Deep Learning for Financial Time Series Forecast Fusion and Optimal Portfolio Rebalancing

Siddeeq Laher
School of CSAM
University of the Witwatersrand
Johannesburg, South Africa
siddeeqlaher@gmail.com

Andrew Paskaramoorthy
School of CSAM
University of the Witwatersrand,
Johannesburg, South Africa
andrew.paskaramoorthy@wits.ac.za

Terence L. van Zyl
Institute for Intelligent Systems
University of Johannesburg,
Johannesburg, South Africa
tvanzyl@gmail.com

***Abstract*—Portfolio selection is complicated by the difficulty of forecasting financial time series and the sensitivity of portfolio optimisers to forecasting errors. To address these issues, a portfolio management model is proposed that makes use of Deep Learning Models for weekly financial time series forecasting of returns. Our model uses a late fusion of an ensemble of forecast models and modifies the standard mean-variance optimiser to account for transaction costs, making it suitable for multi-period trading. Our empirical results show that our portfolio management tool outperforms the equally-weighted portfolio benchmark and the buy-and-hold strategy, using both Long Short-Term Memory and Gated Recurrent Unit forecasts. Although the portfolios are profitable, they are also sub-optimal in terms of their risk to reward ratio. Therefore, greater forecasting accuracy is necessary to construct truly optimal portfolios.**

***Index Terms*—deep learning, forecasting, financial time series, portfolio management**

I. Introduction

Portfolio selection involves analysing the investible asset universe to determine which combination of securities might be profitable in the next trading period [1]. Then, an investor's predictions are combined with his constraints within a portfolio selection framework. The intent is to produce weights for each security, which represent how the investor's capital will be allocated. At the end of the trading period, the portfolio's performance is evaluated and rebalanced to meet the investor's constraints [2].

There exist numerous frameworks to determine portfolio weights, with Mean-Variance Optimisation (MVO) being the most common choice [3]. However, there are numerous challenges when using MVO to construct portfolios. One challenge is a sensitivity to estimation errors in the input parameters, which leads to inefficient portfolio weights [4]. Another challenge is that MVO is a single-period framework that leads to significant turnover when applied in a multi-period setting, accumulating substantial transaction costs.

Improvements in forecasting can theoretically reduce these estimation errors resulting in better-performing portfolios [5]. However, some central postulates in the financial literature suggest that this task may be futile. In particular, the strong form of the Efficient Market Hypothesis (EMH) states that asset prices "fully reflect all available information" [6]. The EMH suggests that asset analysis will not result in consistent profits over time as security prices adjust rapidly to the arrival of new information. Numerous studies have questioned the validity of the EMH with mixed consensus [7, 8].

Despite the EMH, many studies have tried to analyse financial time series to forecast future returns. This approach is valid to the extent that historical returns are representative of the future distribution of returns. However, the distribution of returns is constantly changing, sometimes suddenly and violently, as market participants adapt to environmental conditions. This makes forecasting challenging, with many models failing to capture the underlying non-linear trends [9].

There is growing interest in the use of non-linear models, such as Machine Learning (ML) models, to achieve better forecasting accuracy. Deep Learning (DL), a subset of ML, has achieved significant success in Computer Vision and Natural Language Processing and has shown some promise in finance [10]. Among the advantages of DL models, are that they can process large amounts of data, extract non-linear trends and perform cross-learning, making them suitable for financial problems [11].

This research aims to compare the ability of a late fusion of Long Short Term Memory (LSTM) and a Gated Recurrent Unit (GRU) models to forecast weekly stock returns in our multi-period adjusted portfolio management framework. Our framework uses MVO to rebalance the portfolio weekly whilst accounting for transaction costs. In our implementation, we construct our forecasts using common technical analysis indicators. Our results show that our framework can generate returns in excess of the equally weighted benchmark, net transaction costs.

II. Related Work

DL models can uncover patterns in time series data with more success than conventional ML and statistical models [12–17]. Among the DL models, LSTM has proven to be the most successful model [12–14, 16, 17] which can be explained by its ability to learn long-term dependencies in time series data [10, 12, 13]. "Pure" ML models often produce inferior time series forecasts, possibly due to difficulties in determining optimal parameters from limited data [14] or their tendency to overfit [18]. Makridakis *et al.* [18] suggests combining multiple models to improve accuracy through a combination or hybrid of forecast models. The different models discover different components of the patterns in the time series, and individual model errors can be reduced through ensemble learning.

Wu and Gao [14] conduct a study that compares the ability of AdaBoost ensemble and non-ensemble ML models to forecast exchange rates and market indices against their singular model equivalents. Their results show that the ensemble models significantly outperform the singular models, with the ensemble LSTM being the most accurate and ARIMA being the worst.

Hossain *et al.* [10] tests the ability of an LSTM and a GRU to forecast stock prices. They find that the LSTM forecasts are overestimated, and the GRU forecasts are underestimated. A hybrid LSTM-GRU model is used to improve these results by offsetting the singular models' forecasts. A downside to this approach is significant variation in stock prices, making it more challenging to forecast than the returns.

Takeuchi and Lee [19] forecast stock price returns using a stacked restricted Boltzmann machine (RBM) as a classifier to forecast stock returns. The portfolio is constructed using a momentum strategy by longing the top 50% and shorting the bottom 50% of stocks. The results show that the model achieves good returns, outperforming a basic momentum strategy. As risk is not accounted for in this strategy, Lee and Yoo [13] construct threshold-based portfolios that account for risk by pre-selecting stocks based on their risk-reward characteristics. They find that the LSTM achieves greater accuracy than a simple Recurrent Neural Network (RNN) and a GRU for forecasting stock prices. A drawback of the implemented strategies is that they incur non-trivial trading costs [13, 19].

