PLOS ONE

RESEARCH ARTICLE

Investigating the dynamics and uncertainties in portfolio optimization using the Fourier-Millen transform

Muhammad Hilal Alkhudaydi[1]*, **Aiedh Mrisi Alharthi**[2]

1 Department of Mathematics and Statistics, College of Science, Taif University, Taif City, Saudi Arabia, **2** Department of Mathematics, Turabah University College, Taif University, Taif City, Saudi Arabia

* mh.ayedh@tu.edu.sa; amharthi@tu.edu.sa

OPEN ACCESS

Citation: Alkhudaydi MH, Alharthi AM (2025) Investigating the dynamics and uncertainties in portfolio optimization using the Fourier-Millen transform. PLoS One 20(6): e0321204. https://doi.org/10.1371/journal.pone.0321204

Editor: Juan E. Trinidad-Segovia, University of Almeria: Universidad de Almeria, SPAIN

Received: June 27, 2024

Accepted: March 3, 2025

Published: June 17, 2025

Data availability statement: All relevant data are included within the paper and its Supporting Information files.

Funding: The authors would like to thank Deanship of Graduate Studies and Scientific Research at Taif University Saudi Arabia for funding this project no.202311. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Many investors and financial managers view portfolio optimisation as a critical step in the management and selection processes. This is due to the fact that a portfolio fundamentally comprises a collection of uncertain securities, such as equities. For this reason, having a solid understanding of the elements responsible for these uncertainties is absolutely necessary. Investors will always look for a portfolio that can handle the required amount of risk while still producing the desired level of expected returns. This article uses feature-based models to investigate the primary elements that contribute to the optimal composition of a specific portfolio. These models make use of physical analyses, such as the Fourier transform, wavelet transforms and the Fourier–Mellin transform. Motivated by their use in medical analysis and detection, the purpose of this research was to analyse the efficacy of these methods in establishing the primary factors that go into optimising a particular portfolio. These geometric features are input into artificial neural networks, including convolutional and recurrent networks. These are then compared with other algorithms, such as vector autoregression, in portfolio optimisation tests. By testing these models on real-world data obtained from the US stock market, we were able to obtain preliminary findings on their utility.

Introduction

The overarching goal of this research is to identify the most effective strategy for portfolio management. To achieve this, we employed various methods for extracting information to identify the most effective components for portfolio optimisation.

Portfolio optimisation is a key component of contemporary finance, as it helps investors maximise profits while lowering risk. Due to rapid improvements in technology and the increasing complexity of financial markets, there is a growing need for sophisticated strategies to efficiently optimise portfolios [1]. This study seeks to provide a thorough understanding of cutting-edge approaches and their applications to portfolio optimisation using MATLAB. An outline of the study is first presented. Then, a section on the theory explores the guiding concepts of each of the seven topics covered.

Competing interests: No authors have competing interests.

The study goes on to compare the outcomes of three portfolio optimisation strategies: vector autoregression (VAR) (as a baseline), continuous wavelet transform (CWT) combined with a convolutional neural network (CNN), and the Fourier–Mellin transform (FMT) combined with a recurrent neural network (RNN). The purpose of the paper to determine the best method for optimising portfolios under various market scenarios by contrasting the performance of several feature extraction methods based on deep learning.

The application of machine learning methods, including automated machine learning (AutoML), CNNs, and long short-term memory (LSTM) networks, has garnered considerable interest within the dynamic field of investment strategies. The use of these methodologies, which are frequently combined with traditional statistical approaches such as VAR, CWT, and the FMT, presents opportunities for improving investment decision-making and portfolio management.

Portfolio optimisation is the process of choosing the best feasible set of assets to accomplish a particular investment goal, while taking the investor's risk tolerance into account [2]. Modern portfolio theory (MPT), which established the concepts of diversification and the efficient frontier, serves as the theoretical underpinning of portfolio optimisation [3]. To reduce risk, diversification includes spreading investments across a variety of assets, while the efficient frontier is the set of ideal portfolios that provide the best expected return for a given level of risk. The Black–Litterman model, downside risk measures, and resilient optimisation techniques are only a few of the new optimisation techniques and risk measures that have been introduced as a result of the continued development and extension of MPT by numerous studies [4].

A study by Feng et al. conducted an in-depth examination of the contributions of different frequency components to image characteristics using weighted standard deviation (WSD) and weighted column standard deviation-based filtering to investigate image features [5]. The suggested methodology employs a weighted evaluation, which highlights specific aspects based on their importance and provides a numerical measure for quantitative analysis. Although WSD is computationally complex and requires careful interpretation, its ability to detect both high- and mid-frequency components allows for a comprehensive understanding of image quality [5]. The authors use wavelet spectral density to build a strong statistical model that accurately shows how different frequency components affect each other. This could make the model useful for processing and analysing images [5].

Although the work introduces advanced approaches for image registration and analysis using WSD methodologies, it also directly addresses the influence of noise as a key limitation, especially in short-exposure images. For rotational and scaling shifts, the authors agree that noise may reduce the accuracy of relative shift calculations. In this regard, the proposed FMT approach may not be as effective. Robust noise reduction methods must be used to address this issue and achieve improved outcomes from image processing.

The goal of portfolio optimisation is to provide investors with guidance for controlling their portfolios by determining the optimal allocations in different markets. Ali et al. analysed different portfolios by experimenting with changes in optimal weights for two eras – pre-COVID and during the COVID pandemic – in Asia Pacific [6]. They implemented various models as a core feature of their study, including dynamic conditional correlation (DCC), which is an extension of constant correlation estimation; multivariate and bivariate DCC-generalised autoregressive conditional heteroskedasticity (GARCH) models; the standard GARCH (1,1) model; and portfolio optimisation methods such as mean-variance (MV) and safe-haven dynamics, to provide protection during market conditions (a hedge) [6].

Additionally, the use of DCC-GARCH models was investigated to study green and non-green cryptocurrencies, diversification, risk management, and the impact of green assets

on various equity portfolios by analysing risk–return dynamics [7]. This study evaluated the effectiveness of several portfolio optimisation techniques and weight constraints on assets to gain a deeper understanding of their practical implementations. By testing the safe-haven dynamics of specific cryptocurrencies, the study offers valuable insights into cross-market hedging and safe-haven opportunities for investors during periods of extremely low returns [7]. The assumptions used in the study, including GARCH modelling, which fails to capture certain market dynamics (such as jumps), may impact its robustness. Methods used to determine the optimal weights may also be significantly affected by the addition of factors such as liquidity limitations, investor preferences, and transaction costs, among others. Therefore, combining these studies with deep learning methods is important to investigate the key factors behind portfolio optimisation dynamics. Deep learning approaches have the power to handle various complexities, such as non-linear patterns in data, which is missing in traditional statistical models such as GARCH. Another advantage to deep learning methods is that they learn from data automatically, in contrast to the GARCH model, which depends more on manual feature engineering [8].

Artificial neural networks with many layers are the focus of the machine learning field known as deep learning. These networks can recognise subtle correlations and patterns in data [4], making them appropriate for a range of financial applications including portfolio optimisation [9]. Large datasets can be used to train deep learning models, enabling them to recognise complex patterns in financial data that may be challenging to find using more conventional techniques [10]. This may result in more accurate portfolio allocation decisions and better asset return estimates. Feedforward neural networks, CNNs, RNNs, and autoencoders are a few of the common deep learning architectures used in finance.

Specialised neural network architectures such as CNNs and RNNs are designed for particular kinds of data. CNNs have been used to evaluate financial time series data for a variety of tasks, including predicting stock prices and spotting market trends [11]. CNNs are particularly well suited for processing grid-like data, such as pictures or time series data. The temporal dependencies in financial time series data have been modelled using RNNs, which are built to process sequential data. Due to their capacity to recognise intricate patterns and correlations in financial data, both CNNs and RNNs have demonstrated promise in financial applications including portfolio optimisation. Popular RNN variants have been used to solve the vanishing gradient problem (which can occur in conventional RNNs during trainings), including LSTM and gated recurrent unit (GRU) networks.

By projecting high-dimensional data onto a lower-dimensional space while retaining as much of the data's variance as possible, dimensionality reduction techniques seek to reduce the complexity of huge datasets. One common linear transformation method used to reduce dimensionality is principal component analysis (PCA) [12]. PCA can help uncover important characteristics influencing asset returns in the context of portfolio optimisation and help minimise the dimensionality of the optimisation issue [13]. This can in turn reduce the danger of overfitting and increase the computational efficiency of portfolio optimisation algorithms. Other dimensionality reduction methods have also been investigated for a variety of applications within finance, including t-distributed stochastic neighbour embedding and independent component analysis.

Multivariate extensions of autoregressive models known as VAR(p) models are capable of capturing the dynamic interactions between several financial variables. Important insights for portfolio management have been made by using these models to simulate the coupled dynamics of asset returns, interest rates, and other macroeconomic variables [14]. VAR(p) models can assist in increasing the accuracy of portfolio optimisation algorithms and improve

investment decisions by incorporating data on many financial factors [15]. To address key difficulties in financial time series analysis, such as cointegration and model uncertainty, extensions of VAR models have been developed, such as the vector error correction model and the Bayesian vector autoregressive model.

A time series can be divided into multiple frequency components using a wavelet transform, which enables researchers to examine the data at various time scales [10]. This method has been used to capture both short- and long-term patterns in asset returns in financial time series data [15]. Wavelet transforms can facilitate increased precision in portfolio optimisation algorithms and result in better investment decisions by combining data from various time scales [16]. Numerous financial applications, including volatility forecasting, risk management, and market efficiency analysis, have used wavelet-based techniques.

The FMT combines the Fourier transform and polar coordinates to examine a signal's frequency and scale content [17]. By using this combination, the FMT can examine the frequency and scale content of financial data, resulting in a more thorough understanding of the data [18]. The FMT has been used by researchers in a number of image processing applications, including classification problems, to showcase its ability to process and analyse data using deep learning methods [5].The FMT can assist in increasing the accuracy of portfolio optimisation algorithms, resulting in better investment decisions by using data from both the frequency and scale domains.

This study introduces feature-based models that contribute to the field of portfolio optimisation in the following ways:

1. Feature extraction is considered as a vital component for data analyses, with financial data being no exception. Investigating the underlying uncertainty of portfolio dynamics is a difficult task when relying only on historical prices or returns of assets, necessitating the use of a model able to learn from the data, such as a deep learning model.
2. The FMT has many applications in image processing, including registration and rotation. This works by extracting the important geometric features using the Fourier transform along with polar coordinates. In addition, the wavelet transform is a beneficial tool in financial time series analysis, because it can handle non-stationary financial data, whereas the Fourier transform only deals with stationary cases.
3. Integrating deep learning with the wavelet transform and FMT can lead to state-of-the-art feature-based deep neural network models capable of capturing the important elements of a financial portfolio. Our goal is not only to assess our models' predictability but also to evaluate their ability to investigate the core data behind the portfolio dynamics.

This paper is divided into four sections. Section Mathematical explanation of tools used in this paper explains the mathematical methods necessary to build feature-based model blocks. These include VAR, PCA, CWT, RNNs, CNNs and the FMT. Section System design focuses on the design of three main feature-based models, from data selection to their structure. These models include the VAR automated machine learning investment strategy model (VAR(1)-AutoML), and the CWT investment strategy model (CWT-CNN), and the FMT recurrent neural network (FM-LSTM). Section Results evaluates the models' performance, beginning with statistical analysis for the probability measures. Section Conclusions and future work discusses the conclusions of the study by highlighting its strengths and limitations.

Mathematical explanation of tools used in this paper

The goal of this section is to describe the key mathematical concepts employed in this study.

Vector autoregression

VAR is a statistical model widely used in econometrics and applied statistics. It is employed to analyse the linear relationships between several time series variables [19]. The VAR technique enables the modelling of each variable in a multivariate time series by considering its past values as well as the past values of other variables [19,20].

Consider a multivariate time series $\{\mathbf{y}_t\}$, where each $\mathbf{y}_t$ is a $p \times 1$ vector of p endogenous variables at time t:

$$\mathbf{y}_t = \begin{bmatrix} y_{1t} \\ y_{2t} \\ \vdots \\ y_{pt} \end{bmatrix}. \tag{1}$$

A VAR model of order p (VAR(p)) can be represented as:

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \ldots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{u}_t, \tag{2}$$

where:

- $\mathbf{c}$ is a $p \times 1$ vector of constants (intercepts),
- $\mathbf{A}_i$ (for $i = 1, 2, \ldots, p$) are $p \times p$ coefficient matrices, and
- $\mathbf{u}_t$ is a $p \times 1$ vector of error terms.

The error terms $\mathbf{u}_t$ are considered to have the following additional characteristics:

- The expected value of $\mathbf{u}_t$ is $\mathbf{0}$ ($E(\mathbf{u}_t) = \mathbf{0}$), indicating it has a mean of zero.
- The expected value of $\mathbf{u}_t\mathbf{u}_t'$ is Σ ($E(\mathbf{u}_t\mathbf{u}_t') = \Sigma$), where Σ is a positive definite matrix of size $p \times p$ that represents the covariance matrix of the error terms.
- The expected value of $\mathbf{u}_t\mathbf{u}_s'$ is $\mathbf{0}$ ($E(\mathbf{u}_t\mathbf{u}_s') = \mathbf{0}$) for $t \neq s$, ensuring that the errors are not correlated with each other over time [19–21].

