Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project – ACSE 9

Final Report

Financial Security Price Prediction with Deep Learning

By Jack Quested

jack.quested20@imperial.ac.uk

GitHub ID: acse-jaq15
GitHub Repo: https://github.com/acse-2020/acse2020-acse9-finalreport-acse-jaq15

Supervisor: Dr. Cedric John

Co-Supervisor: Prof. Stephen Neethling

June 2021

Abstract
A comparative study between Deep Learning architectures and their ability to perform univariate time series forecasting was undertaken. Financial time series data from 2014 - 2019 was used, with the period 2014 - 2018 forming the training dataset and 2019 being the test dataset. Models were tasked with predicting the next day's closing price based upon the preceding 30 days of closing prices. 12 models were used, including LSTM, GRU, CNN and MLP based architectures. 12 securities were examined, drawing from each of the major financial asset classes: commodities, currencies, equities and fixed income. Each model was used on each security and underwent hyperparameter tuning. Model predictions were assessed via error metrics: MSE, RMSE and MAE. The loss function used by all models was the MSE of the prediction. A dummy model was used as a control against which predictions were also assessed. The dummy model returned the arithmetic mean of the 30 day window as its prediction. A points system was used to rank all models according to the MSE of their predictions. The same system was used to rank all securities based on how difficult models found them to predict. All models performed better than the dummy model, with GRU based models performing the best. LSTM based models performed marginally worse than GRUs, while CNN based architectures were the worst performers. GRU and LSTM based models performed substantially better than all others. Hyperparameter tuning results across models suggested that SELU was the most appropriate activation function, followed by tanh. Similarly, the most regularly used optimisers were Adam and nAdam, while SGD and RMSprop resulted in notably worse performance by comparison. Fixed income securities generated the lowest mean squared errors, whilst commodities produced the highest. Inverse correlation between the volatility of a security and the models ability to predict accurately was found. The number of trainable parameters was found to have minimal influence over the accuracy of a models predictions.

Key words: deep learning, financial time series, univariate time series analysis, price prediction, GRU, LSTM

Acknowledgements
I would like to thank my supervisor Dr Cedric John for his helpful guidance with this project. Thanks are also owed to Marijan Beg for his input and help as leader of the study group and to the ACSE faculty for delivering the course in a highly unusual and difficult environment. Gratitude is extended to my employer, the UK Debt Management Office, for allowing me to take a sabbatical in order to undertake this degree. Finally, I would like to thank my friends and family for their support in a year of challenges.

Introduction

For as long as there have been financial securities, market participants have sought methods to forecast prices (Galbraith, 1955). A major riposte to this practice came in 1970 with the publication of the Efficient Market Hypothesis (EMH) (Malkiel & Fama, 1970). The EMH posited that stocks and, by wider extension, financial securities (Malkiel, 1973), follow a random walk, making consistently accurate forecasting virtually impossible.

Advances in computational hardware, and the adoption of mathematical models, led some investors from the late 1980's onwards to achieve success (Ziemba, 2020; Harding, 1994). These quantitatively focused methods of investment were unorthodox (Malkiel, 1973) and would only gain wider acceptance from the mid 2000's (Lewis, 2014; Lo, 2017). David Harding and Jim Simons are early notable examples of investors adopting such an approach in a successful manner (Harding, 1994; Ziemba, 2020).

The advent of Deep Learning (DL), and the advances it has wrought, now means that an active area of research is in financial time series prediction using DL (Sezer, Gudelek & Ozbayoglu, 2020).

Financial time series are known to be non-linear, and as such are difficult to predict (Rezaei, Faaljou & Mansourfar, 2021; Jayanth, Harish Ram & Nair, 2018). However, this has not deterred researchers: numerous statistical, Machine Learning (ML) and DL based forecasting approaches exist (Sezer, Gudelek & Ozbayoglu, 2020).

Some researchers approached the problem by studying only ML approaches (Bontempi, Taieb & le Borgne, 2013), while others adopted a purely DL based method (Sezer, Gudelek & Ozbayoglu, 2020).

It has been found that some ML based approaches perform no better, or marginally worse, than statistical implementations (Cao, Leggio & Schniederjans, 2005). Conversely, DL is at the forefront of current research, with DL implementations now outperforming ML and statistical approaches (Sezer, Gudelek & Ozbayoglu, 2020; Hiransha *et al.*, 2018).

The computerisation of financial markets has led to a dearth of data being available. This, combined with the development of automated trading systems, has led to a surge in the number of research articles focusing on DL and financial time series analysis, as noted in Fig. 1.

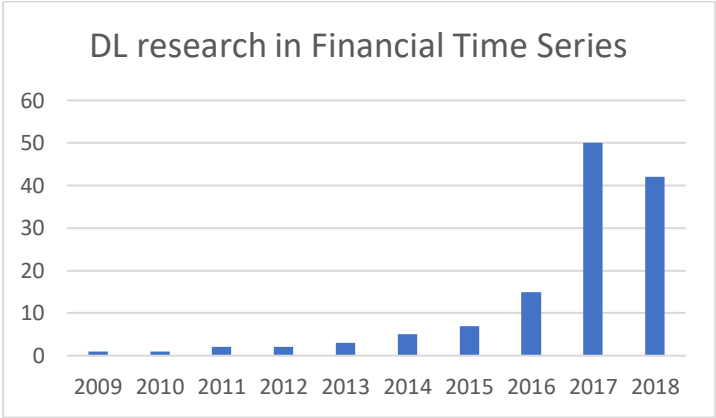


Figure 1. Histogram of the number of published papers on DL applications in financial time series analysis since 2009 (Sezer, Gudelek & Ozbayoglu, 2020)

What precisely makes DL perform so well is an area of active research and debate. In general terms, it is understood that DL makes use of the unknown structure of the input distribution in

order to discover representations of the data (Bengio, 2012) - though it is noted that DL architectures are prone to overfitting and computational complexity. Various regularisation mechanisms exist to combat the former, whilst newer, more efficient, architectures can combat the latter (Bengio, Boulanger-Lewandowski & Pascanu, 2013; Dahl, Sainath & Hinton, 2013).

Early advances in DL came from the use of a Deep Multi-Layer Perceptron (MLP) (Goodfellow, Bengio & Courville, 2016), where the network has fully connected input, hidden and output layers, with each neuron using a non-linear activation function (Goodfellow, Bengio & Courville, 2016). MLP networks have been successfully used in classification and regression type problems (Gardner & Dorling, 1998).

Though MLPs represented a significant step forward, they were not without issues. The use of fully connected layers causes computational complexity considerations, while MLP networks are also susceptible to vanishing gradients (Sezer, Gudelek & Ozbayoglu, 2020; Gardner & Dorling, 1998).

Recurrent neural networks (RNN) are another approach that use internal memory to interpret inputs, thus making them well suited to sequenced data such as time series (Deng & Yu, 2014). RNNs themselves make use of backwards temporal dependence (Rumelhart, Hinton & Williams, 1986). However, when learned knowledge is stored for too long, continuing to train the network becomes difficult (Pascanu, Mikolov & Bengio, 2013).

Long Short-Term Memory (LSTM) networks are a subtype of RNNs that have come to dominate the field of financial time series analysis (Sezer, Gudelek & Ozbayoglu, 2020). LSTMs were first introduced in 1997 (Hochreiter & Schmidhuber, 1997) and were subsequently refined in 1999 with the addition of forget gates (Gers, Schmidhuber & Cummins, 1999). The use of an LSTM can go some way towards ameliorating the drawbacks of RNNs (Sezer, Gudelek & Ozbayoglu, 2020).

A Gated Recurrent Unit (GRU) is a network type that has been found to be performant in handling time series (Chung *et al.*, 2014). Similar to an LSTM, it lacks an output gate and may represent an efficient choice in time series analysis.

Convolutional Neural Networks (CNN) are a type of DL network commonly used in computer vision applications. They have also been used effectively in analysing financial time series where the use of kernel filters allows for computationally efficient implementations (Sezer, Gudelek & Ozbayoglu, 2020).

Autoencoders (AE) have been demonstrated to be of use in time series analysis (Sezer, Gudelek & Ozbayoglu, 2020). AEs effectively remap the features of the input data in such a way that the remapped data are representative of structures that can be used for classification (Hinton & Salakhutdinov, 2006). It is noted that AE feature extraction can struggle with highly dimensional data (Meng *et al.*, 2017).

There has also been an increasing volume of research into the specific topic of DL applications within financial time series analysis. This can generally be divided into 2 main areas of research: price prediction and trend prediction. The former attempts to forecast the actual price within a given window (and the focus of this project) and the latter simply tries to predict whether the market will rise, fall or remain neutral (Sezer, Gudelek & Ozbayoglu, 2020).

The asset class most frequently examined in the literature is that of equities, followed by foreign exchange and commodities. Fixed income remains a field with a far smaller body of research (Sezer, Gudelek & Ozbayoglu, 2020).

The models mentioned in this discussion have all been selected for inclusion in this study. All the chosen models have a wide and established research base (Sezer, Gudelek & Ozbayoglu, 2020) and either have been, or are currently, state of the art. Thus, the model selection is representative of what is most widely accepted to be the forefront of research in DL based time series analysis. Much of the literature discussed looks at individual DL architectures without broad based comparisons to other network types. From this, much of the motivation to conduct to a comparative study stems.

See Appendix 1 for a table of the models implemented. For detailed graphic representation of the exact model architectures used, please refer to the model_graphs directory in the GitHub repo. A detailed explanation of the model architectures is beyond the remit of this comparative study. A large body of relevant literature exists in which architectural features of the models are discussed at length.

Securities used in this study were selected so as to adequately represent the 4 major financial asset classes; commodities, currencies, equities and fixed income (Graham, Dodd & Buffett, 2009). The securities themselves are liquid and have reliable historical price data which can be noted from Appendix 2.

The prices that form the historical data are daily closing prices, calculated at the end of a trading day, from the first trading day of 2014 till the final trading day of 2019. Each security has a differing number trading days in a year, depending on the trading calendar of the listing jurisdiction.

The methodologies for arriving at these closing prices can vary from security to security. For those securities listed on an exchange, the exchange operator publishes detailed information on the process. Some securities (currencies and US Treasuries) are not listed on an exchange and are traded in over the counter (OTC) markets. For currencies, the specific data vendor, in this case Bloomberg LP, has their own methodology for the calculation of closing prices. These are commonly based on London trading hours as it is the world centre of currency trading (Hull J, 2012). Exactly how the closing prices are determined is largely inconsequential to this study, but further details can be obtained via the documents published by said exchange operators or data vendors.