Fischer and Krauss [12] and Zhou [16] adopt similar methodologies to that of Takeuchi and Lee [19] and achieve similar results using an LSTM to forecast price directional movements and use constant transaction costs when evaluating portfolio net returns. Zhou [16] uses TAQ intraday data to estimate transaction costs and an LSTM that incorporates fundamental features to forecast stock returns. The LSTM portfolio returns turn negative after accounting for transaction costs, highlighting the importance of transaction costs for portfolio selection. Even though Zhou [16] poses a solution to deliver more accurate results by accounting for transaction costs, a step forward is to include transaction costs into the portfolio selection model to minimise costs and maximise returns.

Hsiao *et al.* [15] achieve good results constructing mean-variance portfolios from forecasting stock prices and the covariance matrix using a bootstrapping RNN. Wang *et al.* [17] makes use of technical indicators and lagged variables to forecast stock returns. The focus of the study is on the impact of stock pre-selection in constructing mean-variance portfolios, where pre-selection is based on forecasted returns. They find that a combination between the LSTM, pre-selected stocks and MVO to construct a portfolio outperforms an SVM and random pre-selection.

III. Background

Sezer *et al.* [11] found that RNN DL models are the most popular for financial time series forecasting. However, training RNNs to capture long term dependencies is difficult due to vanishing or exploding gradients [12, 13]. Gated RNNs such as the LSTM and GRU overcome these problems and are effective for sequential tasks [20].

A. Long Short-Term Memory

The description of LSTMs follow from [10, 12–14, 17, 20]. An LSTM consists of an input layer, one or more hidden layers, and an output layer. The defining characteristic of LSTMs is their memory cells contained in their hidden layers. The memory cell consists of the cell state (s_t), forget gate (f_t), input gate (i_t), and output gate (o_t) where t represents the current timestep of the memory cells' internal representation. The three gates control the flow of information in the memory cell and are responsible for updating the cell state. The forget gate determines the information deleted from the cell state, the input gate determines the information added to the cell state, and the output gate determines the information from the cell state that should be used for the layers' output (h_t). The formulae for the gates, cell state and layers' output are as follows:

$$
\begin{aligned}
f_t &= \sigma(U_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \\
\tilde{s}_t &= \sigma(U_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}) \\
i_t &= \sigma(U_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \\
s_t &= f_t \odot s_{t-1} + i_t \odot \tilde{s}_t \\
o_t &= \sigma(U_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \\
h_t &= o_t \odot tanh(s_t)
\end{aligned}
$$

where $h_0 = s_0 = 0$, $\sigma(\cdot)$ represents the sigmoid function, $\tanh(\cdot)$ is the hyperbolic tangent function, $\odot$ represents the Hadamard product, x_t is the current input, $U_{f,x}$, $U_{\tilde{s},x}$, $U_{i,x}$, and $U_{o,x}$ are input weight matrices, $W_{f,h}$, $W_{i,h}$, and $W_{o,h}$ are recurrent weight matrices into the respective gates, $W_{\tilde{s},h}$ is a recurrent weight matrix into the memory cell, and b_f, $b_{\tilde{s}}$, b_i, and b_o are biases. The biases are initialised to 0 except for the forget gate where b_f if initialised to 1.0.

B. Gated Recurrent Units

The description of GRUs follow from [10, 13, 20]. GRUs are similar to LSTMs with the difference that a single gate controls the forgetting factor and updates the cell state. Therefore, GRUs require fewer parameters and can be trained faster. A memory cell in a GRU consists of an update gate:

$$u_t = \sigma(U_{u,x}x_t + W_{u,h}h_{t-1} + b_u),$$

and reset gate:

$$r_t = \sigma(U_{r,x}x_t + W_{r,h}h_{t-1} + b_r).$$

The update gate determines how much information is passed into the future and the reset gate determines how much information should be forgotten. The formulae for the output layer are:

$$h'_t = tanh(U_{h,x}x_t + W_{h,h}(r_t \odot h_{t-1}) + b_h)$$
$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot h'_t$$

where $h_0 = 0$, $U_{u,x}$, $U_{r,x}$, and $U_{h,x}$ are input weight matrices, $W_{u,h}$ and $W_{r,h}$, are recurrent weight matrices into the respective gates, $W_{h,h}$ is a recurrent weight matrix into the memory cell, and b_u, b_r, and b_h are biases.

C. Mean Variance Optimisation

Mean-variance optimisation is a mathematical framework for portfolio selection that aims to minimise risk whilst maximising expected returns. In the original formulation [3], a quadratic objective function is used to select an optimal portfolio subject to an investor's desired trade-off between the mean and variance of a portfolio's return. This formulation can be extended to constrain portfolio weights to be positive and is given by:

$$\begin{aligned} \max_{w}. \quad & w^{\mathrm{T}}\mu - \frac{\delta}{2}w^{\mathrm{T}}\Sigma w \\ \text{s.t.} \quad & 0 \le w_i \le 1, \forall i \in \{0, n\}; \qquad w^{\mathrm{T}}\mathbf{1} = 1 \end{aligned} \tag{1}$$

where w is the vector of portfolio weights, μ is the vector of expected returns, δ is an investor's risk aversion factor reflecting his desired trade-off between risk and return, Σ is the covariance matrix of returns, $\mathbf{1}$ is a vector of ones, and n is the number of stocks being considered. The constraints ensure that the portfolio is a fully-invested long-only portfolio.

Since the expected returns and the covariance matrix of returns are unknown in practice, these must be estimated. Consequently, they contain estimation errors that are recognised (post-hoc) as substantially inflated estimates of returns or deflated estimates of variances, leading to suboptimal portfolio allocations [4].