VAR models are used for predicting the behaviour of a set of variables,
performing impulse response analysis to investigate the impact of shocks,
and conducting variance decomposition to analyse the individual impact of each variable on the predicted errors [22].

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{u}_t, \tag{3}$$

is the formula for the VAR(1) model.

The given equation states that $\mathbf{c}$ is a vector of constants, $\mathbf{A}_1$ is a matrix of coefficients, and $\mathbf{u}_t$ is a vector of error terms [19–21].

Based on the assumptions made by the model, the error terms $\mathbf{u}_t$ are assumed to meet certain criteria:

- **Zero mean:** $E[\mathbf{u}_t] = \mathbf{0}$.
- **Constant covariance:** $E[\mathbf{u}_t\mathbf{u}_t'] = \Sigma$.
- **No serial correlation:** $E[\mathbf{u}_t\mathbf{u}_s'] = \mathbf{0}$ for all $t \neq s$.

Ordinary least squares (OLS) regression can be used to approximated the parameters **c** and $\mathbf{A}_1$ [23]. To obtain the estimates, we run a regression on all the variables using their lagged values.

For the VAR(1) model to be stable, each eigenvalue of $\mathbf{A}_1$ must have a modulus smaller than one. The way the system reacts to shocks is set by the characteristic equation obtained from $\mathbf{A}_1$.

One way to observe the long-term effects of a single variable shock in a VAR(1) model is through the impulse response functions. The answers are obtained by raising $\mathbf{A}_1$ to an appropriate power.

Econometricians and finance econometricians frequently employ VAR models when studying the fluctuating connections between different time series. Financial market analysts also often use VAR models to track asset returns so as to analyse interactions and predict future moves [19–21].

VAR models are widely employed in the fields of econometrics and finance econometrics to examine the dynamic connections among numerous time series. They are commonly used in financial markets to analyse the log returns of assets, with the aim of understanding their relationships and making predictions about future movements.

Logarithmic returns are commonly employed in finance because of their advantageous characteristics, including time additivity and improved management of the multiplicative nature of asset values. The log return of asset i from time $t-1$ to t for a given collection of assets is calculated using the formula:

$$r_{i,t} = \log\left(\frac{P_{i,t}}{P_{i,t-1}}\right). \tag{4}$$

The symbol $P_{i,t}$ denotes the price of asset i at time t.

Given a set of n assets, the VAR model represents the log returns of these assets as:

$$\mathbf{r}_t = \mathbf{c} + \mathbf{A}_1\mathbf{r}_{t-1} + \mathbf{A}_2\mathbf{r}_{t-2} + \ldots + \mathbf{A}_p\mathbf{r}_{t-p} + \mathbf{u}_t. \tag{5}$$

The variable $\mathbf{r}_t$ is the vector of log returns at time t,

c is the vector of intercepts,

$\mathbf{A}_1, \ldots, \mathbf{A}_p$ are the coefficient matrices, and $\mathbf{u}_t$ is the vector of error terms, assumed to be white noise with covariance matrix Σ.

The OLS method is commonly used to estimate the parameters of the VAR model. The model also enables the examination of impulse response functions. Forecast error variance decomposition is a method used to analyse the impact of shocks to individual assets and how they contribute to the forecast error variances of all assets [19–21].

Principal component analysis

PCA is a statistical method commonly used for dimensionality reduction in a variety of domains, including financial mathematics. In the context of portfolio optimisation, PCA can be of great assistance in the process of extracting dominating financial ratios from a wide variety of economic data. These ratios can then be used to design an optimal investment portfolio [24]. The power of PCA in the field of financial mathematics stems from its capacity to transform complex multidimensional data into information easier to manage without compromising essential information. This, in turn, enables better informed investment decisions. It is interesting to note that although PCA has traditionally been linked to the maximisation

of variance, greater variance does not always signal more important information for portfolio optimisation [24].

PCA is a prominent statistical methodology that employs an orthogonal transformation to transform a set of correlated observations of multiple variables into a set of principal components, which are linearly uncorrelated values. This procedure is frequently used to decrease the number of dimensions in datasets, improving comprehensibility while minimising the loss of information [25].

Prior to PCA analysis, a dataset is commonly denoted as an $n \times p$ matrix referred to as $\mathbf{X}$. Each row in this matrix represents a unique replication of the experiment, while each column represents a specific type of feature, such as data collected from a particular sensor [26]. To achieve precise outcomes, the dataset undergoes preprocessing to ensure that the empirical mean of each column is zero, suggesting that the sample mean of each characteristic has been corrected to zero.

The transformation in PCA is determined by a collection of l p-dimensional weight vectors or coefficients, denoted as $\mathbf{w}_{(k)} = (w_1, \ldots, w_p)_{(k)}$. The weight vectors are used to transform each row vector $\mathbf{x}_{(i)}$ of $\mathbf{X}$ into a new vector $\mathbf{t}_{(i)} = (t_1, \ldots, t_l)_{(i)}$ of principal component scores, which are calculated as follows:

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)} \quad \text{for } i = 1, \ldots, n \quad \text{and } k = 1, \ldots, l. \tag{6}$$

The scores $t_1, \ldots, t_l$ are calculated in a way that maximises the variance they derive from $\mathbf{X}$, while ensuring that each weight vector $\mathbf{w}$ is a unit vector.

To maximise variance, the initial weight vector $\mathbf{w}_{(1)}$ is obtained by solving the following optimisation problem:

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_{1(i)})^2 \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{x}_{(i)} \cdot \mathbf{w})^2 \right\}. \tag{7}$$

Alternatively, this can be represented in matrix format as:

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \|\mathbf{X}\mathbf{w}\|^2 \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{X} \mathbf{w} \right\}. \tag{8}$$

The first principal component score for a data vector $\mathbf{x}_{(i)}$ is determined by the dot product of $\mathbf{x}_{(i)}$ and $\mathbf{w}_{(1)}$, resulting in $t_1(i) = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(1)}$.

The k-th principal component can be obtained by first removing the influence from the previous $k - 1$ components. This modified data matrix, denoted as $\hat{\mathbf{X}}_k$, is written as:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathsf{T}}. \tag{9}$$

Here, $\mathbf{w}_{(s)}$ represents the weight vectors based on the first $k - 1$ principal components.

The weight vector for the k-th principal component, $\mathbf{w}_{(k)}$, is then determined by solving the following optimisation problem:

$$\mathbf{w}_{(k)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg\max \left\{ \frac{\mathbf{w}^{\mathsf{T}} \hat{\mathbf{X}}_k^{\mathsf{T}} \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^{\mathsf{T}} \mathbf{w}} \right\}. \tag{10}$$

Solving this maximisation optimisation problem yields an eigenvector that has a strong relationship with the largest eigenvalue of $\hat{\mathbf{X}}_k^{\top}\hat{\mathbf{X}}_k$, which helps to determine the direction of maximum variance.

The k-th principal component score for a data vector $\mathbf{x}(i)$ can be evaluated as:

$$t_k(i) = \mathbf{x}(i) \cdot \mathbf{w}(k). \tag{11}$$

Here, $\mathbf{w}(k)$ represents the k-th eigenvector of $\mathbf{X}^{\top}\mathbf{X}$.

The complete principal component decomposition of $\mathbf{X}$ can be written as:

$$\mathbf{T} = \mathbf{XW}. \tag{12}$$

In this context, the matrix of weights, denoted as $\mathbf{W}$, consists of columns that are the eigenvectors of $\mathbf{X}^{\top}\mathbf{X}$. These columns, which are scaled by the square root of the corresponding eigenvalues, are referred to as loadings.

The covariance matrix, denoted by $\mathbf{Q}$, is proportional to the transpose of the matrix X multiplied by $\mathbf{X}^{\top}\mathbf{X}$. Specifically, $\mathbf{Q}$ is related to the PCA of the dataset, and the covariance between different principal components can be expressed as follows [26]:

$$Q(\mathrm{PC}_{(j)}, \mathrm{PC}_{(k)}) = \lambda_{(k)}\mathbf{w}_{(j)}^{\top}\mathbf{w}_{(k)}. \tag{13}$$

The eigenvalues are represented by the symbol $\lambda_{(k)}$. The eigenvectors corresponding to various eigenvalues are orthogonal, which ensures there is no covariance between different principal components.

The matrix transformation that diagonalises the empirical sample covariance matrix is often referred to as the whitening or sphering transformation. In formal terms, it is represented as follows:

$$\mathbf{W}^{\top}\mathbf{QW} \propto \Lambda. \tag{14}$$

Here, Λ represents the diagonal matrix of eigenvalues.

PCA is an effective method for transforming original variables into a new coordinate system. In this new system, the axes are the directions of maximum variance, which are orthogonal to each other and ranked according to the variance they explain. This transformation is essential for applications in data reduction, noise reduction, and exploratory data analysis.

PCA transforms a data vector $\mathbf{x}(i)$ from the original space defined by p variables into a new space of p uncorrelated variables. This transformation is defined by the matrix equation:

$$\mathbf{T} = \mathbf{XW}. \tag{15}$$

This equation represents the relationship between the original data matrix (denoted as $\mathbf{X}$) and the matrix of eigenvectors (denoted as $\mathbf{W}$) ,where the columns of $\mathbf{W}$ are the eigenvectors of the covariance matrix of $\mathbf{X}$.

However, it is not mandatory to preserve all principal components. Preserving only the first L principal components, which correspond to the first L eigenvectors, leads to a truncated transformation:

$$\mathbf{T}_L = \mathbf{XW}_L, \tag{16}$$

where $\mathbf{T}_L$ contains n rows but only L columns, effectively reducing the dimensionality of the data. The transformation yielded by PCA takes the following form:

$$\mathbf{t} = \mathbf{W}_L^{\top}\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^p, \quad \mathbf{t} \in \mathbb{R}^L. \tag{17}$$

Here, $\mathbf{W}_L$ represents a $p \times L$ matrix that forms an orthogonal basis for the L features [27].

The purpose of this transformation is to optimise the score matrix $\mathbf{T}_L$ so that it maximises the amount of variance from the original data that is preserved and minimises the total squared reconstruction error. This can be quantified by the following equation:

$$\|\mathbf{T}\mathbf{W}^{\top} - \mathbf{T}_L\mathbf{W}_L^{\top}\|_2^2 \quad \text{or} \quad \|\mathbf{X} - \mathbf{X}_L\|_2^2. \tag{18}$$

The difference between the reconstructed data from the full transformation and that from the truncated transformation is represented by this error measure. This difference highlights the effectiveness of PCA in identifying the most important aspects of the data.

By using PCA, key features of the dataset can be effectively preserved in a smaller number of dimensions. This makes analysis and visualisation less complex, while preserving essential information on the variability of the data.

Continuous wavelet transform

The CWT can be used to analyse non-stationary signals. This is accomplished by splitting the signals into wavelets, which are functions that are both time- and frequency-localised. Unlike Fourier transforms, the CWT provides a representation of time and frequency, making it particularly well suited for signals whose frequency content is subject to changes over time.

The CWT of a function $f(t)$ is represented as [28]:

$$W_f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right) dt. \tag{19}$$

The function $\psi(t)$ is the mother wavelet, where a and b represent the scale and translation parameters, respectively, and ψ^* signifies the complex conjugate of ψ [28,29].

To meet the admissibility requirement—having a zero mean and a finite energy—it is crucial for the mother wavelet $\psi(t)$ to act as a band-pass filter. Furthermore, it needs to be zero-integrated [28,29].

The signal processing community uses CWT for a variety of tasks, including feature extraction, noise reduction, and time-frequency analysis.

Recurrent neural networks

An RNN is an artificial neural network in which the connections between nodes follow a specific temporal sequence. This allows them to display temporally dynamic behaviour for the given time sequence. RNNs differ from feedforward neural networks in that they may process input sequences using their internal state (memory). This opens up a world of possibilities for application, such as to speech recognition and unsegmented linked handwriting recognition.

The fundamental building blocks of RNNs are nodes, also known as neurons. At each time step, nodes modify their activation and transmit this activation to the subsequent time step [30,31].

According to [32], this can be expressed mathematically as follows. The input at each step t is represented by each element of the input vector $\{x_1, x_2, \ldots, x_T\}$. The RNN uses these equations to find the hidden vector sequence $\{h_1, h_2, \ldots, h_T\}$ and, if necessary, the output vector sequence $\{y_1, y_2, \ldots, y_T\}$ from $t = 1$ to T:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \text{ and} \tag{20}$$

$$y_t = W_{hy}h_t + b_y. \tag{21}$$

Here, the hidden state at time t is represented by h_t. The activation function is usually a non-linear function like tanh or ReLU, and the weight matrices (parameters) for input-to-hidden, hidden-to-hidden, and hidden-to-output connections are W_{hh}, W_{xh}, and W_{hy}, respectively.

The bias vectors are denoted as b_h and b_y, whereas the output at time t is represented by y_t.

In the case of output gradients that are very small relative to the parameters, the vanishing gradient problem arises, making the network incapable of identifying long-range correlations in the input data. This is one of the main obstacles to training RNNs. According to [30], two common RNN variations that aim to address this problem are LSTM units and GRUs.