The loss function used in the optimisation of all models was the mean squared error (MSE) of predictions. Other error metrics used in addition to MSE were root mean squared error (RMSE) and mean absolute error (MAE) where:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|}{n}$$

With n being the number of data points and $Y_i - \hat{Y}_i$ being the difference between the true and predicted value.

Model	Activation Function	Batch Size	Epochs	Dropout	Input Layer Size	Hidden Layer Size	Learning Rate	Optimiser
CNN	SELU, RELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
CNN GRU	SELU, RELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop, SGD
CNN LSTM	SELU, RELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop, SGD
GRU	tanh, SELU, RELU	32, 64, 128	50, 100, 200	0.2, 0.3, 0.4	25, 50, 100	25, 50, 100	0.005, 0.001, 0.0005	Adam, nAdam, SGD
GRU AE	tanh, RELU	32, 64, 128	50, 100, 200	0.1, 0.2, 0.3	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
GRU LSTM	tanh, RELU, SELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
LSTM	tanh	32, 64, 128	50, 100, 200	0.2, 0.3, 0.4	25, 50, 100	25, 50, 100	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
LSTM AE	tanh, RELU	32, 64, 128	50, 100, 200	0.1, 0.2, 0.3	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
LSTM GRU	tanh, RELU, SELU	32, 64, 128	50, 100, 200	0.1, 0.2, 0.3	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop
MLP	RELU, SELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop, SGD
MLP AE	RELU, SELU	32, 64, 128	50, 100, 200	-	-	-	0.005, 0.001, 0.0005	Adam, nAdam, RMSprop, SGD

Table 1. Model hyperparameters included in tuning

A number of activation functions were included in hyperparameter tuning: RELU, SELU and tanh, where tanh is the hyperbolic tangent. RELU and SELU are expanded below:

$$RELU(x) = \max(0, x)$$ (Fukushima, 1980)

$$SELU(x) = \lambda \begin{cases} x, & x < 0 \\ \alpha e^x - \alpha, & x \geq 0 \end{cases}$$ (Klambauer *et al.*, 2017)

All models implemented used a degree of dropout, either 0.1, 0.2 or 0.3. For some models, this was included in the hyperparameter search, for others a dropout value of 0.1 was used with no tuning. This was done in order to limit the size of the hyperparameter tuning sweeps in order to facilitate completion within a feasible timeframe. Please refer to Table 1 for more details, where no entry is found, dropout was 0.1.

Software Description and Methodology

The study was performed by following a workflow, see Fig. 2, where each model was used to make predictions for each security. The workflow was performed in a Google Colaboratory notebook.

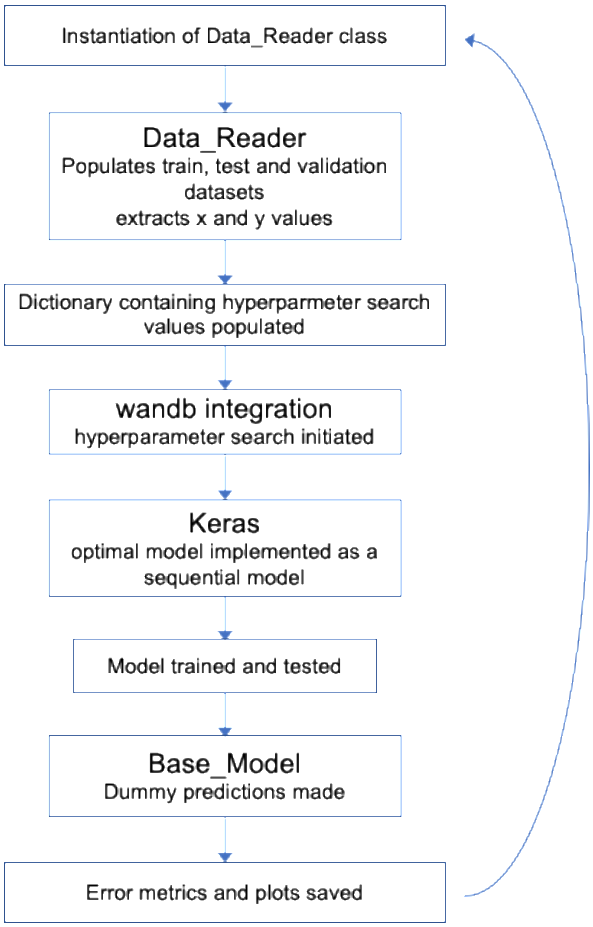


Figure 2. Model workflow

First the data was loaded via a call to the Data_Reader class, then separated into training, validation and test datasets. The training dataset forms the time period 2014 - 2018, with the test dataset consisting of the data from 2019. The validation dataset, used to perform hyperparameter tuning, spans the date range 2014 - 2018, with 2014 - 2017 being the validation training set and 2018 the validation test set.

Using the same Data_Reader class, X and y datasets were extracted. The X dataset is used by the model to makes predictions, which were evaluated against the extracted y dataset. This was performed for the training, testing and validation datasets. The X and y datasets are made up of normalised data. Data is normalised via (0,1) min/max feature scaling, where:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

With x' being the normalised value of original value x, where $x \in X$, with X being the entire unnormalised dataset.

Hyperparameter tuning was then conducted with Weights and Biases (wandb) integration, wandb is a service that performs hyperparameter tuning, among other ML related tasks. Table 1 shows the hyperparameters for which each model was tuned. Tuning was conducted on the validation dataset, making use of the respective validation training and test datasets.

Tuning took place using a random search methodology in order to discover potentially more powerful settings earlier than if conducting a grid search (Bergstra & Bengio, 2012). Hyperparameter tuning continued until certain stopping criteria were reached. Provided the loss was below a certain threshold, and had ceased to change by a certain degree, searching was terminated. Across all searches a total of 5,347 different model combinations were tested on the validation set. See Fig. 3 for an example visualisation of a wandb search run and how different parameters influence the loss function.

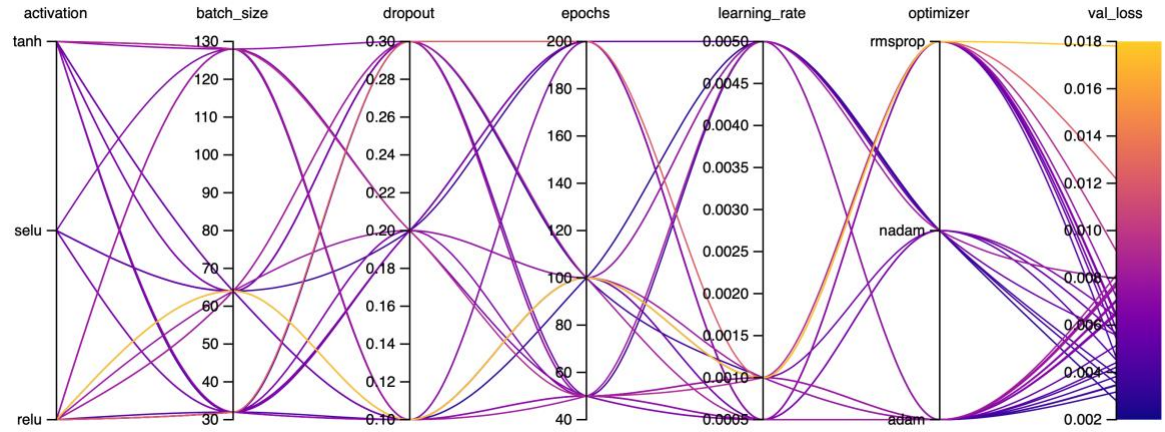


Figure 3. Visualisation of hyperparameter searching with wandb

Following the completion of hyperparameter tuning, optimal settings were then used to create a model that was trained for the number of epochs specified by the search. This model was then tested on the test dataset and assessed using the error metrics. A diagrammatic representation of the model and a plot of predicted and actual prices were then saved to memory.

The GPU typically used when connected to Google Colaboratory was an Nvidia Tesla P100 Server Graphics Card. Where model architectures permitted, this GPU allowed for Nvidia's CUDA Deep Neural Network parallel programming platform to be enabled, lowering hyperparameter tuning and model training times.

Code metadata

The language used in this study was Python 3.2.10, Python has seen widespread adoption among the DL community (Sezer, Gudelek & Ozbayoglu, 2020) and a wide variety of useful packages and languages are available. One such library is Keras which runs on top of

Google's well known TensorFlow. The Keras version used was 2.6.0 and was selected due to the ease of constructing DL architectures. This library contains many DL architectural units that run out of the box such as LSTM, GRU and CNN layers. Scikit-Learn (SK Learn) was used to perform normalisation of price data, the version used was 0.22.2.

Pandas 1.1.5 was used for handling data and storing model evaluation data. Under the hood data manipulation performed by multiple modules, discussed later, used NumPy version 1.19.5. Testing within the GitHub repo was done using PyTest version 3.6.4 and plotting handled by Matplotlib version 3.2.2. The operating system used was macOS version 11.4.

Implementation and Code Description

In order to facilitate the previously mentioned workflow, a number of modules were implemented, these included the data_reader, base_model and model_loader modules.

The data_reader module contains a Data_Reader class and was used to manipulate the time series data prior to being fed through a model. The class takes in two positional arguments: a string and an integer. The string defines which security will have its data loaded, the integer denotes what data from the time series will form the test dataset. For all use cases in this study, the integer argument was always 2019, meaning the test dataset comprised all closing price data from that year.

Instantiation of the Data_Reader class creates various other attributes, which are added to as additional class methods are called. These methods must be called in the correct order or else an error is returned. The first method to be called is .extract_train_test, which generates training and test datasets, both normalised and unnormalised. Unnormalised datasets are retained as class attributes, despite the additional memory overheads, so as to facilitate plotting of real prices once predictions have been made.

The Data_Reader class also generates X and y variables for all datasets via a call to .extract_xy method. This method takes in two positional arguments, the window length and a Boolean that determines the dimensions of the resulting X arrays. The window length variable tells the Data_Reader class the number of preceding days of closing prices upon which the single day ahead prediction will be based. For all use cases in this study the window length was 30, meaning the model based its predictions on the previous 30 days of closing price data.

Where the Boolean is set to False, the resulting X arrays have a number of columns equal to the window length, in this case 30, and a number of rows equalling the number of days in the dataset, minus the window length. When this Boolean is set to True, the X arrays have a modified shape. This allows for convolutional model architectures to process the data. Models that required this Boolean to be set to True were CNN GRU and CNN LSTM.

The base_model module contains the Base_Model class. This class acts as a control against which DL predictions can be judged. The Base_Model class takes the preceding day's closing prices and returns the mean of that period as its prediction. In some cases, partly due to overfitting, or use of models with inadequate depth, some ML and DL models were found to generate predictions no better than the mean (Cao, Leggio & Schniederjans, 2005). The use of this class as a control therefore allows a low hurdle against which the DL models can be compared.