To reduce these estimation errors, shrinkage estimators can be used to estimate the means and the covariance matrix [21, 22]. Alternatively, the impact of these estimation errors can be reduced by adding a penalisation term to the portfolio weights in the MVO objective function [23] as follows:

$$\max_{w} \quad w^{\mathrm{T}}\mu - \frac{\delta}{2}w^{\mathrm{T}}\Sigma w - \gamma||w||^2. \tag{2}$$

The L_2-constrained MVO with the regularisation parameter γ adjusts the objective function in Equation 1. The penalisation term is a form of L_2 regularisation which stabilises the amplification of errors arising from the smaller eigenvalues of the covariance matrix.

IV. Methodology

A. Dataset and Feature Engineering

The dataset consists of daily historical data of 15 stocks listed on the Johannesburg Stock Exchange (JSE) provided by ShareMagic. The dataset spans 21 years from January 1999 to December 2019 with 5478 data points. The first 200 data points are used for feature engineering to generate 31 features that are fed into the models. The features generated are variations of log-returns over m days:

$$r_t = \ln\left({}^{P_t}/_{P_{t-m}}\right)$$

where P_t is the price of the stock at time t. These log-returns are calculated from the stocks' Open price, High price, Low price, Close price, and Total Return Index (TRI). A complete list of these features is given in Table I where weekly log-returns are primarily used (m = 5).

TABLE I
LOG-RETURN FEATURES.

Feat.	P_t	P_{t-m}	m	Feat.	P_t	P_{t-m}	m
1	Close	Close	5	10	Open	Low	5
2	Close	Close	4	11	Low	Low	5
3	Close	Close	3	12	High	High	5
4	Close	Close	2	13	High	Open	5
5	Close	Close	1	14	High	Low	5
6	Close	Open	5	15	High	Close	5
7	Close	High	5	16	TRI	TRI	5
8	Close	Low	5	17	TP	TP	5
9	Open	Open	5				

The following momentum-based moving average technical indicators are also used:

1) Simple Moving Averages (SMA) and Exponentially-weighted Moving Averages (EWMA): smooth the time series by filtering out the noise in short term price fluctuations [24]. These indicators are calculated over 5, 10, 20, 50, and 200 day periods using Feature 1 (see Table I).

2) Relative Strength Index (RSI): measures the magnitude of fluctuations in the price of a stock [25]. This can be used to evaluate whether stocks are over or under traded which is useful in determining the stock directional movements. It is calculated over a 14-day period using the stocks' Close price.

3) Moving Average Convergence Divergence (MACD): shows the relationship between two EWMAs that indicate buy or sell signals used to determine stock directional movements [25]. Calculated using Feature 1 by subtracting a 12-day EWMA from a 26-day EWMA.

4) Stochastic Oscillator (SO): measures the degree of change between stock returns to determine future trend shifts [25]. It is calculated over 14-days using Feature 1.

5) Commodity Channel Index (CCI): determines the trend strength and direction of a stock to identify cyclical trends and if a stock is approaching under or over-traded conditions [25]. It is calculated over a 20-days using the Typical Price (TP) of a stock which is the average of the stocks' Open, Low, and High prices.

B. Data Preparation

The dataset is scaled to the range of $[-1, 1]$ due to the hyperbolic tangent activation function used in the LSTM and GRU cells [10, 26]. A rolling window forecasting approach [12, 16, 17] is used to train and test the models by splitting the dataset into multiple overlapping training and validation

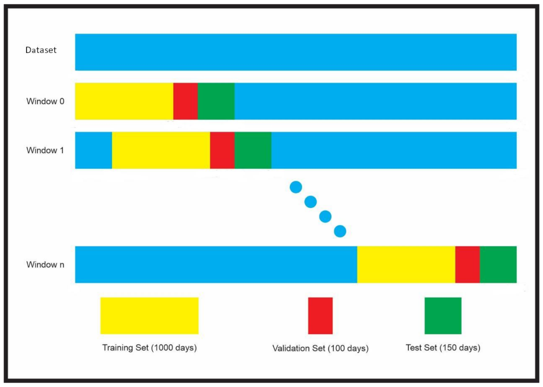


Fig. 1. An illustration of the rolling window forecasting approach.

TABLE II
LSTM AND GRU HYPER-PARAMETERS.

	LSTM					GRU				
Stock	**LR**	**DRR**	**Cells**	**SL**	**BS**	**LR**	**DRR**	**Cells**	**SL**	**BS**
ABG	0.0010	0.0061	256	119	44	0.0025	0.0158	166	128	83
AFE	0.0015	0.0000	256	200	240	0.0100	0.0000	46	189	214
AFX	0.0011	0.1004	204	100	1	0.0080	0.2284	42	113	68
AVI	0.0089	0.2154	248	185	14	0.0026	0.1801	217	187	33
BVT	0.0076	0.1842	192	136	249	0.0028	0.1728	209	117	247
FSR	0.0028	0.0598	256	106	20	0.0010	0.0582	256	107	1
LBH	0.0100	0.0000	245	164	136	0.0002	0.1403	132	199	1
NED	0.0052	0.1910	248	191	75	0.0042	0.0115	63	174	106
NTC	0.0030	0.1594	256	172	23	0.0001	0.1473	203	200	166
PIK	0.0100	0.1478	102	135	28	0.0100	0.0329	134	131	111
RLO	0.0026	0.1801	217	187	33	0.0048	0.1851	227	163	20
SBK	0.0032	0.0520	112	157	9	0.0032	0.1816	256	173	17
SHP	0.0048	0.1851	227	163	20	0.0077	0.142	89	193	198
TBS	0.0026	0.0993	256	100	80	0.0006	0.1731	170	195	179
WHL	0.0059	0.1437	159	200	1	0.0020	0.0797	247	141	82

sets, with non-overlapping testing sets to prevent data leakage. Specifically, the size of each window contains 1250 days, of which the first 1000 days form the training set, the next 100 days form the validation set, and the remaining 150 days form the testing set. The network is then rolled over by 150 days leading to a new window with a different testing set as shown in Fig. 1. With the dataset being used, this results in 26 of these windows in which the models are re-trained on each window.