LSTM networks also employ gating mechanisms to control the information flow, and they keep a separate cell state in addition to the concealed state. A set of gate types are present in every unit. A forget gate determines which pieces of data to remove from the cell's state, while an input gate determines which cell state values should be updated.

It is the job of the output gate to decide at each step which component of the cell state should be output see (Fig 1).

The mathematical expressions for these gates are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{22}$$

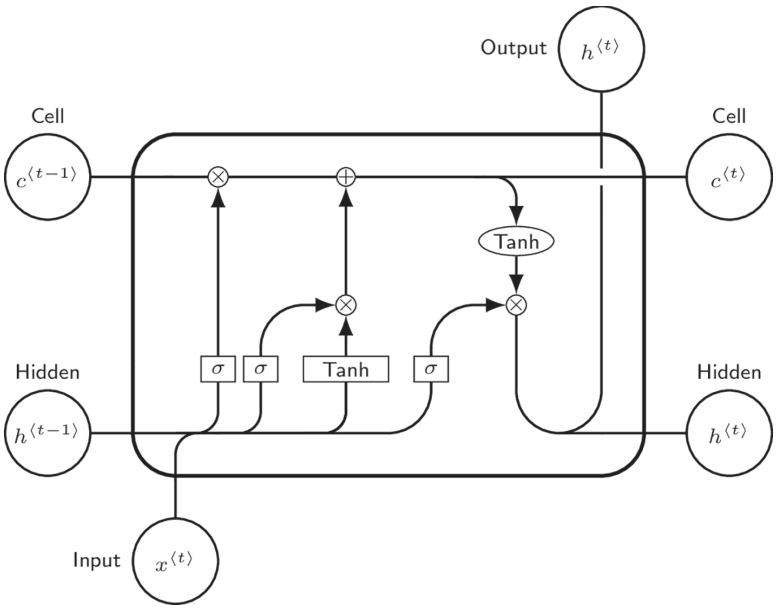

Fig 1. An LSTM model diagram by J. Leon, Beerware. The graph shows the the architecture of LSTM model.

https://doi.org/10.1371/journal.pone.0321204.g001

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{23}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{24}$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{25}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \text{ and} \tag{26}$$
$$h_t = o_t \cdot \tanh(c_t). \tag{27}$$

Convolutional neural networks

Combining concepts from deep learning with signal processing allows for a description of the architecture and functioning of CNNs using wavelet transforms. This is especially important when dealing with financial time series data, such as r log returns. When analysing non-stationary financial time series data, in which features such as volatility might fluctuate over time, wavelet transforms offer a highly beneficial time-frequency analysis tool see (Fig 2).

This article uses CNNs in conjunction with the CWT to analyse financial time series data, with a particular emphasis on $r(t)$ log returns. Improved CNN feature extraction for dynamic financial data is achieved using the fine-grained time-frequency localisation features of CWTs.

Financial instruments' log returns are notoriously difficult to analyse due to their non-linear behaviour and volatility. The CWT is one effective method for breaking down these non-stationary signals into time-frequency space and improving the capacity to detect localised scale-dependent patterns.

A time series $r(t)$ can be transformed into a two-dimensional signal representation in the time and frequency domains using the CWT [33,34]. The definition of the transform is:

$$R(a,b) = \int_{-\infty}^{\infty} r(t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt, \tag{28}$$

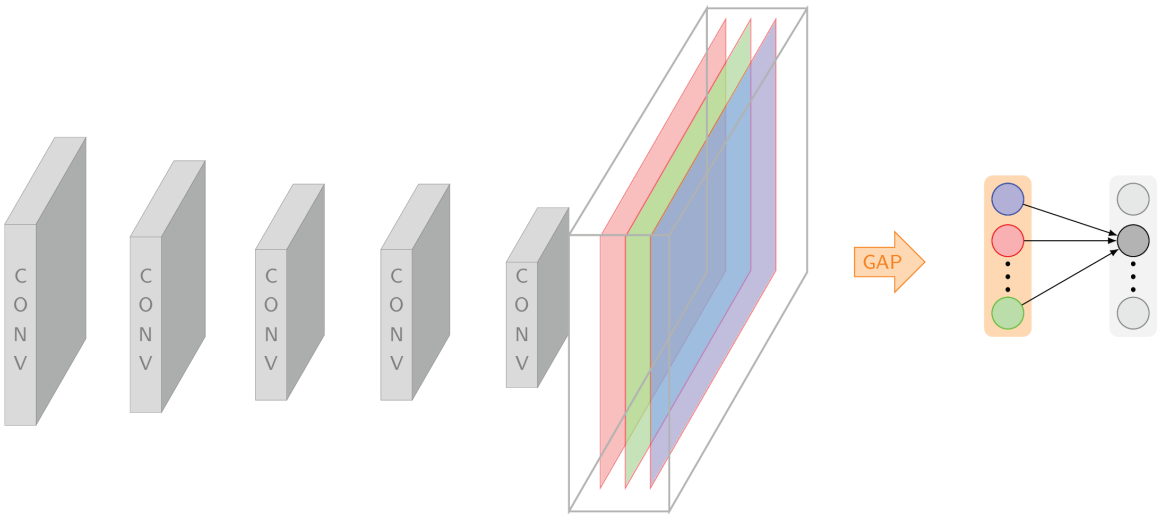

Fig 2. A CNN architecture illustration [36]. The graph shows the the architecture of CNN model.

https://doi.org/10.1371/journal.pone.0321204.g002

where a and b are the scale and translation parameters, respectively, and $\psi(t)$ is the mother wavelet function.

The CNN receives its input from the two-dimensional CWT coefficient matrix, which encodes signal information across both time and frequency.

To extract features from the CWT coefficients, these layers employ various filters via the equation:

$$a_i^{[l+1]} = \sigma\left(b_i^{[l]} + \sum_j f_{ij}^{[l]} \cdot a_j^{[l]}\right). \tag{29}$$

Each filter picks up on unique patterns, such as spikes in frequency around certain dates, which may be indicative of major financial occurrences [35]. $\mathcal{R}_i$ represents the region (or set of indices) over which the pooling is performed for the i-th unit of layer $l + 1$.

The deep neural network is completed by fully linked layers that combine the acquired characteristics to make predictions or classify patterns.

During training, the goal is to improve a loss function suitable for the specific job at hand, such as mean squared error for regression or cross-entropy for classification. This is achieved by employing methods such as stochastic gradient descent.

Combining CWTs with CNNs provides an effective method for analysing complex and dynamic financial time series. This strategy improves the accuracy of predictions and allows for improved extraction of data by collecting specific local characteristics at different levels of detail [34].

Fourier–Mellin transform

This section provides an examination of the image-matching methodology (in this context, by image, we mean the log-returns time series lag of the asset's wavelet transform), showing its limitations and offering the symmetry phase-only FMT (SPOFM) as a potential alternative approach. Two phases are involved in SPOFM: calculating the Fourier–Mellin invariant (FMI) descriptors and matching FMI descriptors to two-dimensional images. Some of the notable advantages of SPOFM are its robustness against noise, accuracy in numerical computations, and effectiveness in data selection. Nevertheless, there are several limitations associated with its use in scenarios involving image rotation and scaling. A proposed alternate approach involves employing a circular harmonic expansion technique, which introduces the additional task of identifying the common centre of rotation. The combination of the FMI and SMPOF methodologies capitalises on their respective strengths and is experiencing growing popularity. This section also examines the role of the FMT in the comparison of image sizes, emphasising its wide peak as a notable limitation in determining the location and identification of object elements.

The primary aim of the image comparison is the recognition of the existence of an image within a background of noise in the form:

$$s(x, y) = r(x', y') + n(x, y). \tag{30}$$

The term on the left-hand side is the noisy scene function $s(x, y)$. The first function on the right-hand side, $r(x', y')$, is the image function, while the remaining function $n(x, y)$ is the zero mean, which remains unaffected by the image signal $r(x, y)$. The following equation can

be used to determine the geometric translation measures between the two images:

$$s(x,y) = r(x'(x,y,p_1,\dots,p_n), y'(x,y,p_1,\dots,p_n)) + n(x,y). \tag{31}$$

Here, the x' and y' components, as well as the geometric components $p_1, p_2, \dots, p_n$, represent the computational functions associated with the geometric translation regarding the variable p. For illustrative purposes, consider a two-dimensional scenario where the translation offset is denoted as (x_0, y_0).

$$R(u,v) = \mathcal{F}(r(x,y)), S(u,v) = \mathcal{F}(s(x,y)). \tag{32}$$

Here, $\mathcal{F}$ is the Fourier transform function, and:

$$x' = x - x_0,\ y' = y - y_0. \tag{33}$$

The transfer function $H(u, v)$ can be considered as the maximum detection of the signal, which at the same time minimises the noise. Mathematically, the equation can be expressed as:

$$H(u,v) = \frac{R^*(u,v)}{|N(u,v)|^2}. \tag{34}$$

The quantity $R^*(u, v)$ can be defined as the complex conjugate of the Fourier spectrum $R(u, v)$. Additionally, it can be expressed as the product of the noise power-spectral intensity $|N(u,v)|^2$.

If the noise spectrum exhibits a uniform distribution, with intensity denoted as n_w, then the transfer function decreases to a level determined by:

$$H(u,v) = \frac{1}{|n_w|^2} R^*(u,v). \tag{35}$$

The result of the filter function is given by the convolution of the functions $s(x, y)$ and $r^*(-x,-y)$, which can be expressed as:

$$q_0(u,v) = \frac{1}{|n_w|^2} \int \int_{-\infty}^{\infty} s(a,b) r^*(a-x, b-y) dadb. \tag{36}$$

The function achieves its highest value at the point (x_0, y_0), according to [37]. This point determines the values of the translation parameters as well as the intensity of the noise, denoted as n_w and calculated as the squared absolute value of $N(u, v)$.

This becomes an obstacle in image matching approaches when there is limited shape differentiation but where the images have the same size and energy content [37].

This also presents a difficulty in identifying the maximum value in the presence of noise, a problem that can be resolved by the use of a phase-only matching filter.

The transfer function is defined as follows:

$$H(u,v) = \text{Phase}(R^*(u,v)) = \exp[j(-\phi_r(u,v))]. \tag{37}$$

In this context, it becomes clear that $j^2 = -1$ due to the fact that the spectral phase preserves only the spatial information of the object, while rejecting the image's energy content.

The use of the phase-only matched filter yields a more distinguishable picture in comparison to the conventional matched filtering technique, hence increasing the ability to discriminate between objects.

The approach can be enhanced by acquiring the correlation phase of both the image and the noise function [37]. This is achieved by including a non-linear filter that produces output according to the following equation:

$$Q(u,v) = \frac{S(u,v)}{|S(u,v)|} \bullet \frac{R^*(u,v)}{|R^*(u,v)|} = \exp[j(\phi_s(u,v) - \phi_r(u,v))]. \tag{38}$$

The spectral phases $\phi_r(u,v)$ and $\phi_s(u,v)$ indicate the image signal $r(x, y)$ and the noise function $s(x, y)$, respectively. When the level of noise is insignificant, the aforementioned equation can be rewritten as:

$$Q(u,v) = \exp[-j2\pi(ux_0 + vy_0)]. \tag{39}$$

The inverse Fourier transform of the above function yields a Dirac delta function centred at the coordinates (x_0, y_0). This function exhibits superior performance in comparison to phase-only matched filtering (POMF).

The technique discussed here offers as an alternative approach to POMF known as symmetry POMF (SPOMF) [37].

The image transformation method consists of three sequential processes: rotation, scaling, and translation. These steps are applied to an image denoted as $r(x, y)$. The technique involves first collecting the phases of the image, followed by performing POMF [37].

This approach can be achieved by the implementation of pre-processing techniques in the spectral phase.

In the given scenario, an object indicated as $s(x, y)$ performs a series of operations resulting in the generation of a comparable image denoted as $r(x, y)$:

$$\begin{aligned} s(x,y) = r[&\sigma(x\cos\alpha + y\sin\alpha) - x_0 \text{ and} \\ &\sigma(-x\sin\alpha + y\cos\alpha) - y_0]. \end{aligned} \tag{40}$$

In the provided context, α represents the rotation angle, and σ denotes the uniform scale factor.

The variables x_0 and y_0 represent the offsets of the transformation. The mathematical representation of the Fourier transform is given by:

$$\begin{aligned} S(u,v) = \exp^{-j\phi_s}\sigma^{-2}|R[&\sigma^{-1}(u\cos\alpha + v\sin\alpha), \\ &\sigma^{-1}(-u\sin\alpha + v\cos\alpha)]|. \end{aligned} \tag{41}$$

The function $\phi_s(u,v)$ characterises the spectral phase of the input image, which is influenced by rotation, scaling, and translation. On the other hand, the spectral magnitude remains unaffected by changes in spatial size:

$$\begin{aligned} |S(u,v)| = \sigma^{-2}|R[&\sigma^{-1}(u\cos\alpha + v\sin\alpha), \\ &\sigma^{-1}(-u\sin\alpha + v\cos\alpha)]|. \end{aligned} \tag{42}$$

The scaling is constant under spectral operations for $u = v = 0$, as it is proportional to σ^{-1}. The two primary processes can be identified by employing polar coordinates to specify the magnitudes of s and r.

$$\begin{aligned} r_p(\theta, \rho) &= |R(\rho\cos\theta, \rho\sin\theta)| \text{ and} \\ s_p(\theta, \rho) &= |S(\rho\cos\theta, \rho\sin\theta)|. \end{aligned} \tag{43}$$

It is well established that the spectral magnitude exhibits periodic behaviour with respect to the angle θ. Thus, it is possible to estimate the transfer function of a filter by using only half of the magnitude field, as long as the original image is authentic. Therefore,

$$s_p(\theta, \rho) = \sigma^{-2} r_p(\theta - \alpha, \frac{\rho}{\sigma}). \tag{44}$$

The function $s_p(\theta, p)$ performs angular rotation, where r_p represents the spectral magnitudes. Conversely, the process of scaling involves the reduction of coordinates while simultaneously amplifying the constant factor denoted as σ^2.