The model_loader module contains Trained_Model and Untrained_Model classes. These classes are used for demonstration purposes and in the evaluation of the finalised models. The Trained_Model class can be used to load saved models. These saved models are hyperparameter optimised models that were used in this study to generate predictions. The

trained models have weights and biases generated from training as per the relevant Colaboratory notebook.

The Untrained_Model class loads untrained models that carry the default settings used prior to hyperparameter tuning. These models have randomly initialised weights and biases, done so according to the Keras documentation.

An additional consideration is the use of a separate GitHub repo from which these modules are loaded in the Colaboratory notebooks. A public repo named feeder_repo was used to allow for safe reproducibility of the notebooks by an independent user. A number of constraints and issues informed the decision to use this dual repo design. One such issue was the security concerns raised by attempting to access a private repo via the notebook. In order to do this, private SSH keys or GitHub usernames and passwords must be saved as variables within the notebook. These concerns were addressed by using the public feeder_repo to clone into the notebook environment. The feeder_repo was populated with the relevant modules and data by manually adding them from the private GitHub repo used for this study. Use of GitHub's repo mirroring was not possible since a private repo cannot be mirrored by a public one.

Results, Discussion and Conclusions

The models were judged using a ranking system based on the MSE of their predictions. For each security, the models were sorted in ascending order according to the MSE of their prediction. The 12 models were then awarded points according to their rank, with the lowest MSE model receiving a rank of 1st and 1 point respectively, whilst the model with the highest MSE was given a rank of 12th and 12 points. This scoring was applied to all model and security combinations and the total points used to give a final ranking. The model with the lowest number of points was ranked 1st, see Table 2.

Model	Points	Rank
GRU_LSTM	36	1
LSTM_GRU	38	2
GRU	41	3
GRU_AE	45	4
LSTM_AE	72	5
MLP	75	6
LSTM	76	7
CNN	77	8
MLP_AE	86	9
CNN_GRU	123	10
CNN_LSTM	123	11
Dummy	144	12

Table 2. Model rankings, sorted by rank

The model that achieved the best ranking was GRU LSTM, while 2 points behind that in 2nd place was LSTM GRU. All models beat the dummy model, which ranked 12th overall and for each individual security. The joint worst performing DL models were the CNN GRU and CNN LSTM models.

The CNN family of models: CNN, CNN GRU and CNN LSTM, were some of the worst performing models, generating the highest cumulative MSE scores and occupying positions 8, 10 and 11. The GRU family of models: GRU, GRU AE, GRU LSTM, returned some of the best scores, occupying ranks 1st, 3rd and 4th.

Close grouping at the top of the ranking table occurred. The top 4 positions all fell within a 9 point range of each other, starting at 36 points for 1st and 45 points for 4th. There was a notable jump of 27 points between 4th and 5th, LSTM AE coming in at 5th with 72 points against the 45 point score of GRU AE in 4th.

The lowest MSE overall was achieved by the GRU AE model on the Bund 10y security, with an MSE of 0.000926. The highest overall MSE achieved by a non-dummy model was MLP AE predicting Amazon prices, with an MSE of 0.00884.

The same ranking methodology was applied to every security in order to gauge each securities influence over the MSE. The securities that had the lowest cumulative MSEs were Bund 10y and Treasury 10y with 19 points each. The security that generated the highest MSEs were aluminium futures with 131 points, see Table 3.

Security	Points	Rank
Bund10y	19	1
Treasury 10y	19	2
Gilt10y	44	3
Nvidia	47	4
EURCHF	70	5
Amazon	82	6
Google	90	7
GBPUSD	92	8
EURUSD	98	9
Cu	115	10
Corn	129	11
Al	131	12

Table 3. Security rankings, sorted by rank

It can be noted that commodities generated the highest MSEs across models, with the 3 securities from this asset class occupying the lowest 3 rankings. The inverse is true of fixed income, where the 3 government bonds, Bund 10y, Treasury 10y and Gilt 10y, occupied the top 3 positions with the lowest cumulative scores.

The volatility of each security was recorded, see Appendix 3. Table 4 shows the correlation between the volatility and the number of cumulative points scored by each security. Some degree of inverse correlation occurs between the number of points and the volatility of the whole, training and test datasets. The security with the highest volatility, Bund 10y, ranked 1st as the security with the lowest cumulative score.

	Whole Dataset & Points	Train Dataset & Points	Test Dataset & Points	Train/Test Ratio & Points
Correlation	-55.40%	-55.83%	-52.35%	34.05%

Table 4. Volatility correlation with cumulative points

Table 5 shows which hyperparameters were most commonly used by the optimal models. SELU was the most regularly used activation function, whilst RELU was by far the least used option. This suggests that the self-normalising nature of SELU (Klambauer *et al.*, 2017) provides a performance benefit when considering time series data with DL architectures.

The most used optimiser by the final models was Adam, whilst a batch size of 32 was preferred to a similar degree. Learning rate choices were the most evenly distributed of hyperparameters.

It should be noted that for some models, SGD was an optimiser included in the hyperparameter search, though none of the best search results returned this choice. This indicates that for time series analysis: Adam and nAdam represent a significant improvement. In addition, RMSprop seems ill-suited to this type of problem, being used only 9 times.

Hyperparameter		Use Count
Activation Function	SELU	65
	tanh	52
	RELU	15
Optimiser	Adam	89
	nAdam	34
	RMSprop	9
Batch Size	32	80
	64	38
	128	14
Learning Rate	0.001	49
	0.005	45
	0.0005	38

Table 5. Most commonly used hyperparameters

As noted from Table 6 the number of trainable parameters displays positive correlation with the number of points scored. The CNN hybrid models: CNN LSTM and CNN GRU, had a much larger average number of trainable parameters than other models, yet were the worst performing models. The comparative magnitude of these models average trainable parameters should be noted. When CNN LSTM and CNN GRU are ignored as outliers, the correlation drops to a mere 14%.

	With Outliers	Without Outliers
Correlation	80.10%	14.44%

Table 6. Trainable parameter correlation with cumulative points

An additional extension exercise was undertaken where the prediction length of the model was increased from 1 day to 15 days, with one prediction for each day. The output vector of the model was changed from a vector with a single element to a vector with 15 elements. As noted from Table 7, there is strong correlation between each of the models 1 day and 15 day MSE, RMSE and MAE values.

	MSE	MAE	RMSE
Correlation	83.09%	78.20%	90.00%

Table 7. Correlation between 1 day and 15 day error metrics

A number of points must be considered before drawing conclusions. A limiting factor in this study was the focus on univariate price prediction. There is heavy correlation between security prices and certain economic factors, either as leading or lagging indicators (Graham, Dodd & Buffett, 2009). These factors may improve the performance of the models to a varying degree depending on architecture, yet these relationships are untouched by this study.

Intraday price moves are one of the major contributing factors to a securities closing price (Taleb, 1997). More specifically, large standard deviation price moves in the first half of the trading day are more likely than not to be sustained into the market close (Taleb, 2004), and as such heavily influence the closing price. These moves are not captured or examined by this study.

The limited scope of the extension exercise is additionally noted. The use of best performing models means limited conclusions can be inferred, poorly performing models may perform better at predicting 15 day windows than 1 day windows. This was not investigated as they were omitted from the exercise.

Furthermore, with 12 securities used in this study, though drawing from each of the major asset classes, the sample size is small compared to the large universe of potential securities. This results in a constrained ability to extrapolate conclusions beyond those securities that have been examined.

Moreover, the timeframe in which the study was conducted limited the length of time that could be dedicated to hyperparameter tuning. It is noted that hyperparameter tuning was relatively constrained in approach when compared to studies that focus on single model architectures (Sezer, Gudelek & Ozbayoglu, 2020).

The small size of the training dataset is another consideration. With roughly 1250 individual data points and being only ~5x larger than the test dataset, some models examined may benefit from a larger training period and improve their resulting performance.

What can be concluded is that GRUs are the most powerful models used in this study and represent a significant improvement on LSTMs from which they are derived. LSTM based models still perform strongly, being second only to GRU based models in the ranking system. LSTM and GRU hybrid models occupy the top 2 positions and represent powerful choices for researchers in this field.

CNNs struggle in this limited instance of univariate time series prediction. This may be due to the large number of parameters used requiring longer training periods and being prone to overfitting. Ignoring CNNs, it can be inferred that the number of trainable parameters has little bearing on the performance of a model, what is more important is the nature of the architectural features of the model.

According to the results of hyperparameter tuning, SELU and tanh represent compelling choices of activation functions when considering time series data. On a similar basis, Adam and nAdam are the optimisers that performed most strongly. They represent meaningful performance improvements when compared to RMSprop and SGD, which seem inappropriate choices for DL architectures applied to time series data.

The volatility of the time series being analysed somewhat explains a models ability to deliver accurate predictions. The higher the volatility, the lower the MSE is likely to be. This suggests there is an underlying relationship between the volatility of the security and the models ability to predict 1 day ahead prices, albeit in a counterintuitive inverse manner.

A models ability to predict 1 day ahead prices is not necessarily indicative of its ability to predict 15 days ahead due to the limited size of the extension exercise. Aside from that, it should be noted that the error metrics in the models used were heavily correlated between their 1 and 15 day ahead values.

There are a number of areas that may form the body of future research. A wider choice of hyperparameters may result in the discovery of new performant parameter combinations. Deeper models, that take longer to train, may return improved results when compared to the relatively short time available to train models used in this study. A wider comparison between a models ability to predict 1 and 15 day ahead prices may reveal interesting findings about certain architectures. In addition, the expansion to multivariate time series analysis would be the next logical step based upon the results of this study.