C. Assumptions

The assumptions made in conducting the research are: *1)* stocks are infinitely divisible, *2)* stocks are highly liquid and can be bought or sold at closing prices, *3)* dividends are excluded and *4)* there are constant transaction costs of 30bps, $c = 0.003$.

D. Forecasting

The models compared are LSTM and GRU. These are parameterised independently for each stock resulting in 15 LSTM and 15 GRU models, each with unique hyper-parameters. The objective is to forecast a stocks' weekly log-returns:

$$y_{t+5|t} = \ln\left(\text{Close}_{t+5}/\text{Close}_{t}\right)$$

with the information available up until time t.

Models were implemented using Keras [27]. The network architectures are comprised of two layers consisting of the same number of LSTM/GRU cells followed by a fully-connected output layer consisting of one output cell with a linear activation function. The AMSgrad stochastic optimiser [28] is used in training the networks [17]. As the rolling window forecasting approach is followed, the training set is relatively small for data-hungry DL models. To prevent overfitting, both early stopping with a patience of 20 epochs and a small dropout rate are used [29].

Bayesian hyper-parameter optimisation [30] is employed using the first 1250 data points of the dataset. This corresponds to the first window of the dataset, and thus the testing of the model starts from the second window. The selected hyper-parameters are given in Table II and were sampled from a Learning Rate (LR) between 0.0001 and 0.01, Dropout Regularisation Rate (DRR) between 0 and 0.25, LSTM/GRU Cells between 5 and 256, Sequence Length (SL) between 100 and 200, and Batch Size (BS) between 1 and 250.

The Autoregressive–moving-average (ARMA) model is used as a baseline forecasting model [31]. The Naive and Drift models are used as they perform well on economic and financial time series [32]: Naive forecasts the last known true observation and drift allows the last known true observation value to drift by the average past change seen.

E. Portfolio Management

Mean-variance portfolios are constructed from the forecasts of the four models. The forecasts range from May 2005 to December 2019 across all the testing sets. The initial portfolio is constructed on 9 May 2005 and rebalanced every subsequent Monday until 23 December 2019 resulting in 764 weekly trading periods where t will refer to the current trading period and $t + 1$ to the one a week later.

The forecasted weekly log-returns ($\hat{y}$) are converted to percentage returns $\tilde{y} = e^{\hat{y}} - 1$. The percentage returns of each stock are placed into a vector μ_t of expected (forecast) returns and $\mathbf{x_t}$ denotes the true percentage returns during the trading period t.

The portfolio weights are determined at the beginning of week t with the objective of maximising returns until week $t + 1$. The covariance matrix of expected returns is estimated from the past 51 weekly percentage returns ($\mathbf{x_{t\text{-}51\ldots t\text{-}1}}$) and the forecast percentage returns (μ_t) using the Ledoit-Wolf shrinkage estimator [22, 26]. The forecasted returns and covariance matrix are combined in the MVO (see Equation 1) to determine the portfolio weights at the beginning of week t. The portfolio is rebalanced weekly but weights change daily due to fluctuations in the price of stocks. This is known as Portfolio Drift:

$$w_{t,e} = w_{t,b}(1 + \mathbf{x_t}) \tag{3}$$

$$w_{t,e} = \frac{w_{t,e}}{w_{t,e}^{T}\mathbf{1}} \tag{4}$$

where $w_{t,b}$ are the portfolio weights at the beginning of the trading period t, and $w_{t,e}$ are the portfolio weights at the end of the trading period t. Equation 3 calculates the Portfolio Drift and Equation 4 normalises the portfolio weights so that they sum to 1. The portfolios' gross returns ($r_{gross,t}$), net returns ($r_{net,t}$), and risk ($\sigma_{p,t}$) for the trading period t are then calculated as follows:

$$
\begin{aligned}
r_{gross,t} &= w_{t,e}^T \mathbf{x_t} \\
TC_t &= c \sum_{i=1}^{n} |w_{t,b} - w_{t-1,e}| \\
r_{net,t} &= (1 + r_{gross,t})(1 - TC_t) - 1 \\
\sigma_{p,t} &= \sqrt{w_{t,e}^T \Sigma_t w_{t,e}}
\end{aligned}
$$

where TC_t are the transaction costs incurred which is a proportion of the absolute change in weights from before and after rebalancing the portfolio at the start of trading period t [33].

To extend MVO from a single-period to a multi-period setting, a weight change penalty with the parameter λ and a maximum weight bound constraint is added to the L_2-constrained MVO objective function to account for transaction costs:

$$
\begin{aligned}
&\max_{w} \; w^{\mathrm{T}}\mu - \frac{\delta}{2} w^{\mathrm{T}}\Sigma w - \gamma||w||^2 - \lambda c \sum_{i=1}^{n} |w_i - w_{prev,i}| \\
&\text{s.t.} \quad 0 \leq w_i \leq 0.25, \forall i \in \{0, n\}; \qquad w^{\mathrm{T}}\mathbf{1} = 1
\end{aligned} \tag{5}
$$

where w_{prev} are the portfolio weights before rebalancing. The parameter λ indicates the magnitude of the weight change penalty. If $\lambda = 1$, then the weight change penalty is equivalent to transaction costs. Increasing λ to be greater than 1 essentially overestimates transaction costs and further discourages large weight changes when rebalancing the portfolio. This also helps in reducing the sensitivity that the optimiser has towards estimation errors along with the maximum weight bound constraint.