The reduction of scaling can be achieved by implementing logarithms of scale in the radial coordinates, as proposed by [37]. These logarithmic transformations are defined as:

$$r_{\mathrm{pl}}(\theta, \lambda) = r_p(\theta, \rho). \tag{45}$$

Here pl stands for polar-logarithmic representation, and

$$s_{\mathrm{pl}}(\theta, \lambda) = s_p(\theta, \rho) = \sigma^{-2} r_{\mathrm{pl}}(\theta - \alpha, \lambda - k). \tag{46}$$

Here, $\lambda = \log(p)$, and $\kappa = \log(\sigma)$.

In the given equation, which is presented in polar logarithmic form, the operations of scaling and rotation have been simplified to translation. The resulting equation can be expressed as follows:

$$S_{\mathrm{pl}}(v, \varpi) = \sigma^{-2} \exp^{-j2\pi(v\kappa + \varpi\alpha)} R_{\mathrm{pl}}(v, \varpi). \tag{47}$$

Additionally, the computations for rotation and scaling are performed independently. Therefore, this numerical approach can be considered dependable. The natural visual system has significant parallels to the log-polar mapping technique.

The many strategies that have been mentioned can be integrated together to obtain an optimal outcome [37]. The comparison of the FMI descriptor of an image may be made with a variety of methods, including cross-correlation (CC), MF, POMF, and SPOMF. SPOMF exhibits a pronounced correlation peak, as demonstrated in the works of [37].

Let $s(x, y)$ and $r(x, y)$ denote the images obtained from the application of the SPOMF algorithm. Then:

$$Q_0(v, \varpi) = \frac{R^*_{\mathrm{pl}}(v, \varpi) \bullet S_{\mathrm{pl}}(v, \varpi)}{|R^*_{\mathrm{pl}}(v, \varpi)| \bullet |S_{\mathrm{pl}}(v, \varpi)|}. \tag{48}$$

The fundamental procedure for the SPOMF of the FMI involves the following steps:

Algorithm 1 The core mechanism of the FMI-SPOMF algorithm.

The procedure begins by applying the Fourier transform to the FMI descriptor of the reference image, which is represented as $R_{\mathrm{p1}}(\nu, \varpi)$.

After that, it proceeds to extract the phase $\exp[-j\phi_r(\nu, \varphi)]$ from $R_{\mathrm{p1}}(\nu, \varphi)$.

Then, the Fourier transform $S_{\mathrm{p1}}(\nu, w)$ of the FMI descriptor of the observed image s(x, y) is computed.

This is followed by the extraction of the phase:

$$\exp[-j\phi_s(\nu, \varpi)] \text{ of } S_{\mathrm{p1}}(\nu, \varpi).$$

The output of the SPOMF can be determined using the following equation:

$$Q_0(\nu, \varpi) = \exp[-j(\phi(\nu, \varpi) - \phi_r(\nu, \varpi))].$$

The process for determining the inverse of the Fourier transform is obtained via

$$q_0(\theta, \lambda) = \mathcal{F}^{-1}(Q_0(\nu, \varpi)).$$

The final procedure is to locate the point where the function $q_0(\theta, \lambda)$ is at its maximum.

In the initial phase of the procedure, it is assumed that all the images have identical characteristics, with the fundamental aim being to identify the geometric transformation that establishes correlation between these images.

The execution of the core FMI-SPOMF algorithm is performed as follows:

1. The output of the maximum filter is denoted as $q_0(\theta, \lambda)$ and is subsequently identified at the precise coordinates $\theta = \theta_{max}$ and $\lambda = \lambda_{max}$.
2. The value of the rotational angle is denoted as α and is equal to the maximum angle of rotation, represented by θ_{max}.
3. The scaling factor, denoted as σ, is mathematically defined as the exponential function of the maximum eigenvalue λ_{max}.

The descriptor is generated using half of the spectrum, resulting in an estimated rotation angle ranging from 0 to 180 degrees. The actual angle of rotation depends on a pair of potential values: α or $180^\circ + \alpha$. The second stage involves the use of the image registration method to determine the translation offset, followed by the determination of the rotation angle. A scaling process is performed on the observed image, resulting in two re-scaled images. These re-scaled images are then rotated by an angle of $\alpha + 180^\circ$. When executed accurately, the spectrum of the rotated image exhibits a phase discrepancy with $R(u, v)$, which results from the translation of the image. When performed incorrectly, there is a lack of relationship between the spectrum and the coordinates, denoted as $R(x, y)$.This implies that the appropriate angle of rotation can be calculated. This approach can be effectively employed in the identification of pattern recognition problems.

We can now go on to clarify the structure of image registration. Algorithm 2 below outlines the stages involved in image registration using the FMI-SPOMF.

Algorithm 2 Image registration mechanism.

Initiate the core FMI-SPOMF algorithm.
Identify the location coordinates (α, κ) that correspond to the maximum of $q_0(\theta, \lambda)$.
Resize the image s(x, y) by a factor of σ^{-1} and repeat this process.
Generate two resized versions of the image and rotate them by α and $180^{\circ} + \alpha$, respectively.
Assess the SPOMF between the reference image r(x, y) and the resized, rotated versions of the observed image.
Among the two manipulated images, determine the one with the highest filter maximum.
Identify the coordinates of this maximum.
Use the geometric features obtained from this transformation process.

Having discussed the importance of every element within the image registration framework, we can go on to outline an application that demonstrates how to take advantage of these geometric characteristics to optimise a portfolio.

To obtain these geometric characteristics, we employ the CWT on R_{pca} in a process similar to that described in Algorithm 4. However, it is necessary to provide a reference dataset to implement Algorithm 2.

System design

This section provides a comprehensive explanation of the system's architecture and capabilities, offering a comprehensive understanding of the interactions across its components. This is of crucial significance to developing an in-depth knowledge of the operational dynamics of the system and the impact of design choices on its performance.

Within the scope of our research, the system being analysed comprises three separate investing strategies: VAR(1)-AutoML, CWT-CNN, and FM-LSTM. These methods reflect the system developed for forecasting financial markets and maintaining investment portfolios.

The first element of our system, known as VAR(1)-AutoML consists of the following steps. First, let $P \in \mathbb{R}^{n \times m}$ denote the matrix of stock prices, where P_{ij} represents the price of the i^{th} stock at the j^{th} time point. Then, we find the log returns matrix $R \in \mathbb{R}^{n \times (m-1)}$, which can be calculated as $R_{ij} = \ln\left(\frac{P_{i(j+1)}}{P_{ij}}\right)$. Before applying PCA, the data is standardised. Let $R_{std} \in \mathbb{R}^{n \times (m-1)}$ be the standardised log returns matrix, where $R_{std_{ij}} = \frac{R_{ij} - \mu_i}{\sigma_i}$, and μ_i and σ_i are the mean and standard deviation of the i^{th} stock returns, respectively.

After standardisation, PCA can be applied to obtain the matrix $R_{pca} = \mathrm{PCA}(R_{std})$, where $R_{pca} \in \mathbb{R}^{n \times k}$, and $k \leq m - 1$ is the number of principal components chosen. The method of using retained principal components is called Kaiser's rule. According to this rule, it is recommended to retain only those components with eigenvalues exceeding 1, since they provide a greater quantity of information compared to single variables.

To effectively capture the temporal dynamics of stock prices and predict future returns, we use a rolling window method for VAR modelling. The VAR(1) model, a multivariate extension of the standard autoregressive model, is chosen for its capacity to represent the linear relationship present across numerous time series.

The method we developed is designed to calculate a series of VAR(1) models. Each model is built using a rolling window approach, where the window includes the most recent $L = 22$ trading days. The decision to use L coincides with the normal trading procedure, wherein each model is updated using a dataset that covers approximately 30 days of current data. The use of a rolling window method allows our system to respond effectively to fluctuations in market dynamics, as the estimated model parameters possess the ability to fluctuate over time.

The VAR(1) operates as follows:

1. The system is initialised by specifying an empty VAR(1) model structure V with p variables, where p is the number of stocks in the dataset.
2. The following steps are implemented for each trade day i (from the day after the initial L days to the final day in the dataset):
   - The VAR(1) model P_i is estimated using the logarithmic returns of the p stocks, spanning from day i to $i + L - 1$. The above method produces a set of VAR(1) coefficients of the model, which successfully describe the linear relationships between the stocks, using the L most recent days of data.
   - A vector E_i is initialised to include the predicted returns for each day i.
   - For every day j within the range of $i - L + 1$ (or day 1, if $i - L + 1$ is less than 1) to i, a one-step-ahead forecast is calculated using the VAR(1) model P_i and the log returns from day i to $i + L - 1$. The forecasts are collected and stored in the variable E_i.
   - Eventually, the predictions collected in E_i are averaged across the number of days used in the estimation, which is the minimum between i and L. This process results in an average forecast of returns for day i, specifically, for a one-step-ahead prediction.

Once this procedure is completed, a time series of expected returns E is obtained. Each E_i is calculated from a VAR(1) model, which is estimated using the most recent L days of data. The suggested approach combines VAR modelling, with its advantages for capturing linear connection among shares, with a rolling window method to efficiently react to changing market dynamics.

The division of data into training and testing sets is a critical stage in the application of deep learning or machine learning techniques. In this process, the data is partitioned, with 80% allocated for training and the remaining 20% reserved for testing. The purpose of the training dataset is to aid in the training of the system, while the testing dataset is used to assess and evaluate the performance of the model. Algorithm 3 below illustrates the model's procedure:

Algorithm 3 VAR(1)-AutoML investment strategy.

procedure StockPricePrediction(P, n, m)
  Initialise $R \in \mathbb{R}^{n\times(m-1)}$
  for $i = 1$ to n **do**
    for $j = 1$ to m−1 **do**
      $R_{ij} = \ln\left(\frac{P_{i(j+1)}}{P_{ij}}\right)$
      $R_{std_{ij}} = \frac{R_{ij}-\mu_i}{\sigma_i}$
    end for
  end for
  $R_{pca} = \mathrm{PCA}(R_{std})$ ▷ Apply PCA using Kaiser's Rule
  Split R_{pca} into $R_{pca_{tr}}$ and $R_{pca_{te}}$
  $model = \mathrm{fitrauto}(R_{pca_{tr}})$ ▷ Train the model
  Test the model on $R_{pca_{te}}$
  Determine important feature order in $R_{pca_{tr}}$ and $R_{pca_{te}}$, relying on the model outcomes by masking R_{pca}.
  Calculate cumulative log returns as CL_{tr} and CL_{te} for $R_{pca_{tr}}$ and $R_{pca_{te}}$, respectively
  return CL_{tr}, CL_{te}
end procedure

Important note: fitrauto(...) *is a model that trains with a tool that uses automated procedures to select between Bayesian optimisation and the Asynchronous Successive Halving Algorithm. These algorithms are applied to a range of regression model types with varying hyperparameter values. The output of fitrauto is the model anticipated to yield the most accurate predictions for new data.*

The second suggested approach – CWT-CNN – involves combining a wavelet transform with a CNN. The wavelet transform is used as a first step to divide the investment data into distinct frequency components. Subsequently, a CNN is employed to identify the important features of the data for portfolio optimisation. This model includes an important phase that involves the modification and extraction of features from a dataset with several dimensions. To achieve this, a CWT is performed on specific components of the data collection.

The technique begins by creating new transformed data points, denoted as Q, whose dimensions are determined by the span of the dataset dates (which is decreased by L elements from the end) and the parameter p. The matrix Q is arranged to enable the storage of the results of the repeated wavelet transformations.

After the creation of Q, the algorithm proceeds to participate in a two-level iteration loop. The outer loop sequentially traverses each element in the dataset P, which is a vector that stores the indices of the dataset. For every iteration i in the set P, a nested loop is initiated, which operates p times. The algorithm conducts a CWT on a section of the j-th column of the data matrix R_{pca}, covering the i-th row to row $i + L - 1$, within the nested loops.

The proposed approach involves the application of a CWT to different sections of the multi-dimensional dataset. The results of the transformation outputs are important for capturing vital information from the data. The feature extraction technique plays a crucial role in the data analysis pipeline of the system as a whole.