Appendicies

Appendix 1: Models used and associated abbreviations

Model Name	Abbreviation
Convolutional Neural Network	CNN
Convolutional Neural Network with Gated Recurrent Units	CNN GRU
Convolutional Neural Network with Long Short-Term Memory Units	CNN LSTM
Stacked Gated Recurrent Unit Neural Network	GRU
Stacked Gated Recurrent Unit Neural Network Autoencoder	GRU AE
Stacked Gated Recurrent Unit Neural Network with Long Short-Term Memory Units	GRU LSTM
Stacked Long Short-Term Memory Unit Neural Network	LSTM
Stacked Long Short-Term Memory Unit Neural Network Autoencoder	LSTM AE
Stacked Long Short-Term Memory Unit Neural Network with Gated Recurrent Units	LSTM GRU
Multi-Layer Perceptron Neural Network	MLP
Multi-Layer Perceptron Neural Network Auoencoder	MLP AE

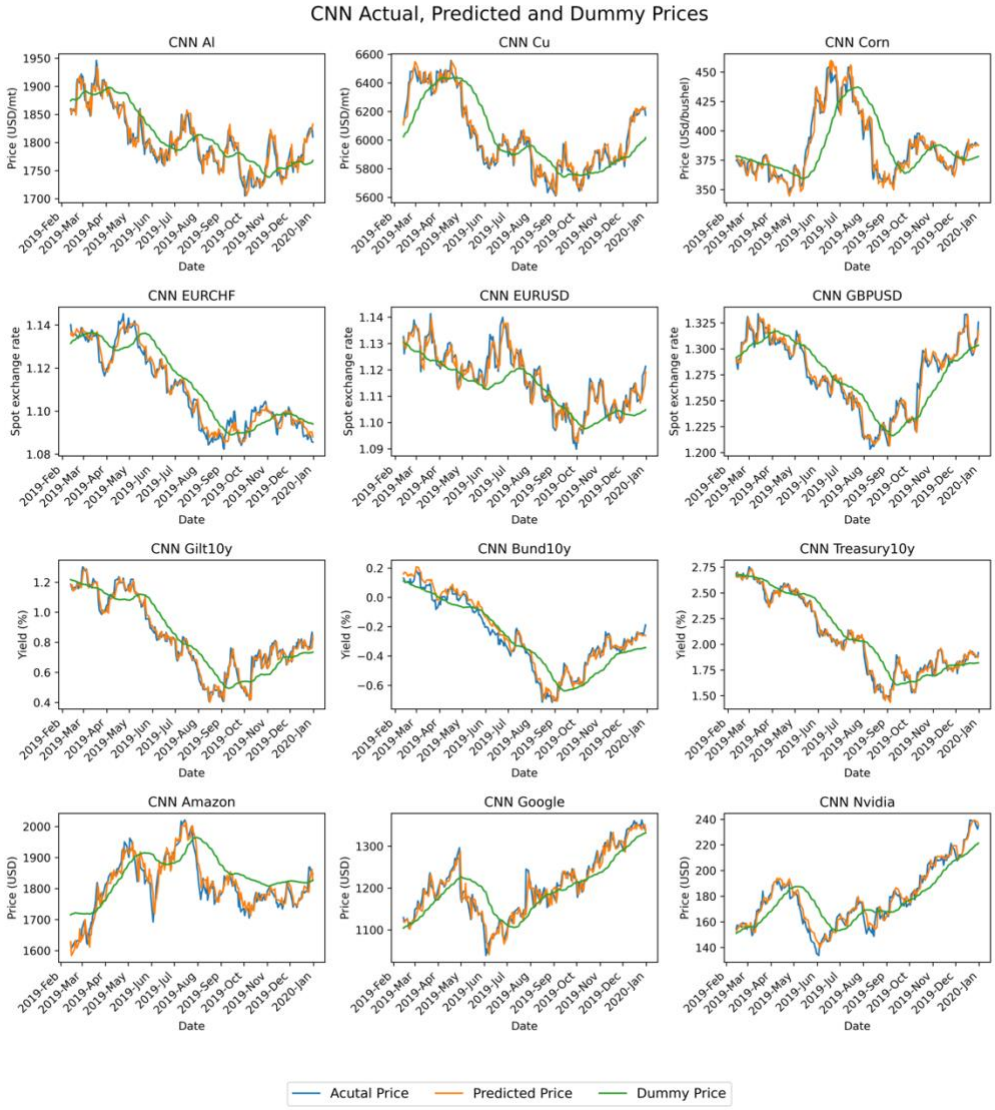
Appendix 2: Securities and associated details

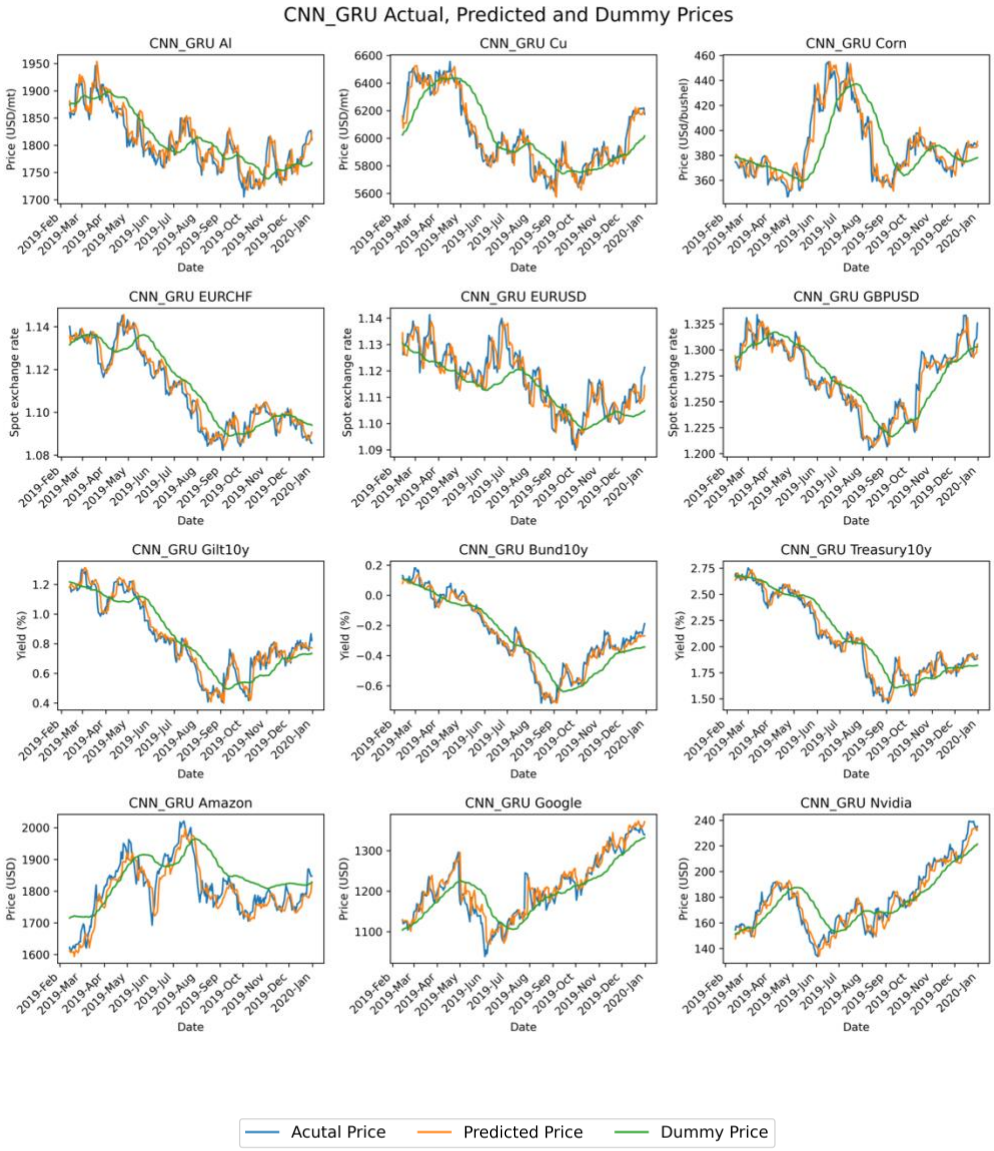
Abbreviation	Security	Description	Exchange	Country	Units
Al	3 month aluminium futures	Physically settled aluminium futures, for delivery in 3 months	LME	UK	USD/mt
Corn	Rolling active month corn futures	Physically settled corn futures, rolling active contract	CME	USA	USd/bushel
Cu	3 month copper futures	Physically settled copper futures, for delivery in 3 months	LME	UK	USD/mt
EURCHF	Euro / Swiss Franc spot exchange rate	Currency pair for settlement in 2 business days	OTC	Eurozone & Switzerland	-
EURUSD	Euro / United States Dollar spot exchange rate	Currency pair for settlement in 2 business days	OTC	Eurozone & USA	-
GBPUSD	British Pound / Euro spot exchange rate	Currency pair for settlement in 2 business days	OTC	UK & USA	-
Gilt10y	Rolling 10y British Gilt yield	The yield of rolling 10y Gilt	LSE/OTC	UK	%
Bund10y	Rolling 10y German Bund yield	The yield of the rolling 10y Bund	FSE/OTC	Germany	%
Treasury10y	Rollitn 10y United States Treasury yield	The yield of the rolling 10y Treasury	OTC	USA	%
Amazon	Amazon.com Inc. common stock	The stock price of Amazon	NASDAQ	USA	USD/share
Google	Alphabet Inc. class A common stock	The stock price of Alphabet	NASDAQ	USA	USD/share
Nvidia	NVIDIA Corporation common stock	The stock price of Nvidia	NASDAQ	USA	USD/share

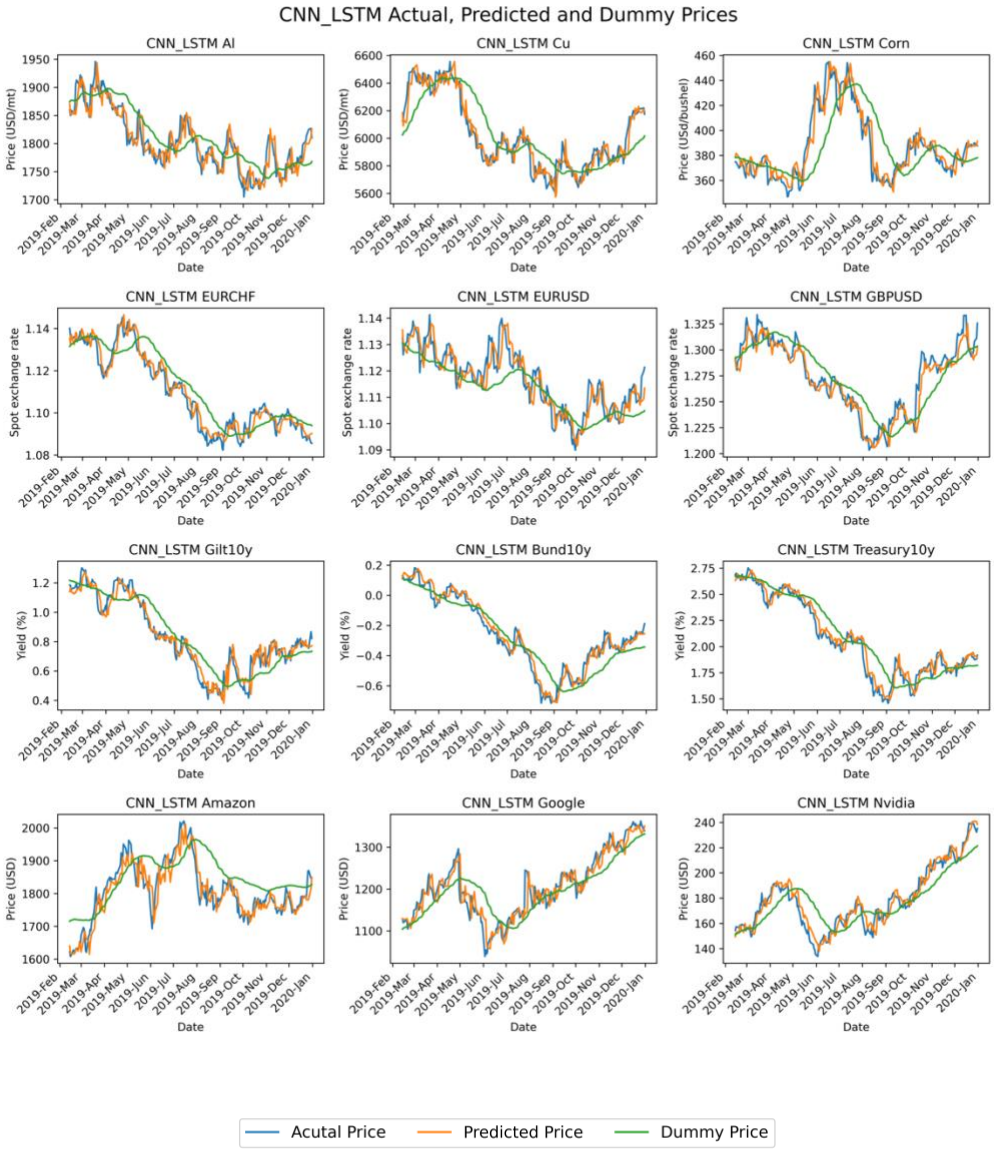
Appendix 3: Security volatilities and cumulative points, sorted by whole dataset volatility