From the forecasts of the four models, the L_2 constrained MVO (Equation 2) and adjusted MVO that accounts for transaction costs (Equation 5) are used to construct and rebalance mean-variance efficient portfolios. Doing this compares the two optimisation objectives and simultaneously evaluates the quality of forecasts of the models. The parameters used in the MVOs are: $\delta = 1$, $\gamma = 0.01$ and $\lambda = 5$.

The Equally Weighted Portfolio (EWP) is used as a benchmark portfolio for comparison to the mean-variance portfolios. The EWP assigns the same weight $1/n$ to each stock in the portfolio for each trading period [33]. A buy-and-hold (BAH) portfolio, where a weight $1/n$ is assigned to each stock in the initial trading period and not rebalanced over subsequent trading periods, is also used for comparison.

V. Experimental Results and Discussion

A. Forecasting Evaluation Metrics

The following metrics are used to evaluate the performance of the models [32]: *1)* Mean Squared Error (MSE), *2)* Mean Absolute Error (MAE) and *3)* Out-of-Sample Coefficient of Determination (R^2).

B. Portfolio Evaluation Measures

The measures used to evaluate portfolio performance, where greater values indicate better performance, are [24]:

1) Cumulative Returns: $r_{cum} = \prod_t (1 + r_t)$ that the portfolio has incurred over the entire investment period assuming profits are compounded over each trading period [13];

2) Sharpe Ratio: $\mathrm{SR}_i = {(\bar{r_i} - \bar{r_f})}/{\sigma_{ER}}$ or risk premium return per unit of unsystematic risk incurred to achieve the excess returns over a risk free investment [34]; and

3) Information Ratio: $\mathrm{IR}_i = {(\bar{r_i} - \bar{r_b})}/{\sigma_{ER}}$ or excess return over the benchmark portfolio per unit of unsystematic risk incurred to achieve the excess returns over the benchmark portfolio [35]. Where r_t is the return of the portfolio during the trading period t, $\bar{r_i}$ is the average rate of return of portfolio i, $\bar{r_f}$ is the average rate of return on a risk-free investment known as the risk-free rate (RFR), $\bar{r_b}$ is the average rate of return of the benchmark portfolio, and σ_{ER} is the standard deviation of excess returns.

SR and IR can be annualised by multiplying them by $\sqrt{T}$ where T is the number of trading periods in a year (52 weekly) [24].

As the RFR is required to calculate the SR, the 91-day treasury bill rates are obtained from the South African reserve bank and used as a proxy for the RFR. The T-bills are quoted at an annualised discount rate (d) and need to be converted to a weekly interest rate. The quarterly value:

$$V = 100\left(1 - d({91}/{365})\right) \tag{6}$$

of the T-bill is first calculated and then the weekly interest/risk-free rate is calculated as:

$$r_f = \left({100}/{V}\right)^{\frac{7}{91}} - 1 \tag{7}$$

C. Results

The results are shown in Table III where bold entries indicate which model performed best on a particular stock. For individual stocks, performance is similar between the LSTM, GRU, and ARMA models, whilst performance between the Naive and Drift models are comparable. There is little variability between the MSE and MAE of the five models indicating that similar results can be achieved on stocks not included in this study's investment universe. The LSTM only underperforms on the LBH stock achieving a lower MSE than the Naive method. This could be attributed to hyper-parameters of the LSTM model used on the LBH stock.

The R^2 of all the models are negative indicating that the models aren't able to learn the trend of the data. This is expected due to the complexity and noise of financial time series data. The LSTM, GRU, and ARMA's R^2 values are closer to 0 meaning the forecasts of these models are "safer" and closer to the mean of the data. However, when forecasting, performance should primarily be determined from accuracy metrics such as the MSE and MAE rather than explanatory metrics such as the R^2 [17].

Overall, the average forecasting accuracy based on the MSE and MAE of the LSTM, GRU, and ARMA is on par. These

TABLE III
FORECASTING RESULTS.