Algorithm 4 CWT-CNN investment strategy.

procedure StockPricePrediction(P, n, m)
  Initialise $R \in \mathbb{R}^{n\times(m-1)}$
  for $i = 1$ to n **do**
    for $j = 1$ to m−1 **do**
      $R_{ij} = \ln\left(\frac{P_{i(j+1)}}{P_{ij}}\right)$
      $R_{std_{ij}} = \frac{R_{ij}-\mu_i}{\sigma_i}$
    end for
  end for
  $R_{pca} = \text{PCA}(R_{std})$ ▷ Apply PCA using Kaiser's rule.
  Apply CWT on R_{pca}.
  Split R_{pca} into $R_{pca_{tr}}$ and $R_{pca_{te}}$.
  $model = \text{CNN}(R_{pca_{tr}})$ ▷ Train the model.
  Test the model on $R_{pca_{te}}$
  Determine important feature order in $R_{pca_{tr}}$ and $R_{pca_{te}}$, relying on the model outcomes by masking R_{pca}.
  Calculate cumulative log returns as CL_{tr} and CL_{te} for $R_{pca_{tr}}$ and $R_{pca_{te}}$, respectively.
  return CL_{tr}, CL_{te}.
end procedure

The third component, the FM-LSTM, implements the FMT in conjunction with an LSTM network. The FMT is employed to transform the time-series investment data into a representation that is invariant to changes in scale and rotation. Following this, an LSTM network is used to find the important features to control and optimise the portfolio.

Here is the outline of the FM-LSTM model:

Algorithm 5 FMT-LSTM investment strategy.

procedure StockPricePrediction(P, n, m)
  Initialise $R \in \mathbb{R}^{n\times(m-1)}$
  for $i = 1$ to n **do**
    for $j = 1$ to m−1 **do**
      $R_{ij} = \ln\left(\frac{P_{i(j+1)}}{P_{ij}}\right)$
      $R_{std_{ij}} = \frac{R_{ij}-\mu_i}{\sigma_i}$
    end for
  end for
  $R_{pca} = \text{PCA}(R_{std})$ ▷ Apply PCA using Kaiser's rule.
  Apply CWT on R_{pca}.
  Split R_{pca} into $R_{pca_{tr}}$, $R_{pca_{re}}$ and $R_{pca_{te}}$.
  Obtain geometric features from using Algorithm 2 on $R_{pca_{tr}}$ and $R_{pca_{te}}$, relying on $R_{pca_{re}}$.
  Assign FM_{tr} and FM_{te} as the matrix geometric features from training and testing data, respectively.
  $model = \text{LSTM}(FM_{pca_{tr}})$ ▷ Train the model.
  Test the model on $FM_{pca_{te}}$
  Determine important feature order in $FM_{pca_{tr}}$ and $FM_{pca_{te}}$, relying on the model outcomes by masking R_{pca}.
  Calculate cumulative log returns as CL_{tr} and CL_{te} for $R_{pca_{tr}}$ and $R_{pca_{te}}$, respectively.
  return CL_{tr}, CL_{te}.
end procedure

Here, $R_{pca_{re}}$ is the log return of the PCA reference dataset.

The computational procedures, including data preprocessing, model implementation, and statistical analysis, were performed using custom MATLAB scripts. These scripts are included in the Supporting Information section to ensure reproducibility.

Results

The portfolio consisted of 1,421 stocks, taken from US stock market between 1 April 2013 and 1 April 2023. The data was obtained from a website reliable for its financial reports and details using the MATLAB connection function whic is available in https://www.mathworks.com/help/datafeed/moneynet.html. The principal purpose of this initial research effort is to experiment with feature-based optimisation approaches. Therefore, the selection was random to ensure a diverse and unbiased sample. This randomness guarantees the assessment of the models avoids the influence of bias from specific sectors or companies.

Additionally, the random precursor allowed for testing of the robustness and generalisation capability of the optimisation models across a broad range of stocks.

We conducted a statistical analysis of the daily prices of the stocks in the portfolio. The aim of the analysis was to identify important features, including the stocks' central tendency, dispersion, and correlation, to capture important factors for investment strategies and portfolio management.

The investigated statistical measures included the mean, median, standard deviation, skewness, and kurtosis (summary statistic). As mentioned, these describe the central tendency and dispersion of returns. Correlation coefficients were computed to examine the relationships between different stocks.

To obtain an in-depth view of the data, visualisations, including time-series plots, histograms, box plots, and correlation heatmaps were created to provide a graphical representation of these statistics. We began with a visualisation of a time-series plot of the stock prices as presented in (Fig 3) below.

The daily stock prices for the firms under study are summarised in the Table 1 below, along with descriptive information. Mean and median price, standard deviation, skewness, and kurtosis are displayed in separate rows.

The average rate of price changes can be calculated using the mean. The median price represents the midpoint of the price distribution and is less volatile than the mean due to extreme values. We found that the average mean and median prices were between 2.108 and 2,870.409 and 1.171 and 2,704.705, respectively.

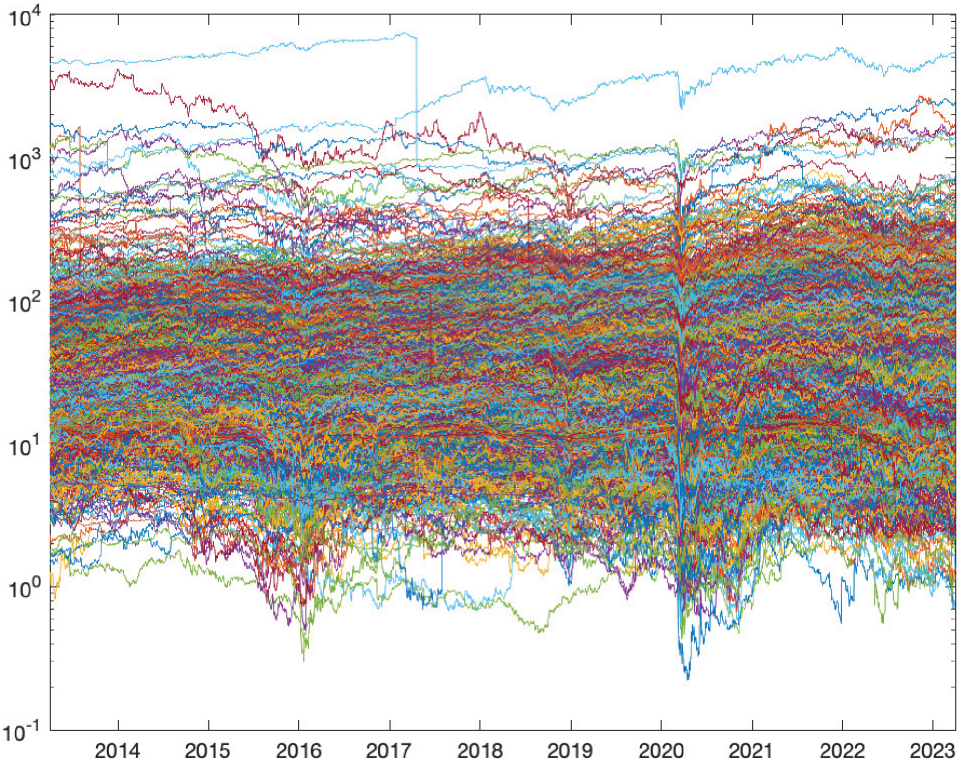

Fig 3. Time series plot of stock prices. The graph shows the performance of the portfolio optimization method using historical stock data.

https://doi.org/10.1371/journal.pone.0321204.g003

Table 1. Descriptive statistics on stock prices.

Measure	Min	Max	Mean	Median	StdDev
Mean Prices	2.108	2870.409	60.031	33.238	133.469
Median Prices	1.171	2704.705	54.893	31.430	108.048
Std Dev	0.281	2300.347	25.232	9.736	88.765
Skewness	-3.686	4.866	0.302	0.295	0.653
Kurtosis	1.186	26.017	2.845	2.625	1.232

https://doi.org/10.1371/journal.pone.0321204.t001

Using the standard deviation, the volatility of the returns can be assessed. Our research shows that the daily prices of these stocks can vary widely, based on the standard deviation of 0.281 to 2,300.347.

The skewness index quantifies the disproportion in a price distribution. In the case of positive skewness, the tail of the prices is far to the right, whereas in the case of negative skewness, it is far to the left. Our results exhibit skewness in the range of –3.686 to 4.866.

The tailedness of a price distribution can be quantified by its kurtosis. A greater kurtosis suggests a distribution with fatter tails, which may portend more dramatic price swings. Our price kurtosis fell between 1.186 and 26.017 degrees.

These numbers are useful since they shed light on the performance of these equities. There appears to be a wide range of stock performance, as indicated by the wide range of mean and median returns. There also appears to be a wide range of stock risk, as indicated by the large difference in standard deviation. Differences in the distribution of stock returns are further highlighted by the skewness and kurtosis, which exhibit substantial diversity across these equities.

We performed hierarchical clustering based on the correlation matrix of the stock prices calculation. Then, we derived measures from these samples. To obtain these measures through clustering, we applied the so-called linkage method. The figure was truncated to show only a few merges, as the 1,412 samples would be difficult to present using a heatmap.

To implement this approach, we used the following steps based on the daily stocks prices:

1. A correlation matrix among stock prices was computed.
2. A transformation was applied from the correlation matrix to the dissimilarity matrix.
3. The average linkage approach was used to apply hierarchical clustering to the dissimilarity matrix.
4. The resultant dendrogram was truncated so only the most recent N mergers are shown.

The stock returns are displayed in a dendrogram in (Fig 4), which shows their hierarchical grouping. The dendrogram is a tree-like diagram in which each node represents a stock, and the distance between nodes represents the degree of dissimilarity between groups of stocks. To simplify interpretation of the larger clusters, we trimmed the dendrogram to display only the most recent N mergers.

To group samples sets, we used an approach that relied on the average correlation with all other samples. The aim of using this approach was to capture the correlation structure in our large datasets comprised of stock prices. The steps for applying this method were as follows:

1. We began by computing the correlation matrix of the dataset using the Pearson correlation coefficient, which measures the linear relationship between two variables. The correlation coefficient ranges from –1 to 1, where –1 indicates a strong negative linear relationship, 1 a strong positive linear relationship, and 0 no linear relationship.
2. We then computed the average correlation for each sample by evaluating the mean of each row in the correlation matrix.
3. We divided the samples into three groups – low, medium, and high – depending on their average correlations. The thresholds for these groups were set randomly and could be adjusted as needed.

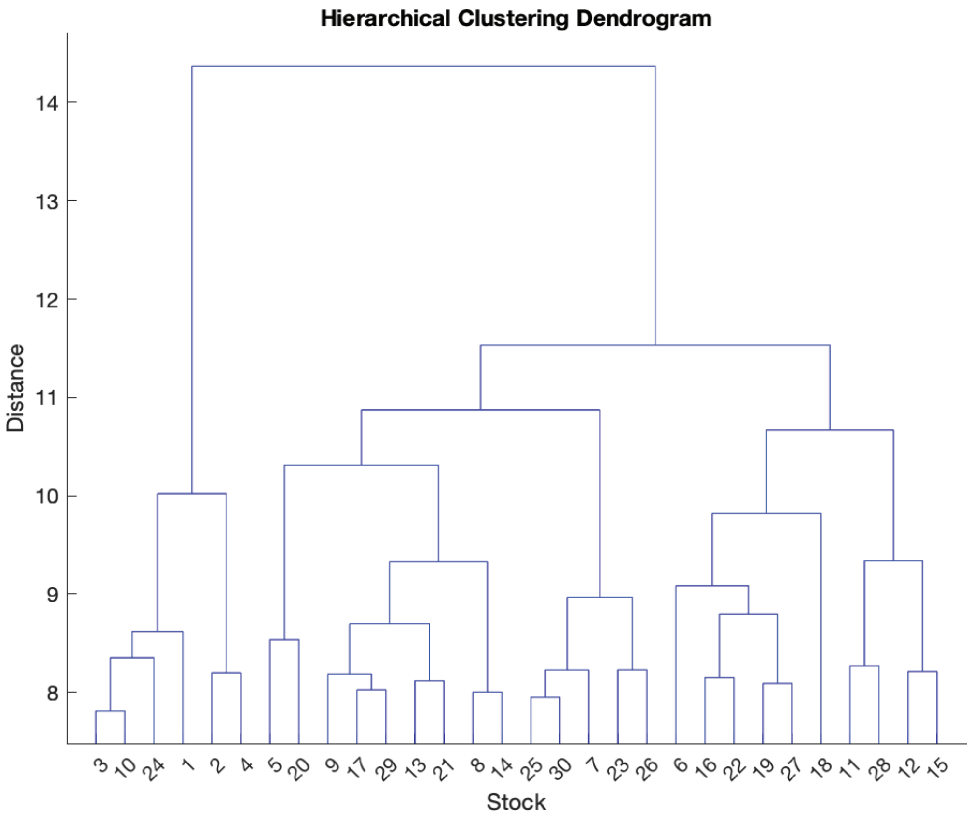

Fig 4. Dendrogram of the stock prices.

https://doi.org/10.1371/journal.pone.0321204.g004

Figs 5 and 6 present a bar chart showing the results of the average correlation of samples grouped by correlation level. The x-axis is the sample index, and the y-axis represents the average correlation values. The three groups are designated as follows: The low correlation group is coloured red, the medium group is green, and the high group is blue. The motivation for including this step was to obtain a clear visual representation of the groupings and allow for clear identification of patterns in the data. The fact that the results of averaging and grouping the correlation samples are sensitive to the specific thresholds used for the groupings should also be taken into account.

