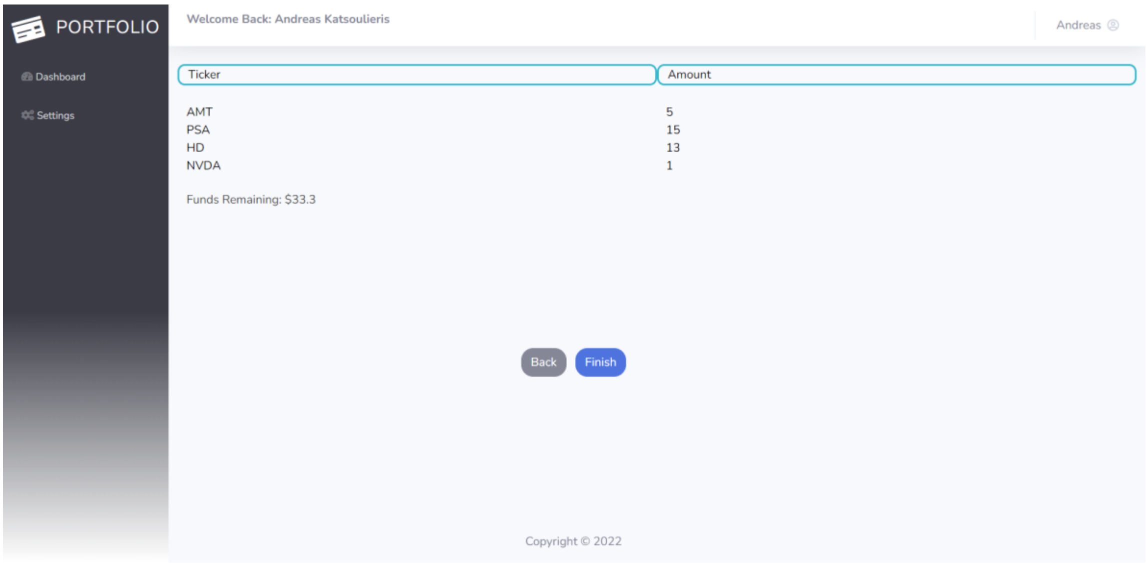
[53] S. Biswas, B. Carson, V. Chung, S. Singh, and R. Thomas, "AI-bank of the future: can banks meet the AI challenge? McKinsey." 2020.

[54] L. D. Wall, "Some financial regulatory implications of artificial intelligence," *Journal of Economics and Business*, vol. 100, pp. 55–63, 2018, doi: https://doi.org/10.1016/j.jeconbus.2018.05.003.

[55] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine learning in Finance*, vol. 1170. Springer, 2020.

[56] TensorFlow, "Time series forecasting | TensorFlow Core," Jan. 26, 2022. https://www.tensorflow.org/tutorials/structured_data/time_series (accessed Nov. 16, 2021).