Stock	LSTM MSE	LSTM MAE	LSTM R^2	GRU MSE	GRU MAE	GRU R^2	ARMA MSE	ARMA MAE	ARMA R^2	Naive MSE	Naive MAE	Naive R^2	Drift MSE	Drift MAE	Drift R^2
ABG	0.0018	0.0321	-0.2146	0.0016	0.0301	-0.0781	**0.0016**	**0.0295**	**-0.0837**	0.0034	0.0435	-1.3481	0.0041	0.0473	-1.7933
AFE	0.0015	0.0278	-0.5562	0.0011	0.0249	-0.1607	**0.0010**	**0.0242**	**-0.0623**	0.0020	0.0335	-1.0728	0.0024	0.0364	-1.4837
AFX	0.0015	0.0283	-0.1409	0.0016	0.0288	-0.1516	**0.0014**	**0.0273**	**-0.0282**	0.0027	0.0390	-1.0352	0.0034	0.0429	-1.5301
AVI	**0.0013**	**0.0258**	**-0.0467**	0.0013	0.0267	-0.0974	0.0013	0.0262	-0.0797	0.0028	0.0378	-1.2639	0.0033	0.0415	-1.7135
BVT	**0.0013**	**0.0267**	**-0.0623**	0.0013	0.0273	-0.1032	0.0013	0.0269	-0.0493	0.0025	0.0383	-1.1193	0.0031	0.0421	-1.5910
FSR	0.0018	0.0320	-0.1260	0.0020	0.0333	-0.2507	**0.0017**	**0.0310**	**-0.0705**	0.0037	0.0452	-1.2702	0.0044	0.0492	-1.7479
LBH	0.0025	0.0301	-1.2646	0.0012	0.0260	-0.0946	**0.0011**	**0.0252**	**-0.0579**	0.0024	0.0365	-1.2003	0.0030	0.0403	-1.7711
NED	0.0017	0.0306	-0.1975	0.0016	0.0304	-0.1486	**0.0015**	**0.0292**	**-0.0781**	0.0031	0.0422	-1.2326	0.0037	0.0459	-1.7004
NTC	0.0015	0.0283	-0.1269	0.0014	0.0281	-0.0956	**0.0014**	**0.0273**	**-0.0351**	0.0027	0.0395	-1.1016	0.0033	0.0432	-1.5805
PIK	**0.0013**	**0.0275**	**-0.0747**	0.0013	0.0275	-0.0987	0.0013	0.0278	0.0269	0.0026	0.0390	-1.1557	0.0031	0.0427	-1.6141
RLO	0.0014	0.0277	-0.0954	0.0015	0.0291	-0.1752	**0.0013**	**0.0273**	**-0.0834**	0.0028	0.0392	-1.2498	0.0034	0.0429	-1.7421
SBK	**0.0015**	**0.0291**	**-0.0890**	0.0016	0.0299	-0.1184	0.0015	0.0293	-0.0728	0.0033	0.0430	-1.3549	0.0040	0.0467	-1.8464
SHP	0.0016	0.0305	-0.1293	0.0015	0.0293	-0.0745	**0.0014**	**0.0288**	**-0.0416**	0.0029	0.0415	-1.1040	0.0036	0.0455	-1.5984
TBS	0.0014	0.0281	-0.1961	**0.0012**	**0.0264**	**-0.0373**	0.0012	0.0266	-0.0485	0.0025	0.0377	-1.1129	0.0031	0.0416	-1.6661
WHL	0.0017	0.0309	-0.1121	0.0018	0.0322	-0.1677	**0.0016**	**0.0304**	**-0.0542**	0.0032	0.0432	-1.0946	0.0039	0.0472	-1.5624
Average	0.0016	0.0290	-0.2288	0.0015	0.0287	-0.1235	0.0014	0.0278	-0.0581	0.0028	0.0399	-1.1811	0.0035	0.0437	-1.6627
Variance	$8 \cdot 10^{-8}$	$3 \cdot 10^{-6}$	0.0904	$5 \cdot 10^{-8}$	$5 \cdot 10^{-6}$	0.0026	$3 \cdot 10^{-8}$	$7 \cdot 10^{-6}$	0.0004	$2 \cdot 10^{-7}$	$9 \cdot 10^{-6}$	0.0095	$2 \cdot 10^{-7}$	$1 \cdot 10^{-5}$	0.0105

TABLE IV
COMPARISON OF PORTFOLIO PERFORMANCE WHEN IGNORING AND ACCOUNTING FOR TRANSACTION COSTS.

Portfolio	Ignoring Transaction Costs: Gross Performance r_{cum}	SR	IR	Ignoring Transaction Costs: Net Performance r_{cum}	SR	IR	Accounting for Transaction Costs: Gross Performance r_{cum}	SR	IR	Accounting for Transaction Costs: Net Performance r_{cum}	SR	IR
LSTM	28.7326	0.7641	0.6485	1.5806	-0.0375	-0.5910	15.3939	0.6670	0.6066	**10.1348**	**0.5281**	**0.3025**
GRU	23.7368	0.6998	0.5445	1.2357	-0.0983	-0.6539	14.3014	0.6518	0.5631	9.5892	0.5161	0.2586
ARMA	0.9851	-0.2415	-1.2104	0.0346	-1.3442	-3.1954	8.6321	0.5177	0.0871	8.3380	0.5044	0.1037
Naive	0.0469	-0.9221	-1.6409	0.0007	-2.0151	-3.0620	0.3449	-0.6374	-2.3920	0.0409	-1.3855	-3.9577
Drift	0.0644	-0.8377	-1.5212	0.0001	-1.9300	-2.9319	0.3320	-0.6442	-2.3727	0.0340	-1.4391	-4.0381

three models achieve roughly double the accuracy of the Drift and Naive methods. The Naive method serves as a good indicator of which stocks have less variability in their returns making them somewhat more predictable.

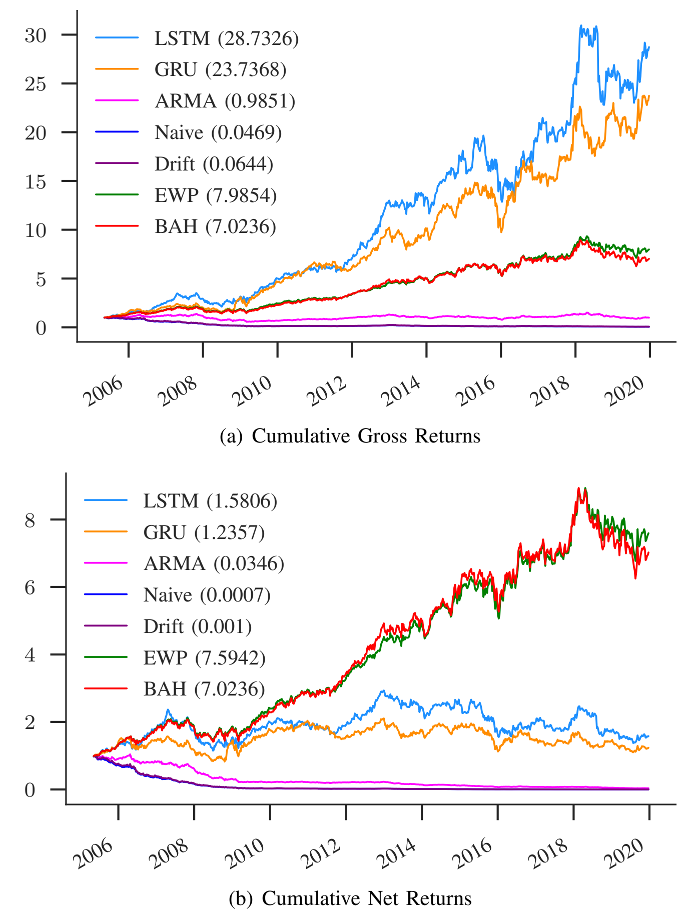
The evaluation measures of the portfolios constructed from the forecasts are given in Table IV when transaction costs are ignored versus when they are accounted for. SR and IR are annualised and the bold entry indicates the best performing portfolio. A visualisation of the cumulative return plots over the testing period for the portfolios are shown in Fig. 2 (ignore transaction costs) and Fig. 3 (transaction costs accounted for).