We also calculated the log returns of the stocks prices and applied PCA to the stocks' log returns. The Kaiser rule was employed to select the optimal principal components of the data, and the first 10 were kept as a representative of the stock returns of the portfolio. The aim of this strategy was to capture the most variance in the data using the fewest number of principal components. PCA removes outliers and unusual patterns while ensuring noise reduction, as outliers and noise have strong negative effects on identifying the best features for optimising the portfolio. Below is a plot of the first 10 principal components of the data see (Fig 7).

We plotted a heatmap of the correlation matrix for the 10 principal components obtained from the stock log returns derived from the data see (Fig 8). On the heatmap, the diagonal values are equal to 1. This means that each PCA correlated perfectly with itself. The numbers not on the diagonal, which show the relationships between major components, are near zero. This illustrates that these principal components are orthogonal and support the use of PCA. The cause of these small off-diagonal numbers is calculation error or background noise.

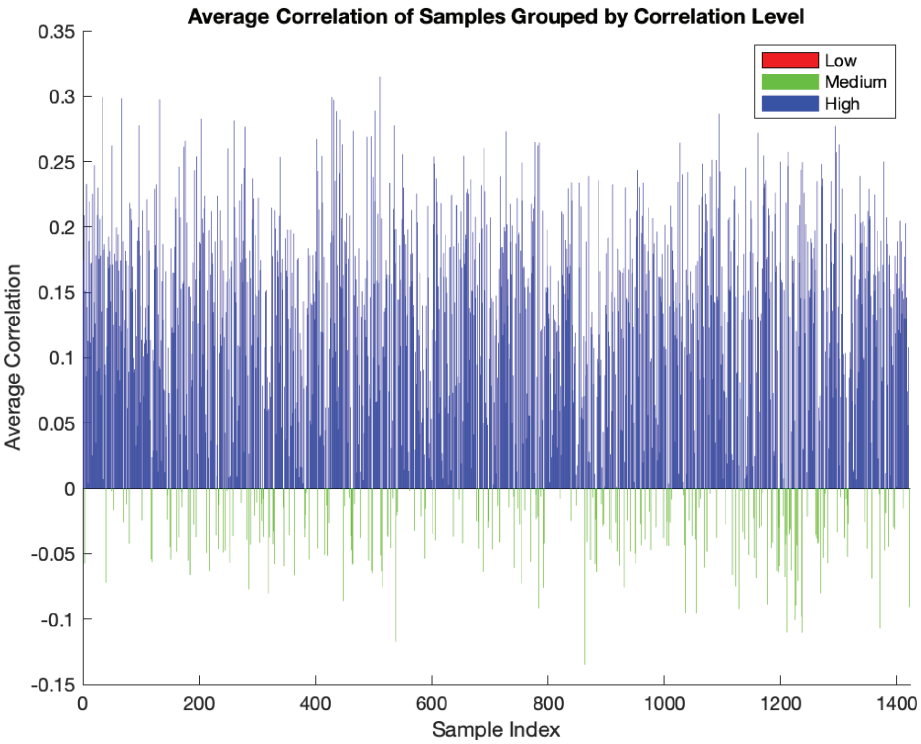

Fig 5. Here, −1, 0, and 1 are the thresholds used to group the correlations.

https://doi.org/10.1371/journal.pone.0321204.g005

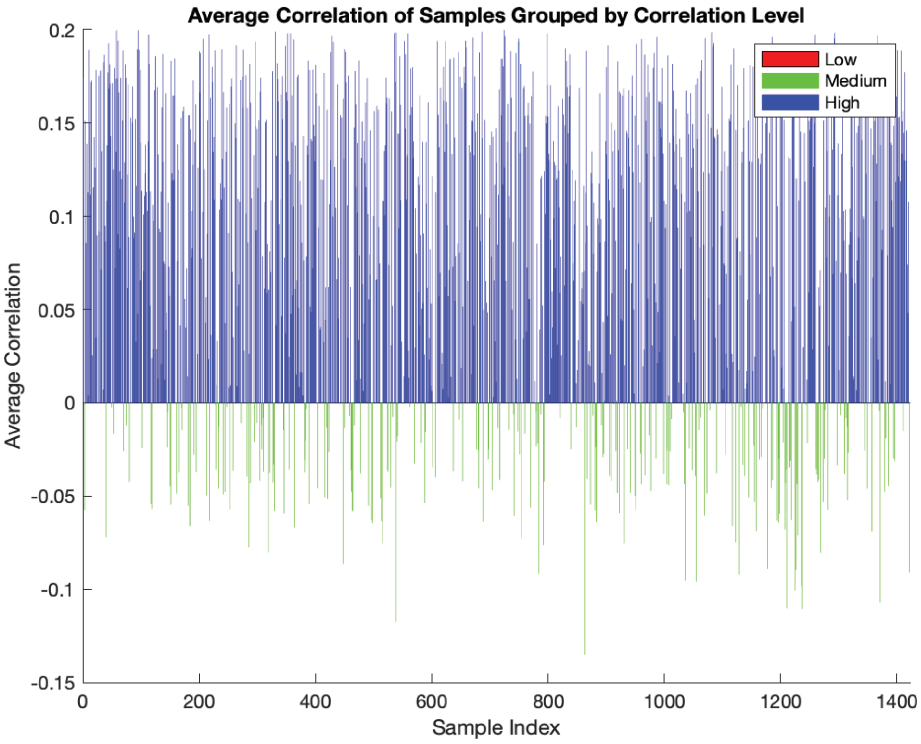

Fig 6. Here, −0.8, 0, and 0.2 are the thresholds used to group the correlations.

https://doi.org/10.1371/journal.pone.0321204.g006

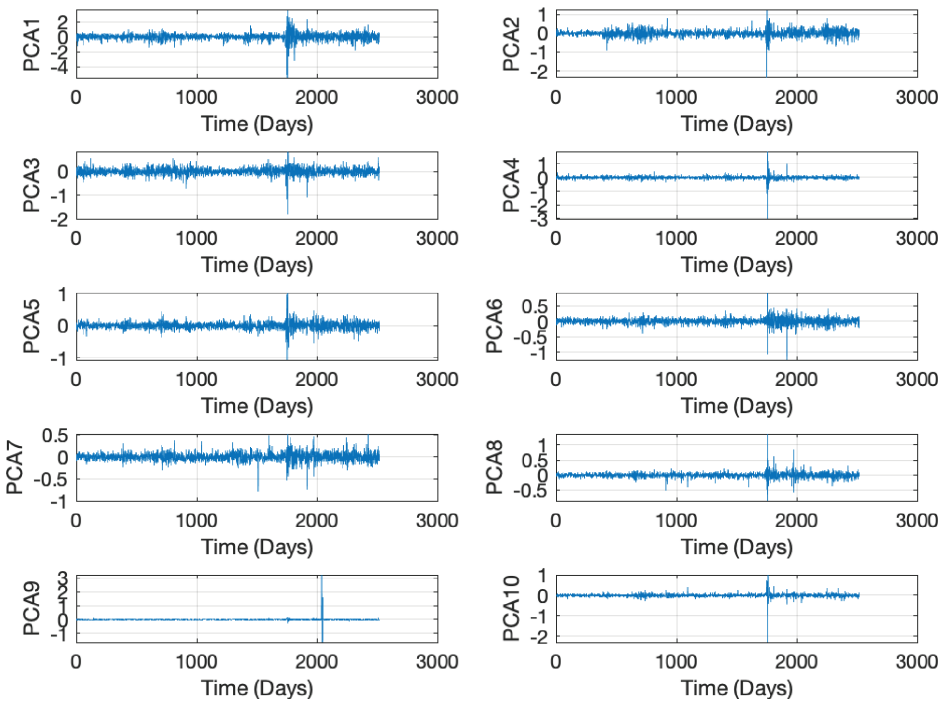

Fig 7. First 10 principal components of the stock log returns.

https://doi.org/10.1371/journal.pone.0321204.g007

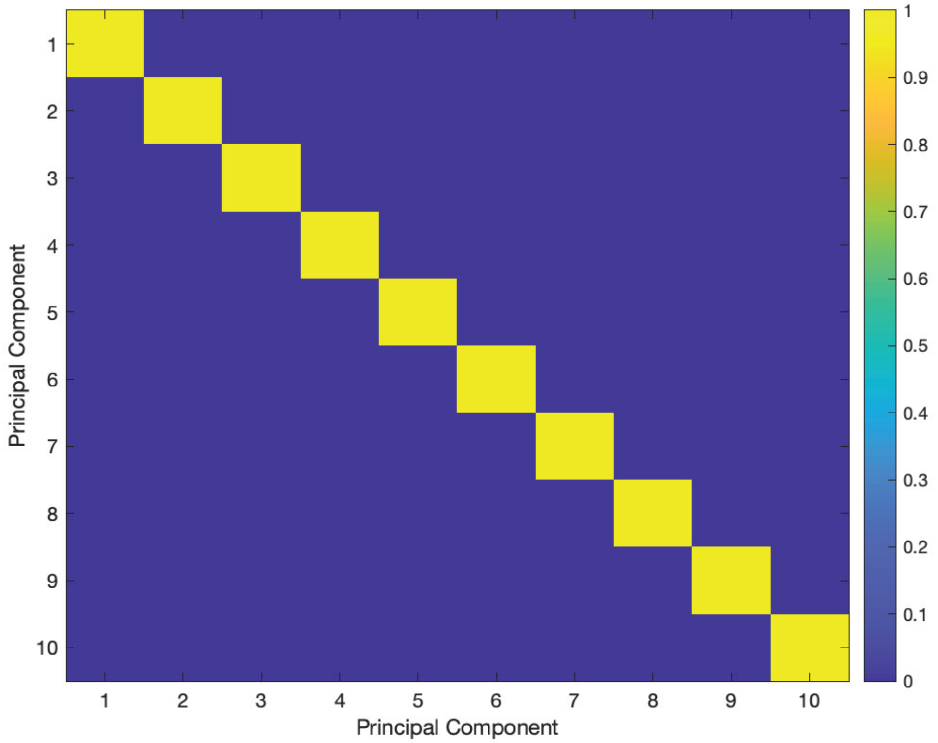

Fig 8. Correlation heatmap of the principal components.

https://doi.org/10.1371/journal.pone.0321204.g008

We also plotted the cumulative log returns of the principal components over time, considering PCA1 to PCA10. (Fig 9) shows that the initial investment was initialised at zero for all cumulative log returns. There are increasing and decreasing log returns over time from 0 to 3,000 days (plotted on the x-axis), which is indicative of the fluctuating nature of returns over time.

The cumulative log returns plotted along the y-axis fall between –1 and 9. Note that a value above zero indicates a positive return, while one below zero indicates a negative return or a loss. These results demonstrate the inherent volatility of the market, with notable fluctuations in cumulative returns for all principal components see (Fig 9).

We also plotted the cumulative log returns of these 10 principal components, as presented in Fig 10 below:

Fig 10 shows that the portfolio's cumulative log return decreases over time, with brief intervals of stability marked by less sharp decreases. Portfolio assets are dynamic and variable, which explains these changes. The steeper downwards slopes suggest larger negative returns, whereas the more flat areas indicate relative stability with no substantial positive log returns.

A wavelet transform divides a signal into frequency components and analyses each component with a scale-matched resolution. Time-series analysis uses wavelet transform to investigate non-stationary signals, such as financial data see (Fig 11).

To evaluate our model, VAR(1)-AutoML, CWT-CNN, and FM-LSTM were used to analyse the data. We compared the results of these investment approaches by creating profitability plots using cumulative log returns.