9 Appendix A - UI Screenshots

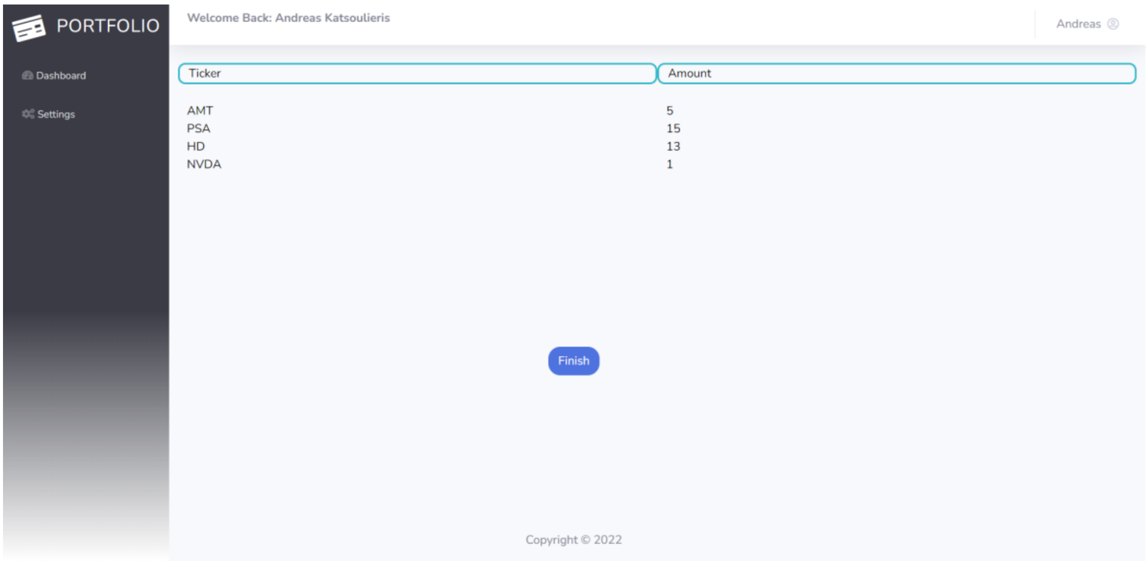


Welcome Back: Andreas Katsoulieris

Ticker	Amount
AMT	5
PSA	15
HD	13
NVDA	1

Funds Remaining: $33.3

Figure 24 Allocation Page



Welcome Back: Andreas Katsoulieris

Ticker	Amount
AMT	5
PSA	15
HD	13
NVDA	1

Figure 25 Rebalance Page

10 Appendix B - Initial Gannt Chart

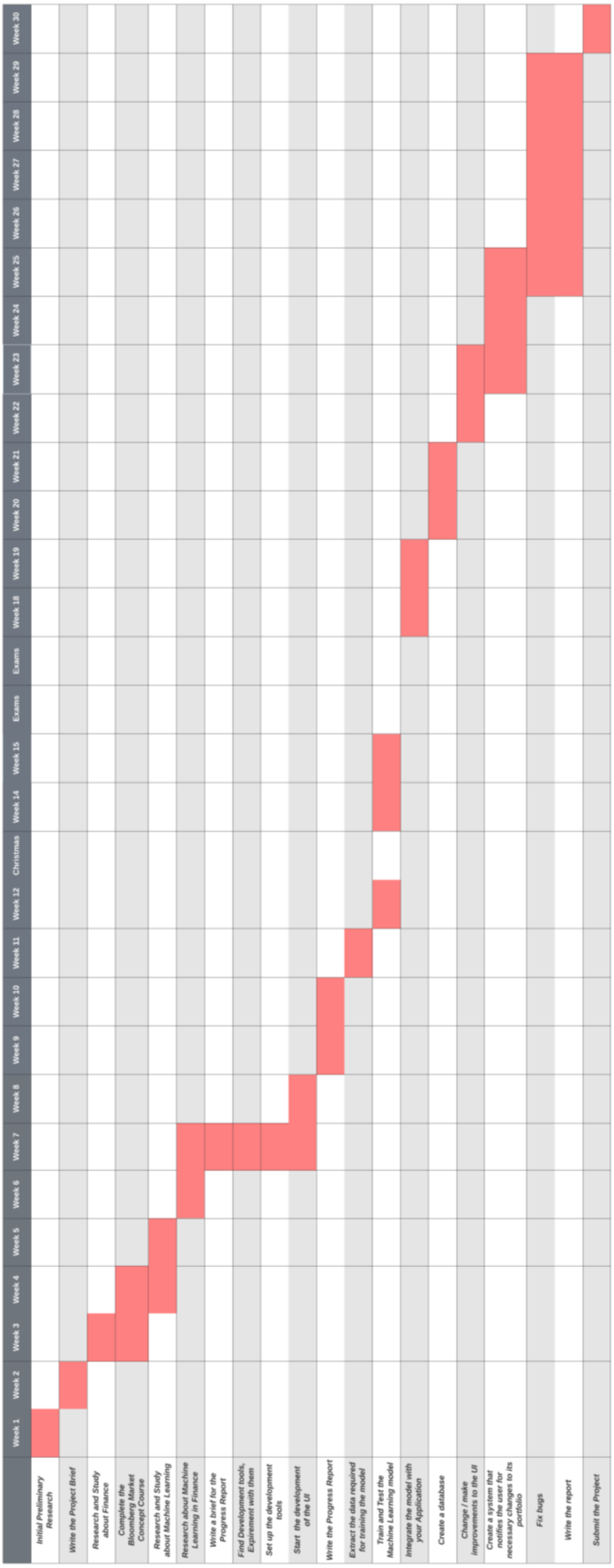


Figure 26 Initial Gannt Char