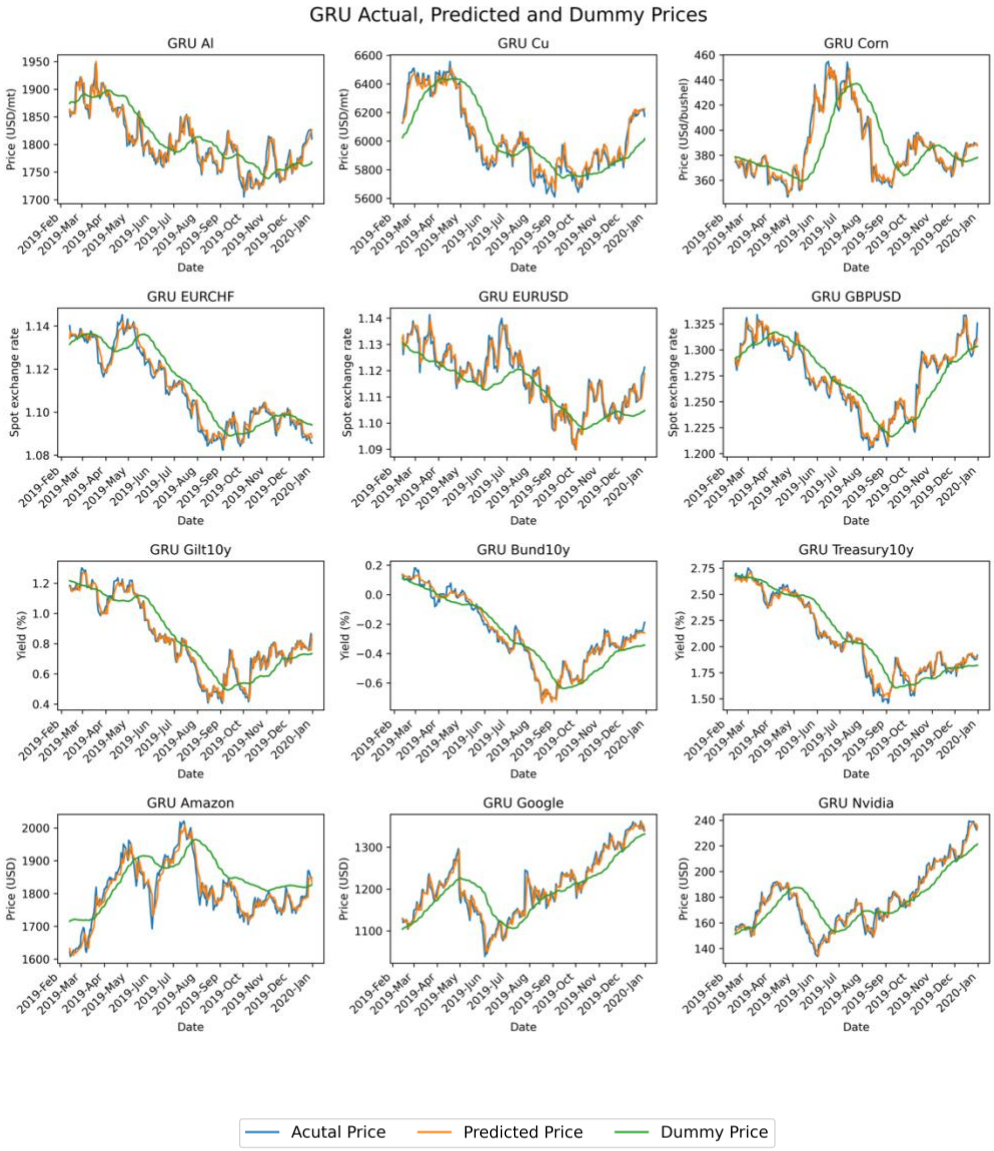
Security	Whole Dataset Volatility	Train Dataset Volatility	Test Dataset Volatility	Train/Test Ratio	Points
Bund10y	4.235148	3.637762	9.861522	0.368884	19
Gilt10y	0.600167	0.52237	0.895036	0.58363	44
Nvidia	0.405764	0.404163	0.414818	0.974314	47
Treasury10y	0.324428	0.317194	0.359283	0.882852	19
Amazon	0.301493	0.313541	0.23195	1.351765	82
Google	0.236191	0.235455	0.240532	0.978891	90
Al	0.185155	0.192454	0.142202	1.353382	131
Cu	0.172372	0.178017	0.138755	1.282964	115
EURCHF	0.091292	0.098694	0.036389	2.712219	70
GBPUSD	0.084042	0.085633	0.075746	1.130523	92
EURUSD	0.074637	0.079173	0.045949	1.723064	98
Corn	0.056046	0.053597	0.058696	0.913116	129