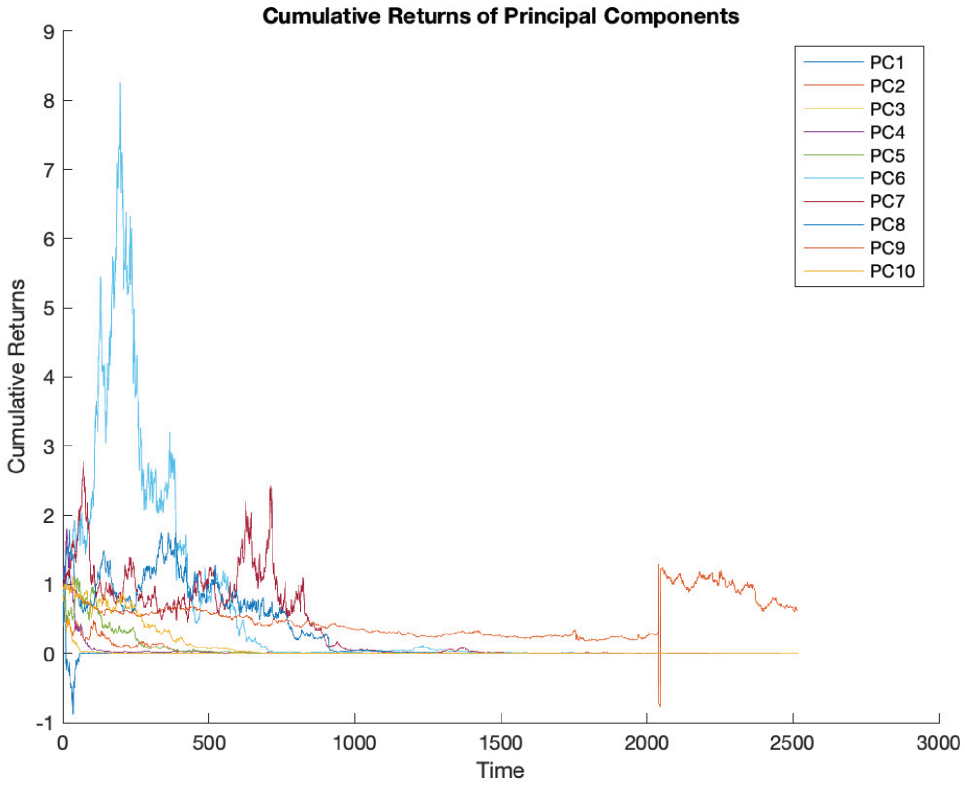

Fig 9. Cumulative returns of the principal components of the log returns.

https://doi.org/10.1371/journal.pone.0321204.g009

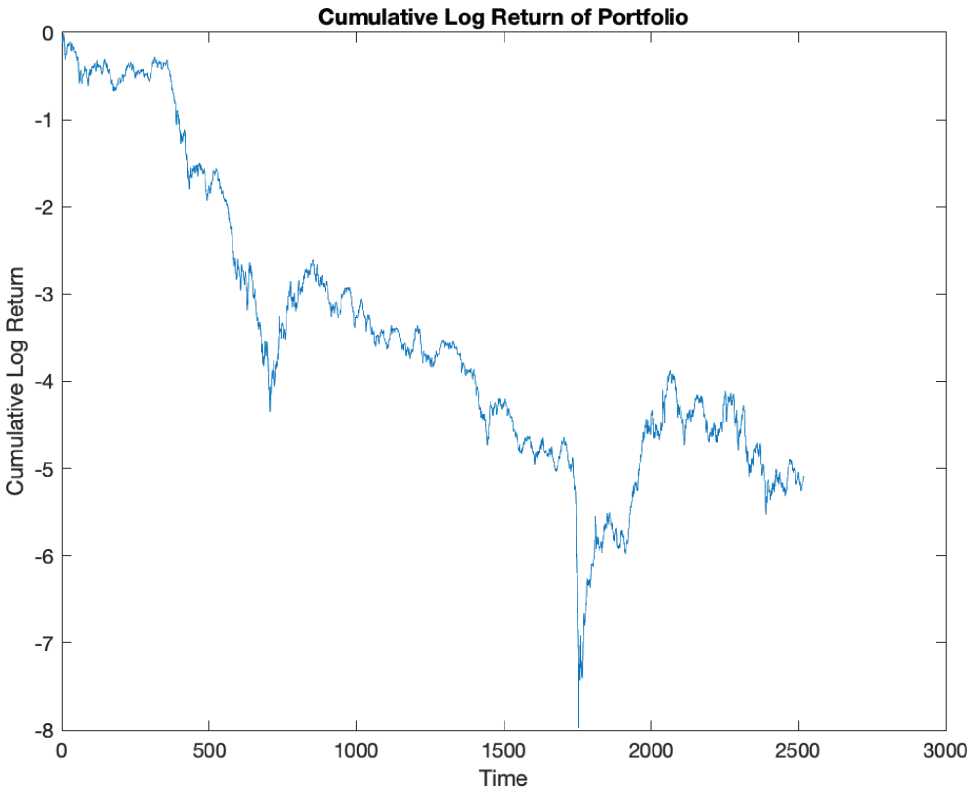

Fig 10. Cumulative returns of a portfolio containing the principal components of the log returns.

https://doi.org/10.1371/journal.pone.0321204.g010

Fig 12 visualises the paths of the three investment strategies for the 10-year period from April 2013 to April 2023. Cumulative log return was employed for comparative analysis, serving as an indicator of the total investment yield and taking into account the compounding factor.

The x-axis of the graph represents the 10-year time span from April 2013 to April 2023. The y-axis represents the cumulative log returns, ranging from –15 to 10. This range signifies the potential for substantial profits, as indicated by positive logarithmic returns, as well as losses, as indicated by the negative logarithmic returns, over a 10-year periods.

The trajectory of each approach on the chart represents its overall performance. The overall patterns indicate varying degrees of effectiveness. The FM-LSTM investment strategy has an upward trajectory on the graph, demonstrating its tendency to generate positive outcomes. In contrast, the VAR(1)-AutoML strategy exhibits periods of decline.

Mean absolute error (MAE) is a statistical metric that quantifies the differences between two observations that represent the same phenomenon. In this context, examples of observations (Y versus X) include expected and observed values, subsequent time and starting time, or a measuring technique and an alternate measurement approach [38]. We computed the MAE by adding the absolute errors, expressed as the Manhattan distance, and then dividing the sum by the sample size:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{49}$$

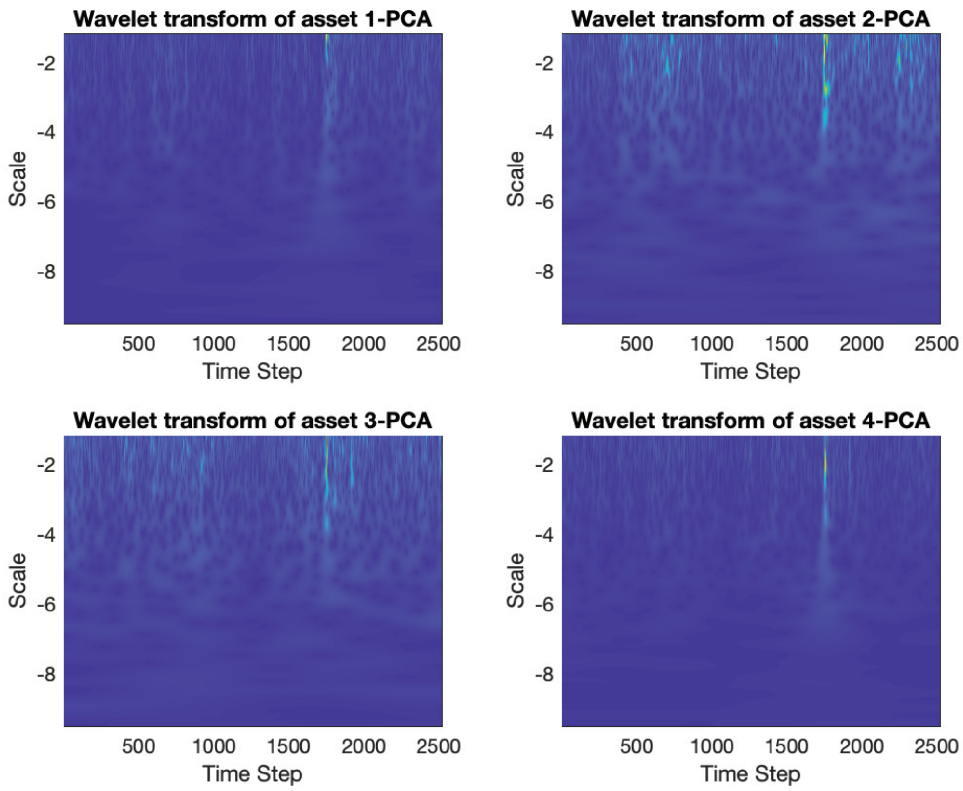

Fig 11. Wavelet transform of some of the principal components of log returns.

https://doi.org/10.1371/journal.pone.0321204.g011

Therefore, the MAE is the geometric mean of the absolute inaccuracies $|e_i| = |y_i - x_i|$, where y_i is the forecast, and x_i is the actual value. Relative frequencies can be used as weight factors in other formulations. To calculate the MAE, the scale must match that of the observed data. As the MAE is dependent on the scale being used, it cannot be used to compare predicted values measured on different scales [39]. In time series analysis, the MAE is frequently used to measure forecast error and is often confused for the more popular mean absolute deviation. The same misunderstanding is also present more broadly [40].

In statistics, the mean squared error (MSE) or mean squared deviation of an estimator quantifies the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value [41]. MSE is a risk function that represents the expected value of the squared error loss. The near-constant positivity of MSE, or its tendency not to drop to zero, arises from randomness or the estimator's inability to incorporate information that could yield a more precise estimate [42,43]. In machine learning, especially empirical risk minimisation, MSE can be used to represent the empirical risk, which is the average loss on a given dataset [44]. It is a close approximation of the real MSE, which is the average loss on the actual population distribution.

Given a vector of n predictions derived from a sample of n data points across all variables, where y_i represents the vector of observed values for the predicted variable, and $\hat{y}_i$ denotes the predicted values (e.g. obtained from a least-squares fit), the within-sample MSE of the predictor is calculated as [44]:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{50}$$

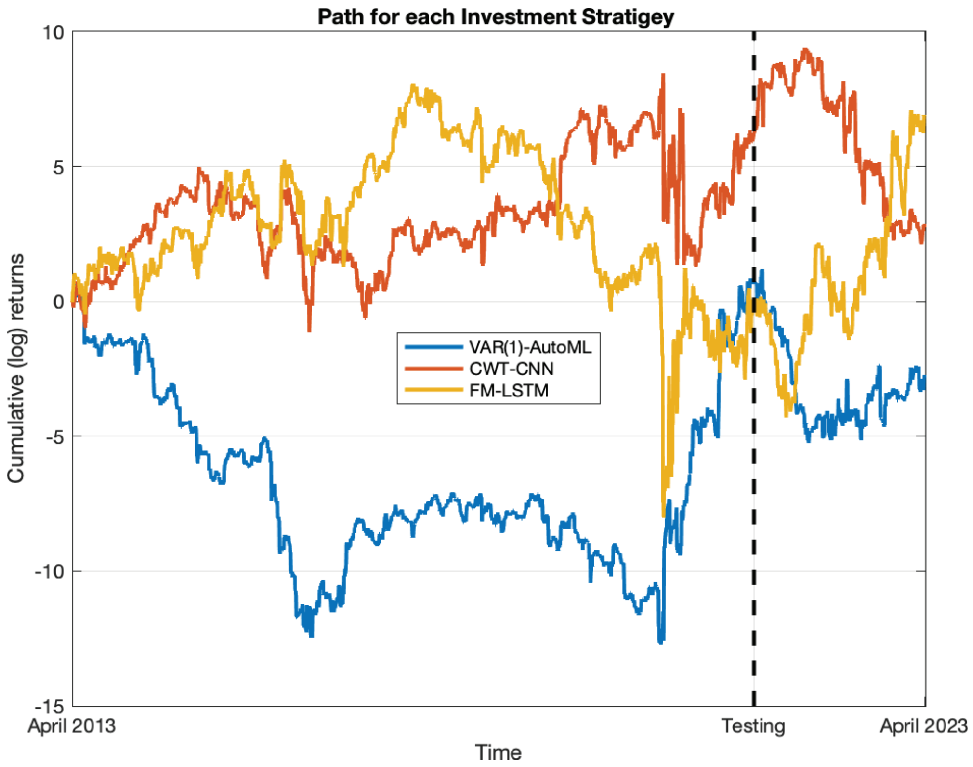

Fig 12. Performance evaluation: the profitability of VAR(1)-AutoML, CWT-CNN, and FM-LSTM.

https://doi.org/10.1371/journal.pone.0321204.g012

This study assessed the efficacy of three different investment methods – VAR(1)-AutoML, CWT-CNN, and FMT-LSTM – using MAE and MSE measures across several principal components. The findings shown in Table 2 demonstrate that CWT-CNN consistently achieved the lowest error rates across the majority of components, indicating enhanced prediction accuracy relative to the alternative approaches. Significantly, for principal component 9 (PC9), CWT-CNN exhibited remarkable accuracy, with an MAE of 0.014 and an MSE of 0.001. Conversely, FMT-LSTM demonstrated elevated error rates, especially for PC6 and PC8,

Table 2. MAE and MSE for the training set across the three methods.

Time Series	VAR(1)-AutoML		CWT-CNN		FMT-LSTM	
	MAE	MSE	MAE	MSE	MAE	MSE
PC1	0.316	0.332	0.336	0.532	0.708	0.930
PC2	0.152	0.073	0.141	0.121	0.168	0.128
PC3	0.138	0.049	0.131	0.065	0.321	0.174
PC4	0.099	0.033	0.092	0.037	0.523	0.336
PC5	0.106	0.030	0.085	0.039	0.122	0.048
PC6	0.102	0.021	0.063	0.016	0.823	0.760
PC7	0.091	0.019	0.073	0.016	0.396	0.204
PC8	0.092	0.020	0.066	0.020	0.838	0.745
PC9	0.066	0.008	0.014	0.001	0.316	0.139
PC10	0.077	0.021	0.057	0.023	Reference	Reference

https://doi.org/10.1371/journal.pone.0321204.t002

indicating possible paths for model enhancement. These findings underscore the efficacy of CWT-CNN for time series prediction tasks in this setting.

The findings shown in the testing dataset presented in Table 3 demonstrate that CWT-CNN consistently outperformed the other approaches, yielding reduced error rates for the majority of PCs. CWT-CNN attained the minimum MAE and MSE for PC1 to PC8, indicating its exceptional prediction proficiency. The FMT-LSTM model, although competitive in certain situations (e.g. PC2), typically exhibited elevated error rates, especially for PC6 and PC8. In particular, PC9 was a challenge for all models, exhibiting markedly elevated MSE values. The findings show that the CWT-CNN methodology was highly effective in identifying the fundamental patterns within this dataset, indicating it could improve accuracy in forecasting for analogue time series applications.