These plots show that the LSTM forecasts are of greater quality than the GRU forecasts, and the GRU forecasts than the ARMA forecasts when used to construct portfolios. The LSTM and GRU forecasts produce better portfolios because their forecasts are less "safe" than the ARMA forecasts. The majority of the ARMA forecasts lie in the range $[-0.01, 0.01]$ whereas the LSTM and GRU have a larger range giving a better indication of which stocks would be profitable when selecting the portfolio. This implies that the LSTM and GRU have a higher error when their forecasts are incorrect explaining why ARMA has marginally better forecasting results.

It is apparent that the forecasting errors of the Naive and Drift portfolio are too great to construct profitable portfolios irrespective of transaction costs. Although, when transaction costs are accounted for, the performance of these portfolios do improve confirming that the MVO that accounts for transaction costs is less sensitive to estimation errors. This can also be seen with the contrasting performance of the ARMA portfolios. When transaction costs are ignored, undiversified portfolios are constructed with a small number of stocks selected resulting in poor performance. As the MVO that accounts for transaction costs has additional constraints that encourage diversification, the resulting portfolio improves significantly outperforming the benchmark portfolios.

When transaction costs are ignored in the MVO, the gross portfolio performance indicates that the LSTM Portfolio outperforms the GRU portfolio achieving an additional 4.9958 cumulative gross returns as well as having a higher SR and IR. Both the LSTM and GRU portfolios have an SR and IR greater than 0 indicating that these portfolios are a better investment than investing in a risk-free investment and the benchmark EWP respectively. However, after taking transaction costs into account, it can be seen that most of the returns generated by these portfolios are lost, with SR and IR becoming negative, indicating their poor performance.

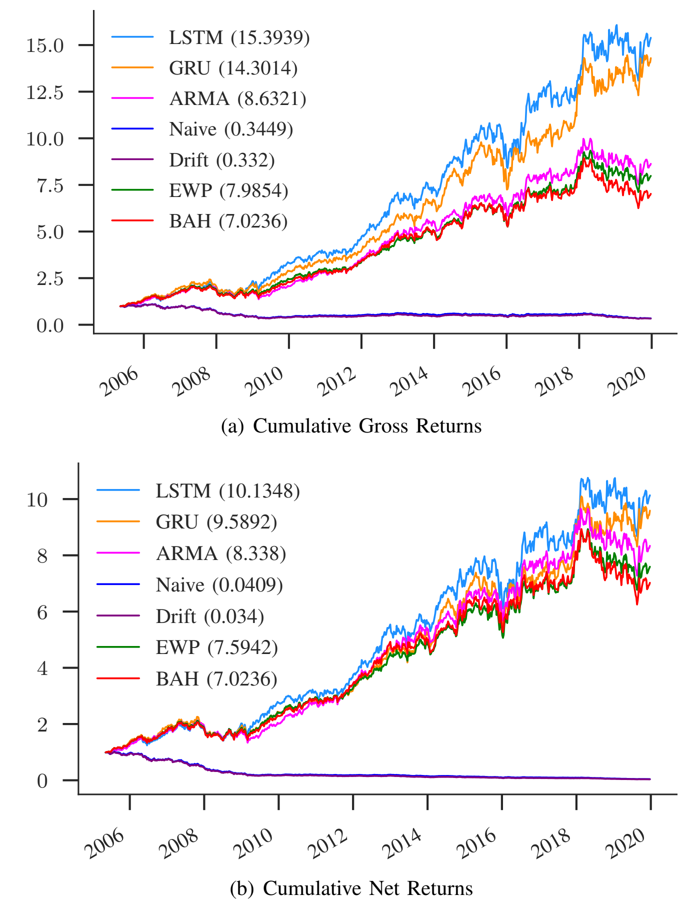
The gross portfolio performance when transaction costs are accounted for is worse than the gross portfolio performance when transaction costs are ignored, but the net portfolio performance is what matters when evaluating portfolios. The cumulative net return of the LSTM and GRU portfolios decrease by 5.2591 and 4.7122 respectively when transaction costs are accounted for compared to 27.1520 and 22.5011 when transaction costs are ignored. The SR and IR decrease but remain positive. This shows the significant improvements when transaction costs are accounted for in a multi-period



(a) Cumulative Gross Returns

(b) Cumulative Net Returns

Fig. 2. Portfolio cumulative returns when ignoring transaction costs.



(a) Cumulative Gross Returns

(b) Cumulative Net Returns

Fig. 3. Portfolio cumulative returns when accounting for transaction costs.

investment setting.

Reilly and Brown [24], state that an SR of at least 1.0 and an IR of at least 0.5 classify the portfolio as a "good" portfolio relative to the amount of risk incurred. Observing the net performance of the LSTM and GRU portfolios when accounting for transaction costs, the SR and IR indicate that the portfolios are sub-optimal as the proportion of portfolio risk is not proportional to returns gained by the portfolio. Even though these portfolios are sub-optimal, they are shown to be profitable and outperform the EWP and BAH portfolios.