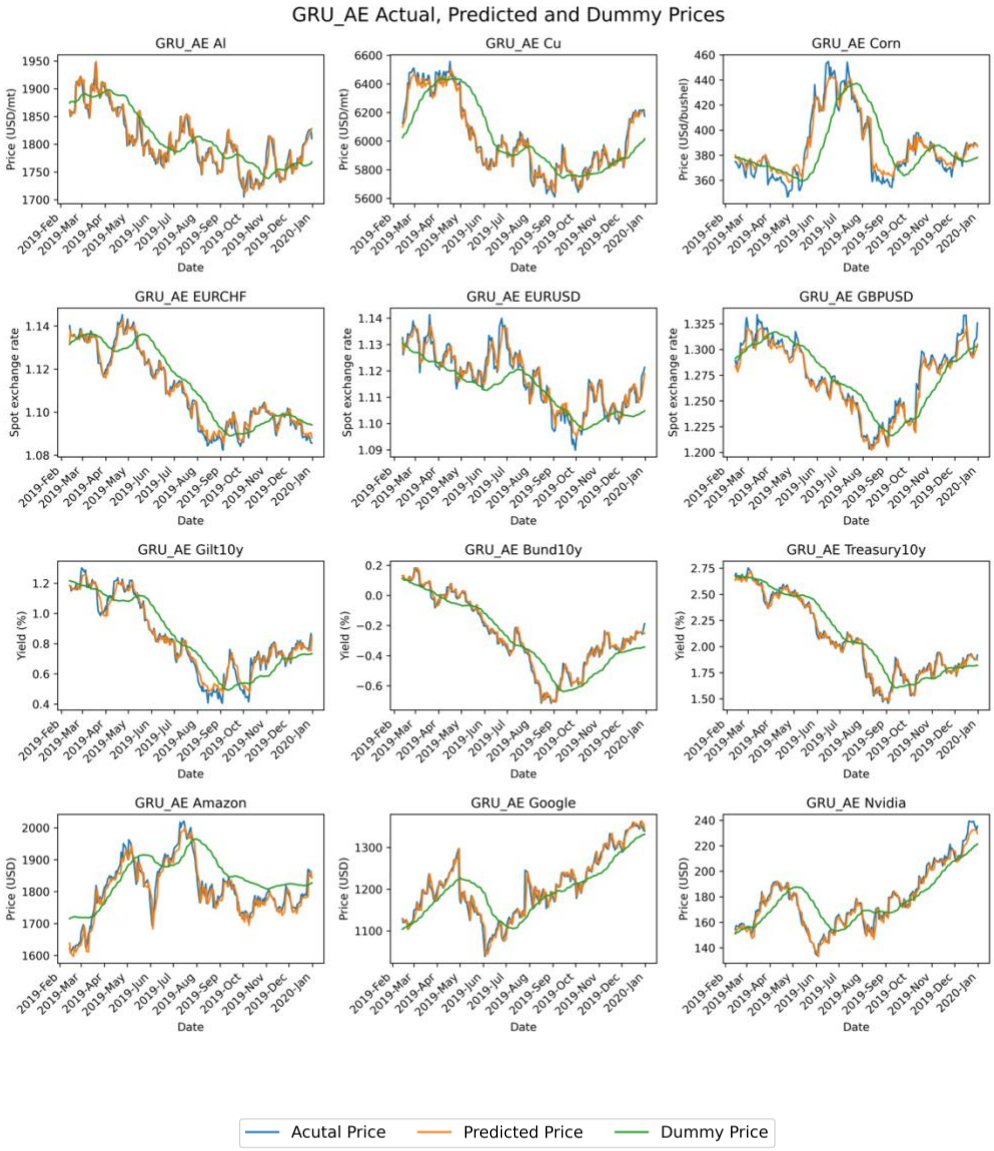
Appendix 4: Plots of model predictions

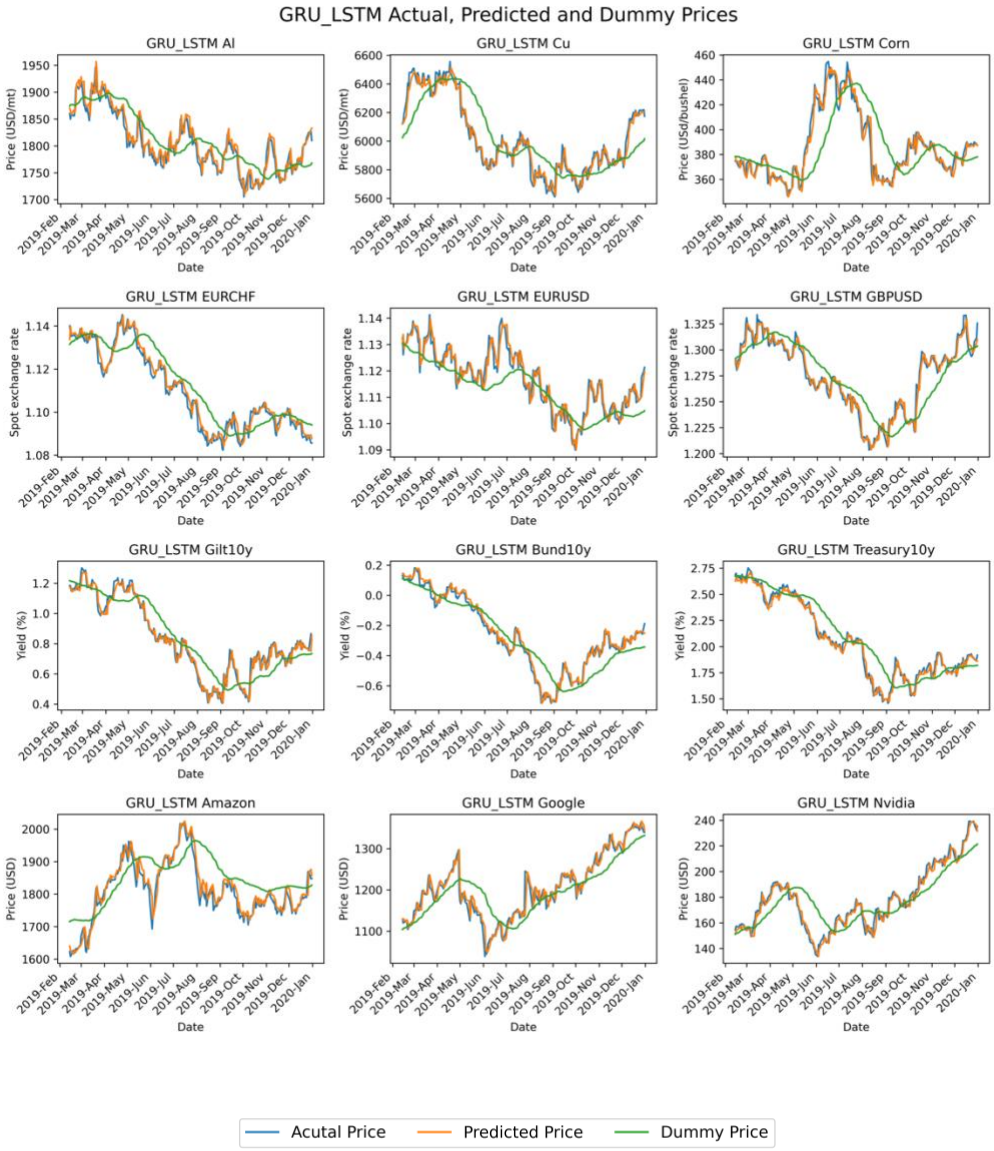


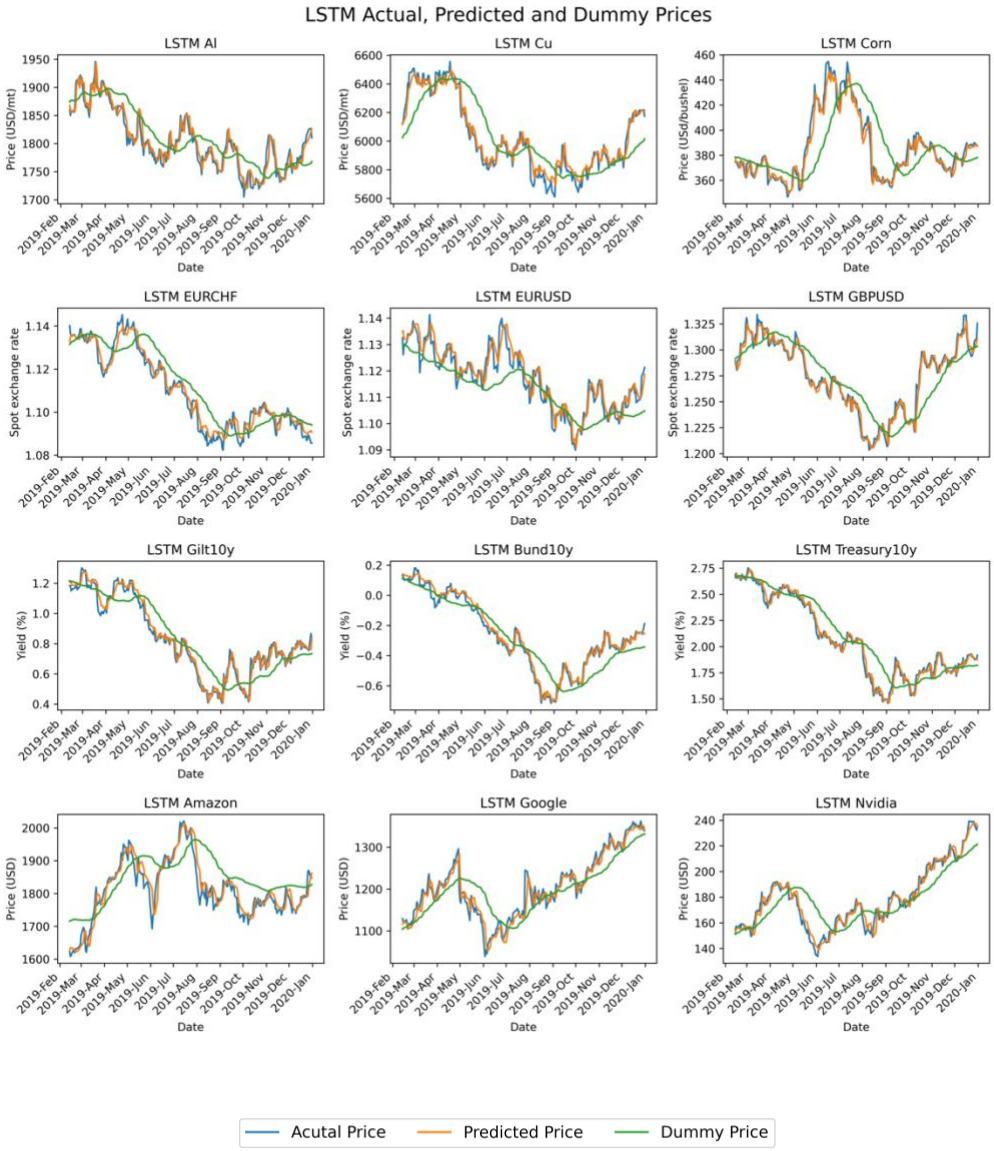

CNN_GRU Actual, Predicted and Dummy Prices
CNN_GRU Al
CNN_GRU Cu
CNN_GRU Corn
CNN_GRU EURCHF
CNN_GRU EURUSD
CNN_GRU GBPUSD
CNN_GRU Gilt10y
CNN_GRU Bund10y
CNN_GRU Treasury10y
CNN_GRU Amazon
CNN_GRU Google
CNN_GRU Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

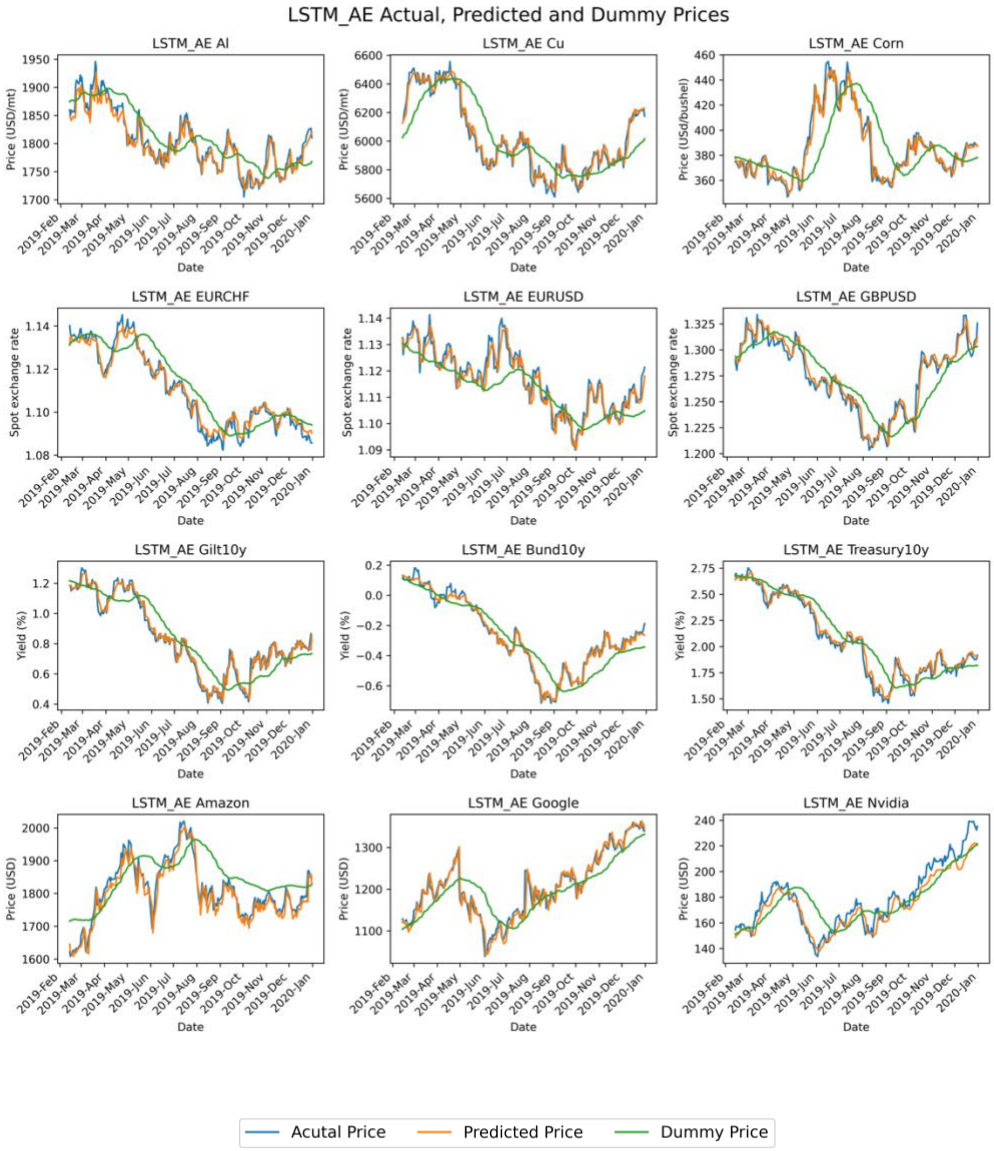

CNN_LSTM Actual, Predicted and Dummy Prices
CNN_LSTM Al
CNN_LSTM Cu
CNN_LSTM Corn
CNN_LSTM EURCHF
CNN_LSTM EURUSD
CNN_LSTM GBPUSD
CNN_LSTM Gilt10y
CNN_LSTM Bund10y
CNN_LSTM Treasury10y
CNN_LSTM Amazon
CNN_LSTM Google
CNN_LSTM Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