Fig 12 presents a comparison of the VAR(1)-AutoML, FM-LSTM, and CWT-CNN models in terms of their efficacy in optimising portfolio returns. Under particular circumstances, the FM-LSTM model exhibited exceptional returns, indicating its successful representation of certain market dynamics. Nevertheless, it is important to emphasise that this improved performance does not indicate an innate capacity to effectively forecast future market trends.

The performance outcomes may have been affected by the random decision to use 20% of the data for testing and 80% for training. The variability introduced by this sampling technique could compromise the model results. Although FM-LSTM produced better cumulative returns, its success better indicates flexibility to historical trends rather than forecasting ability.

By contrast, although it did not produce the same degree of returns, the CWT-CNN model provided an insightful analysis of temporal patterns and volatility. We also respect the complexity of financial markets and the restrictions of modelling techniques in dynamics prediction.

Further study is required to investigate the effects of several data splitting cases and improve model resilience. This will assist in understanding the actual predictive power of these models.

Conclusions and future work

This paper investigated three computational portfolio optimisation strategies employing a PCA of stock log returns. The approaches included VAR(1)-AutoML, CWT-CNN, and FM-LSTM. Each strategy obtained crucial stock return data to enhance the performance of financial portfolio prediction models.

Table 3. MAE and MSE for the testing dataset across the three methods.

Time Series	VAR(1)-AutoML		CWT-CNN		FMT-LSTM	
	MAE	MSE	MAE	MSE	MAE	MSE
PC1	0.491	2.525	0.464	2.480	0.794	2.833
PC2	0.240	0.139	0.180	0.104	0.195	0.108
PC3	0.174	0.571	0.147	0.565	0.350	0.697
PC4	0.096	0.112	0.090	0.112	0.532	0.431
PC5	0.152	0.040	0.104	0.021	0.139	0.033
PC6	0.132	0.060	0.093	0.049	0.844	0.815
PC7	0.139	0.095	0.101	0.084	0.409	0.262
PC8	0.131	0.179	0.095	0.174	0.873	0.904
PC9	0.536	10.098	0.166	6.371	0.451	6.626
PC10	0.095	0.086	0.068	0.079	Reference	Reference

https://doi.org/10.1371/journal.pone.0321204.t003

The VAR(1)-AutoML technique made use of VAR(1)'s statistical power for time series forecasting and AutoML's scalability for model selection and hyperparameter tweaking. This combination accurately captured linear connections and changes in the data over time, providing a solid platform for more complex models.

In contrast, the CWT-CNN approach used the CWT to extract features and decompose financial data series into time-frequency components. Combining these multi-resolution analytical capabilities with the pattern recognition strength of a CNN facilitated the discovery of complex non-linear patterns in data that standard approaches ignore.

After conducting a thorough investigation in the frequency domain using the FMT, the FM-LSTM approach used LSTM networks to model the sequences. This technique helped capture long-term dependencies and cyclic patterns in the data, which are essential for identifying trends and producing more accurate stock performance projections.

While each model has its own benefits, the FM-LSTM model demonstrated promising cumulative log return values, a critical performance indicator in this study. However, the random selection of 20% testing and 80% training data may have influenced the results. For this reason, the results indicate the model's ability to adapt to these data in the portfolio rather than its prediction capability. Even given this limitation, our model can be considered a potentially beneficial tool for portfolio managers to investigate the the uncertain events in volatile financial markets.

The use of PCA as a preprocessing step reduced dimensionality and highlighted the most important characteristics, improving model performance and portfolio optimisation.

Our work demonstrates how improved data processing and machine learning models can affect financial market forecasts. The efficiency of the FM-LSTM model indicates its potential application in examining and predicting long-term financial trends. A future study could combine these approaches or use additional data sources to improve the stability and adaptability of portfolio optimisation systems. While our model exhibits some predictive power, it is important to consider certain constraints. First of all, the FMT can show sensitivity to noise, limiting the model's performance on unclean data. When the FMT and LSTM are combined, the data dimensionality may also be reduced to a small number of features, making the prediction process difficult. Particularly in the case of sparse data, the complexity of the integrated model could lead to overfitting. A lack of data points in high-dimensional domains prevents algorithms from identifying significant trends without overfitting. This can significantly impact a model's generalisability and performance. The issue of dimensionality represents a major obstacle in this study, since it lowers the capacity of predictive models, including LSTM networks, to appropriately forecast financial events. Solving this problem requires increasing the model's dependability and effectiveness. Feature selection approaches or other dimensionality reduction strategies are essential for this purpose.

To make FM-LSTM models function effectively, it is important to find reference data points that can be used to obtain geometric features from both the training and testing datasets. The number of reference points directly influences the richness of the features obtained, but increasing it may result in higher processing costs and complexity. This requirement imposes a significant limitation, since an excessive number of features may cause overfitting, while too few may result in underfitting and suboptimal model performance. Achieving a balance in the quantity of reference points is critical for improving model precision and efficacy, and careful analysis is required in the design of FM-LSTM applications.

Although our methodology has yielded encouraging results, numerous questions remain unanswered. To enhance the model's generalisability, it would be beneficial to use a more diverse inter-class asset, such as cryptocurrencies. Additionally, attempting other designs or

including different methods, such as mean-variance optimisation, could improve the FM-LSTM model. Combining the GARCH and DCC-GARCH models with FM-LSTM could also help obtain a deep understanding of volatility dynamics. Applying the model to real-time data would further allow for its performance to be tested in dynamic circumstances. Finally, to improve decision-making, it may be useful to check whether the model's predictions are simple to understand. The proposed models could also be applied in other fields, such as for classification problems in medical diagnosis and epidemics.

Supporting information

S1 Code. main.m This script executes the primary analysis, running the three models and applying profitability, error, training, and testing analyses.
(PDF)

S2 Code. stack.m Helper function to stack data into a 2D vector using feature extraction for CNN and LSTM.
(PDF)

S3 Code. FM.m Applies the Fourier-Millen transform to extract geometric features.
(PDF)

S4 Code. hipass_filter.m High-pass filter function used within FM.m.
(PDF)

S5 Code. error_analysis.m Computes MAE and RMSE and plots them as time series.
(PDF)

S6 Code. calculateMetrics_csv_1.m Applies MAE and RMSE on VAR(1)-AutoML and CWT-CNN, storing results as CSV files.
(PDF)

S7 Code. calculateMetrics_csv_OO.m Applies MAE and RMSE on FM-LSTM.
(PDF)

S8 Code. transform_Image.m Transforms 2D images to obtain geometric features.
(PDF)

S9 Code. statistical_analysis.m Implements statistical analysis.
(PDF)

S1 File. Matlabcode.
(PDF)

S2 File. Variable table.
(PDF)

Author contributions

Conceptualization: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Data curation: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Formal analysis: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Funding acquisition: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Investigation: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Methodology: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Project administration: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Resources: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Software: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Supervision: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Validation: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Visualization: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Writing – original draft: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

Writing – review & editing: Muhammad Alkhudaydi, Aiedh Mrisi Alharthi.

References

1. Kremmel T, Kubalík J, Biffl S. Software project portfolio optimization with advanced multiobjective evolutionary algorithms. Appl Soft Comput. 2011;11(1):1416–26.
2. Ta VD, Liu CM, Tadesse DA. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. Appl Sci. 2020;10(2):437.
3. Qu H. Risk and diversification of nonprofit revenue portfolios: Applying modern portfolio theory to nonprofit revenue management. Nonprofit Manage Leadership. 2019;30(2):193–212.
4. Platanakis E, Urquhart A. Portfolio management with cryptocurrencies: The role of estimation risk. Econ Lett. 2019;177:76–80.
5. Feng Q, Tao S, Liu C, Qu H. An improved Fourier-Mellin transform-based registration used in TDI-CMOS. IEEE Access. 2021;9:64165–78.
6. Ali F, Sensoy A, Goodell JW. Identifying diversifiers, hedges, and safe havens among Asia Pacific equity markets during COVID-19: New results for ongoing portfolio allocation. Int Rev Econ Fin. 2023;85:744–92.
7. Ali F, Khurram M, Sensoy A, Vo X. Green cryptocurrencies and portfolio diversification in the era of greener paths. Renew Sustain Energy Rev. 2024;191:114137.
8. Ozbayoglu AM, Gudelek MU, Sezer OB. Deep learning for financial applications: A survey. Appl Soft Comput. 2020;93:106384. https://doi.org/10.1016/j.asoc.2020.106384
9. Zhang Z, Zohren S, Roberts S. Deep learning for portfolio optimization. arXiv preprint. 2020. https://doi.org/arXiv:2005.13665
10. Sang Y-F. A review on the applications of wavelet transform in hydrology time series analysis. Atmos Res. 2013;122:8–15. https://doi.org/10.1016/j.atmosres.2012.11.003
11. Çınar A, Tuncer SA. Classification of normal sinus rhythm, abnormal arrhythmia and congestive heart failure ECG signals using LSTM and hybrid CNN-SVM deep neural networks. Comput Methods Biomech Biomed Eng. 2021;24(2):203–14. https://doi.org/10.1080/10255842.2020.1821192 PMID: 32955928
12. Reddy G, Reddy M, Lakshmanna K, Kaluri R, Rajput D, Srivastava G. Analysis of dimensionality reduction techniques on big data. IEEE Access. 2020;8:54776–88.
13. Hasan B, Abdulazeez A. A review of principal component analysis algorithm for dimensionality reduction. J Soft Comput Data Mining. 2021;2(1):20–30.
14. Antonakakis N, Chatziantoniou I, Gabauer D. Refined measures of dynamic connectedness based on time-varying parameter vector autoregressions. J Risk Fin Manage. 2020;13(4):84.
15. Warsono W, Russel E, Wamiliana W, Widiarti W, Usman M. Vector autoregressive with exogenous variable model and its application in modeling and forecasting energy data: Case study of PTBA and HRUM energy. Int J Energy Econ Policy. 2019;9(2):390–8.
16. Mangalathu S, Jeon J. Ground motion-dependent rapid damage assessment of structures based on wavelet transform and image analysis techniques. J Struct Eng. 2020;146(11):04020230.
17. Dolas P, Sharma V. Generalized offset Fourier-Mellin transform & its analytical structure. Int J Innov Sci Eng Technol. 2020;7(10).
18. Xu Q, Kuang H, Kneip L, Schwertfeger S. Rethinking the Fourier-Mellin transform: Multiple depths in the camera's view. Remote Sens. 2021;13(5):1000.
19. Sims CA. Macroeconomics and reality. Econometrica: J Econometr Soc. 1980:1–48.

20. Hamilton JD. Time series analysis. Princeton University Press; 2020.
21. Lütkepohl H. New introduction to multiple time series analysis. Springer Science & Business Media; 2005.
22. Kilian L. Structural vector autoregressions. In: Handbook of research methods and applications in empirical macroeconomics. Edward Elgar Publishing; 2013. p. 515–54.
23. Phillips PC. Fully modified least squares and vector autoregression. Econometrica: J Econometr Soc. 195;63(5):1023–78.
24. Dhingra V, Sharma A, Gupta SK. Sectoral portfolio optimization by judicious selection of financial ratios via PCA. Optim Eng. 2023:1–38.
25. Jolliffe IT, Cadima J. Principal component analysis: A review and recent developments. Philos Trans A Math Phys Eng Sci. 2016;374(2065):20150202. https://doi.org/10.1098/rsta.2015.0202 PMID: 26953178
26. Jolliffe IT. Principal component analysis for special types of data. Springer; 2002.
27. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1798–828. https://doi.org/10.1109/TPAMI.2013.50 PMID: 23787338
28. Daubechies I. Ten lectures on wavelets. SIAM; 1992.
29. Stephane M. A wavelet tour of signal processing; 1999.
30. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276
31. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 2019;31(7):1235–70. https://doi.org/10.1162/neco_a_01199 PMID: 31113301
32. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
33. Lilly J, Olhede S. Generalized Morse wavelets as a superfamily of analytic wavelets. IEEE Trans Signal Process. 2012;60(11):6036–41.
34. Yoo Y, Baek JG. A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network. Appl Sci. 2018;8(7):1102.
35. Sadouk L. CNN approaches for time series classification. In: Time series analysis-data, methods, and applications. 2019;5.
36. StackExchange. Drawing a CNN with Tikz; 2019. Available from: https://tex.stackexchange.com/questions/439170/drawing-a-cnn-with-tikz.
37. Qin-Sheng Chen, Defrise M, Deconinck F. Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. IEEE Trans Pattern Anal Machine Intell. 1994;16(12):1156–68. https://doi.org/10.1109/34.387491
38. Willmott C, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim Res. 2005;30(1):79–82.
39. Hyndman R. Forecasting: Principles and practice. OTexts; 2018.
40. Pontius R, Thontteh O, Chen H. Components of information for multiple resolution comparison between maps that share a real variable. Environ Ecol Stat. 2008;15:111–42.
41. Pishro-Nik H. Introduction to probability, statistics, and random processes. LLC Blue Bell, PA, USA: Kappa Research; 2014.
42. Bickel PJ, Doksum KA. Mathematical statistics: Basic ideas and selected topics, volumes I-II package. Chapman and Hall/CRC; 2015.
43. Lehmann EL, Casella G. Theory of point estimation. Springer Science & Business Media; 2006.
44. James G, Witten D, Hastie T, Tibshirani R, Taylor J. An introduction to statistical learning: With applications in python. Springer Nature; 2023.