VI. Conclusion

This research provides a portfolio management model to forecast weekly stock returns and make use of these forecasts to optimally construct and rebalance portfolios using an extension of the MVO framework to multiple trading periods.

An LSTM and GRU are applied to financial time series data to forecast the weekly returns of 15 stocks listed on the JSE using a rolling window forecasting approach. Using these forecasts, portfolios are constructed and rebalanced over multiple periods using two MVOs: a L_2-constrained MVO that does not account for transaction costs and a variation of the L_2-constrained MVO that has an additional term in the objective function that penalises large weight changes thereby accounting for transaction costs and reducing the sensitivity of the MVO towards estimation errors.

The forecasting accuracy of the LSTM, GRU, and ARMA models, are approximately equal. However, the LSTM produces higher-quality forecasts as the LSTM portfolio outperforms its counterparts in both settings. The MVO that accounts for transaction costs provides significant performance improvements when considering the net performance of the portfolios and can outperform the benchmark EWP and BAH strategies. Even though these portfolios are successful and profitable, they are sub-optimal due to the amount of risk incurred for the returns achieved.

A. Challenges and Future Directions

A limitation of this model and many of the existing models is that the dataset used is from a specific stock market. Since stock markets have different underlying characteristics and responses to macroeconomic variables, some models may be unable to achieve the same or similar results on a different stock market. This also makes it challenging to compare different models. The use of cross-learning can be effective in constructing models that can achieve consistent results across different markets [17, 18].

To further improve financial time series forecasts, model inputs can be expanded to include qualitative data such as fundamental features or features extracted from text mining. Incorporating financial event predictions, such as crises and earnings surprises, into models can lead to the development of trading strategies that capitalise on these events to realise superior profits. Similarly, capitalising on market anomalies such as calendar effects [19, 33] can help increase profitability.

ACKNOWLEDGMENT

This research is supported by the Nedbank Research Chair.

REFERENCES

[1] S. Sooklal, T. L. van Zyl, and A. Paskaramoorthy, "Dricorn-k: A dynamic risk correlation-driven non-parametric algorithm for online portfolio selection," in *Southern African Conference for Artificial Intelligence Research*. Springer, Cham, 2020, pp. 183–196.

[2] A. B. Paskaramoorthy, T. J. Gebbie, and T. L. van Zyl, "A framework for online investment decisions," *Investment Analysts Journal*, vol. 49, no. 3, pp. 215–231, 2020.

[3] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[4] R. O. Michaud, "The markowitz optimization enigma: Is 'optimized'optimal?" *Financial analysts journal*, vol. 45, no. 1, pp. 31–42, 1989.

[5] T. Mathonsi and T. L. van Zyl, "Prediction interval construction for multivariate point forecasts using deep learning," in *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE, 2020, pp. 88–95.

[6] B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[7] B. G. Malkiel, "Reflections on the efficient market hypothesis: 30 years later," *Financial review*, vol. 40, no. 1, pp. 1–9, 2005.

[8] R. Ball, "The global financial crisis and the efficient market hypothesis: what have we learned?" *Journal of Applied Corporate Finance*, vol. 21, no. 4, pp. 8–16, 2009.

[9] A. Timmermann and C. W. Granger, "Efficient market hypothesis and forecasting," *International Journal of forecasting*, vol. 20, no. 1, pp. 15–27, 2004.

[10] M. A. Hossain, R. Karim, R. Thulasiram, N. D. Bruce, and Y. Wang, "Hybrid deep learning model for stock price prediction," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1837–1844.

[11] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020.

[12] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[13] S. I. Lee and S. J. Yoo, "Threshold-based portfolio: the role of the threshold and its applications," *The Journal of Supercomputing*, pp. 1–18, 2018.

[14] Y. Wu and J. Gao, "Adaboost-based long short-term memory ensemble learning approach for financial time series forecasting." *Current Science (00113891)*, vol. 115, no. 1, 2018.

[15] H.-F. Hsiao, J.-C. Huang, and Z.-W. Lin, "Portfolio construction using bootstrapping neural networks: evidence from global stock market," *Review of Derivatives Research*, pp. 1–21, 2019.

[16] B. Zhou, "Deep learning and the cross-section of stock returns: Neural networks combining price and fundamental information," *Available at SSRN 3179281*, 2019.

[17] W. Wang, W. Li, N. Zhang, and K. Liu, "Portfolio formation with preselection using deep learning from long-term financial data," *Expert Systems with Applications*, vol. 143, p. 113042, 2020.

[18] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[19] L. Takeuchi and Y.-Y. A. Lee, "Applying deep learning to enhance momentum trading strategies in stocks," in *Technical Report*. Stanford University, 2013.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[21] P. Jorion, "Bayes-stein estimation for portfolio analysis," *Journal of Financial and Quantitative analysis*, pp. 279–292, 1986.

[22] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.

[23] B. Bruder, N. Gaussel, J.-C. Richard, and T. Roncalli, "Regularization of portfolio allocation," *Available at SSRN 2767358*, 2013.

[24] F. K. Reilly and K. C. Brown, *Investment analysis and portfolio management*. Cengage Learning, 2011.

[25] J. J. Murphy, *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[28] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012, vol. 37.

[31] P. Whittle, *Hypothesis testing in time series analysis*. Almqvist & Wiksells boktr., 1951, vol. 4.

[32] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[33] B. Li and S. C. Hoi, "Online portfolio selection: A survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–36, 2014.

[34] W. F. Sharpe, "The sharpe ratio," *Journal of portfolio management*, vol. 21, no. 1, pp. 49–58, 1994.

[35] T. H. Goodwin, "The information ratio," *Financial Analysts Journal*, vol. 54, no. 4, pp. 34–43, 1998.