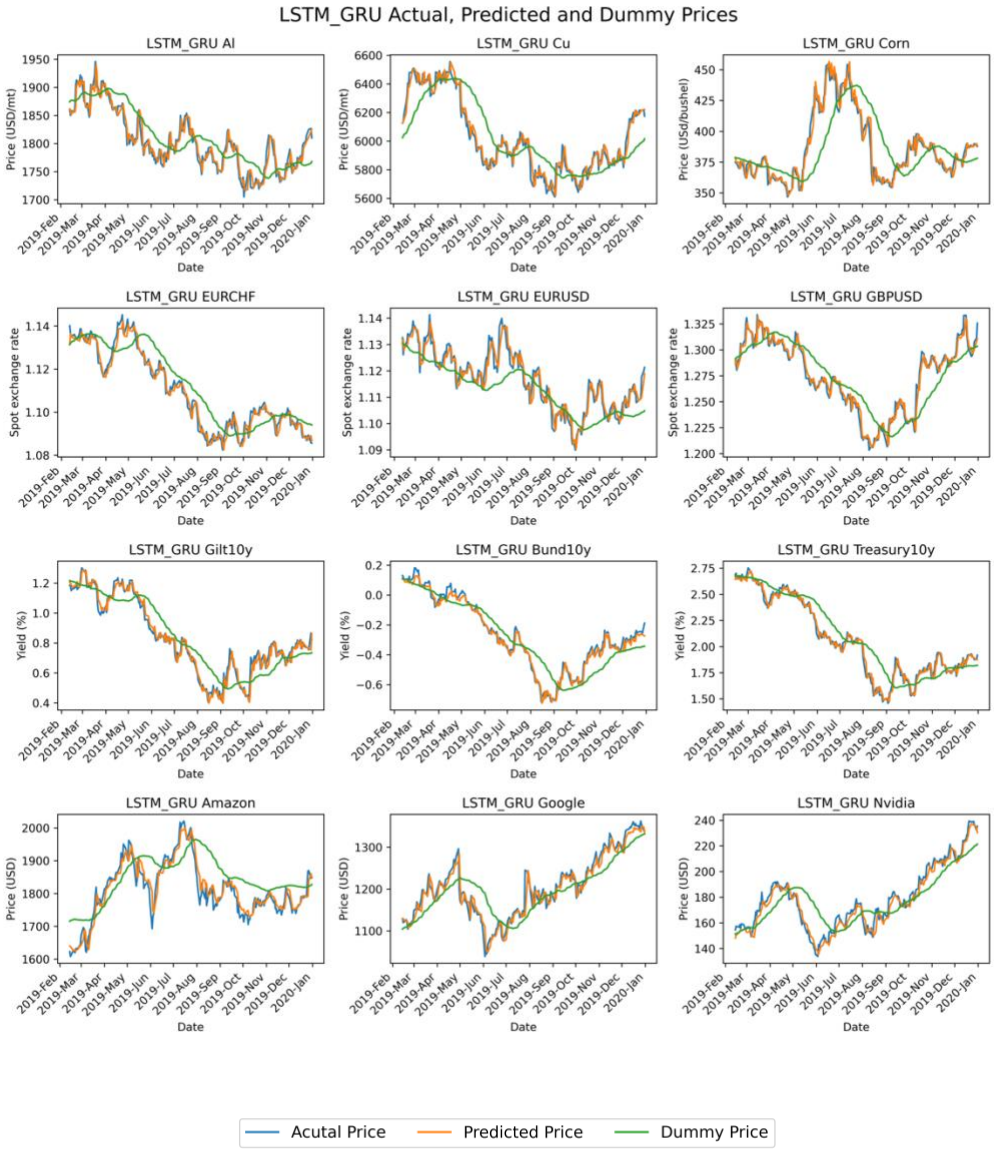

GRU Actual, Predicted and Dummy Prices
GRU Al
GRU Cu
GRU Corn
GRU EURCHF
GRU EURUSD
GRU GBPUSD
GRU Gilt10y
GRU Bund10y
GRU Treasury10y
GRU Amazon
GRU Google
GRU Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

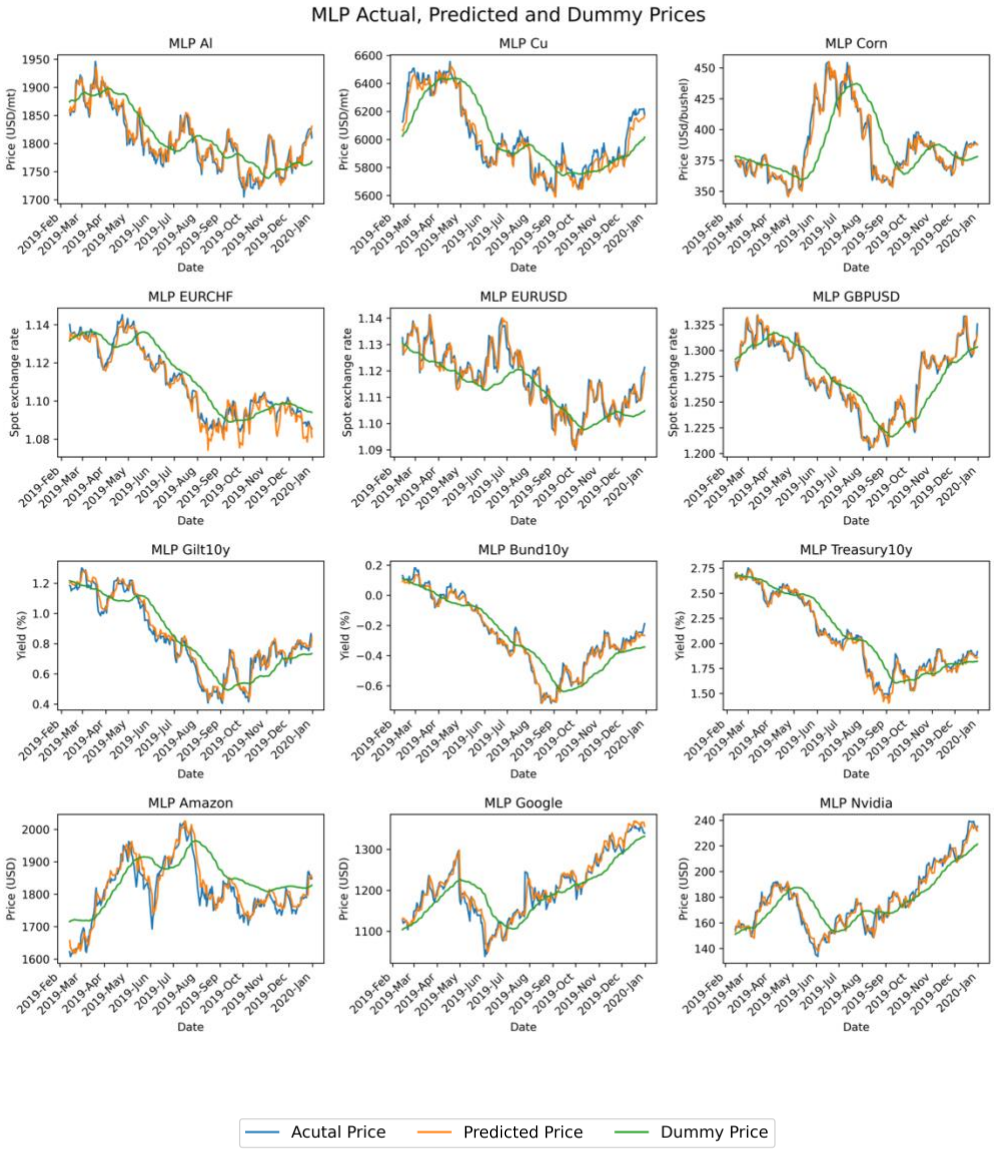

GRU_AE Actual, Predicted and Dummy Prices
GRU_AE Al
GRU_AE Cu
GRU_AE Corn
GRU_AE EURCHF
GRU_AE EURUSD
GRU_AE GBPUSD
GRU_AE Gilt10y
GRU_AE Bund10y
GRU_AE Treasury10y
GRU_AE Amazon
GRU_AE Google
GRU_AE Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

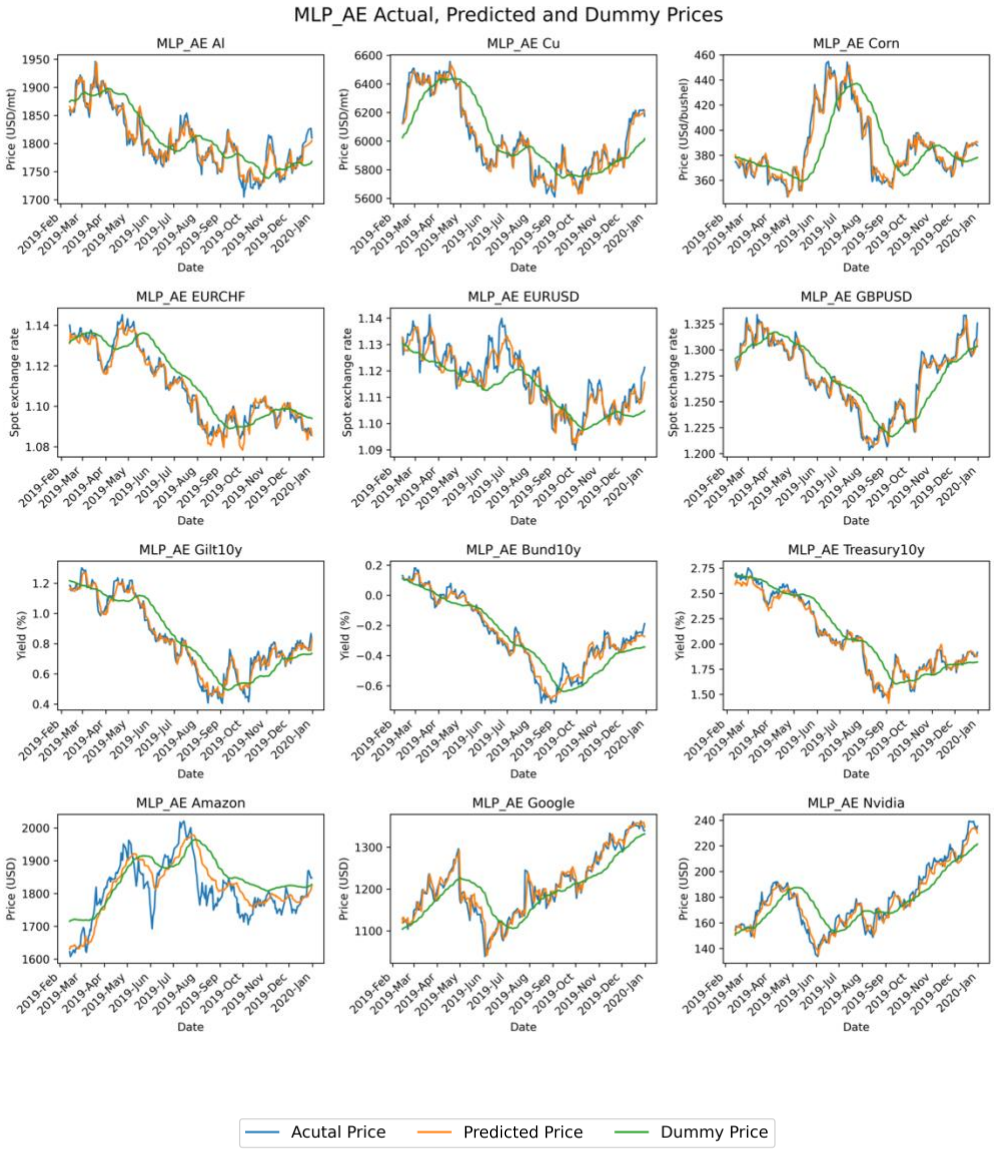

GRU_LSTM Actual, Predicted and Dummy Prices
GRU_LSTM Al
Price (USD/mt)
GRU_LSTM Cu
Price (USD/mt)
GRU_LSTM Corn
Price (USd/bushel)
GRU_LSTM EURCHF
Spot exchange rate
GRU_LSTM EURUSD
Spot exchange rate
GRU_LSTM GBPUSD
Spot exchange rate
GRU_LSTM Gilt10y
Yield (%)
GRU_LSTM Bund10y
Yield (%)
GRU_LSTM Treasury10y
Yield (%)
GRU_LSTM Amazon
Price (USD)
GRU_LSTM Google
Price (USD)
GRU_LSTM Nvidia
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price


LSTM Actual, Predicted and Dummy Prices
LSTM Al
LSTM Cu
LSTM Corn
LSTM EURCHF
LSTM EURUSD
LSTM GBPUSD
LSTM Gilt10y
LSTM Bund10y
LSTM Treasury10y
LSTM Amazon
LSTM Google
LSTM Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

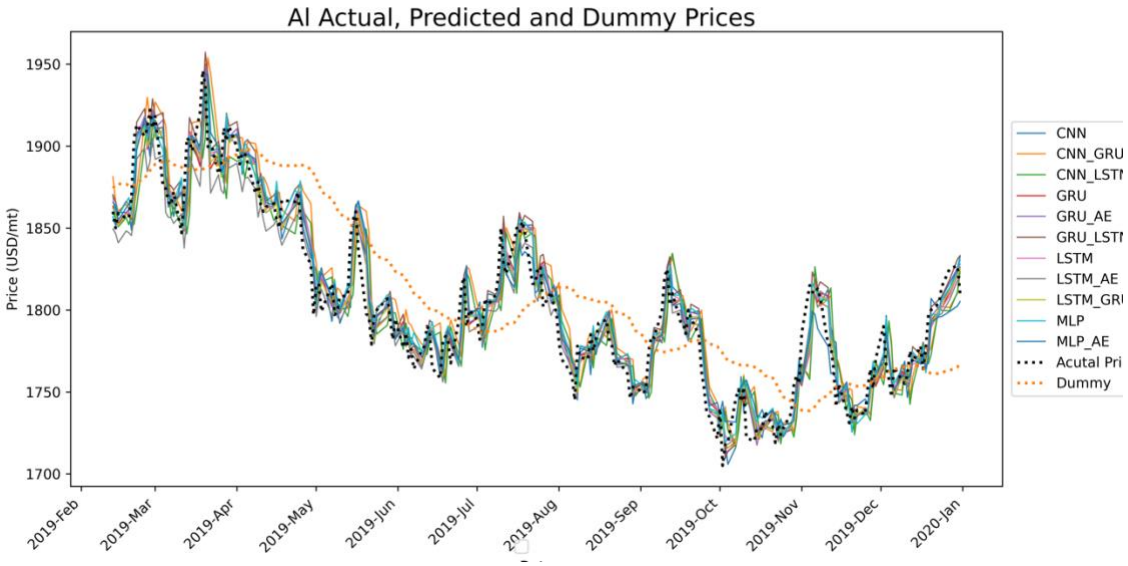

LSTM_AE Actual, Predicted and Dummy Prices
LSTM_AE Al
LSTM_AE Cu
LSTM_AE Corn
LSTM_AE EURCHF
LSTM_AE EURUSD
LSTM_AE GBPUSD
LSTM_AE Gilt10y
LSTM_AE Bund10y
LSTM_AE Treasury10y
LSTM_AE Amazon
LSTM_AE Google
LSTM_AE Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

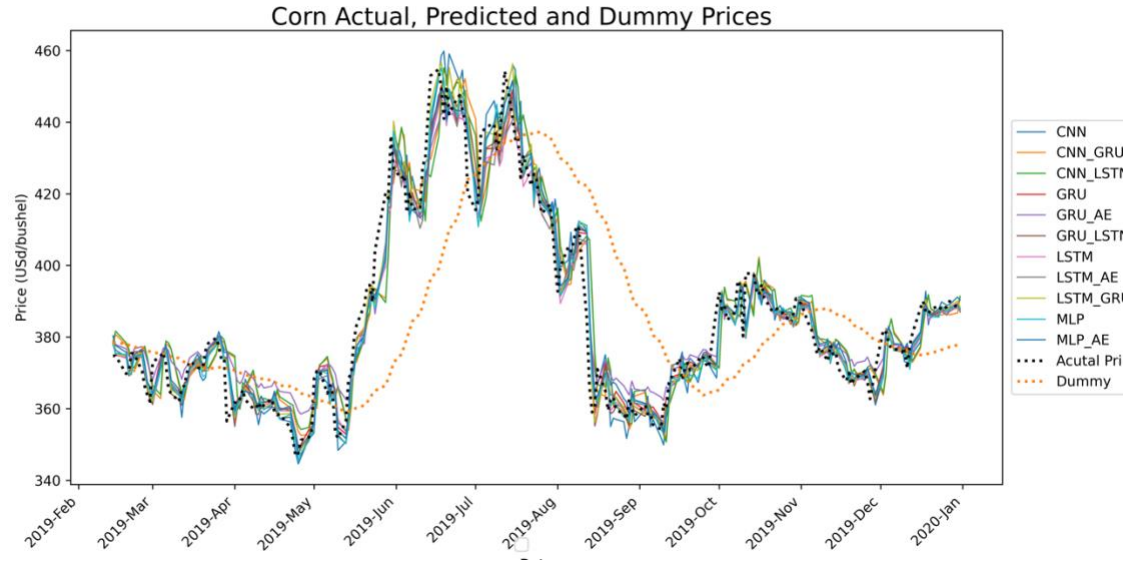

LSTM_GRU Actual, Predicted and Dummy Prices
LSTM_GRU Al
LSTM_GRU Cu
LSTM_GRU Corn
LSTM_GRU EURCHF
LSTM_GRU EURUSD
LSTM_GRU GBPUSD
LSTM_GRU Gilt10y
LSTM_GRU Bund10y
LSTM_GRU Treasury10y
LSTM_GRU Amazon
LSTM_GRU Google
LSTM_GRU Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

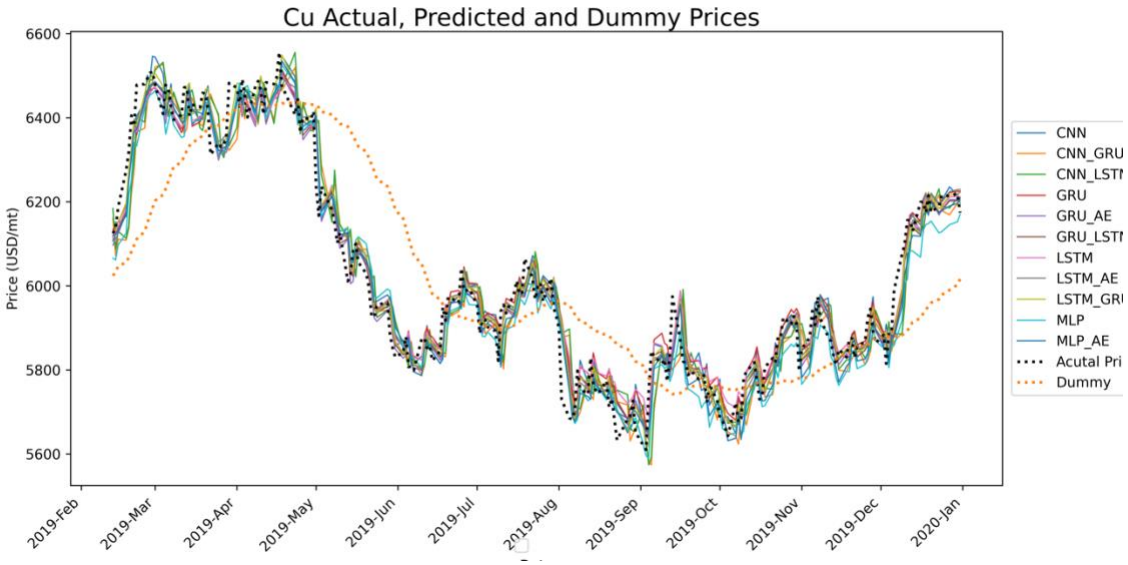

MLP Actual, Predicted and Dummy Prices
MLP Al
MLP Cu
MLP Corn
MLP EURCHF
MLP EURUSD
MLP GBPUSD
MLP Gilt10y
MLP Bund10y
MLP Treasury10y
MLP Amazon
MLP Google
MLP Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

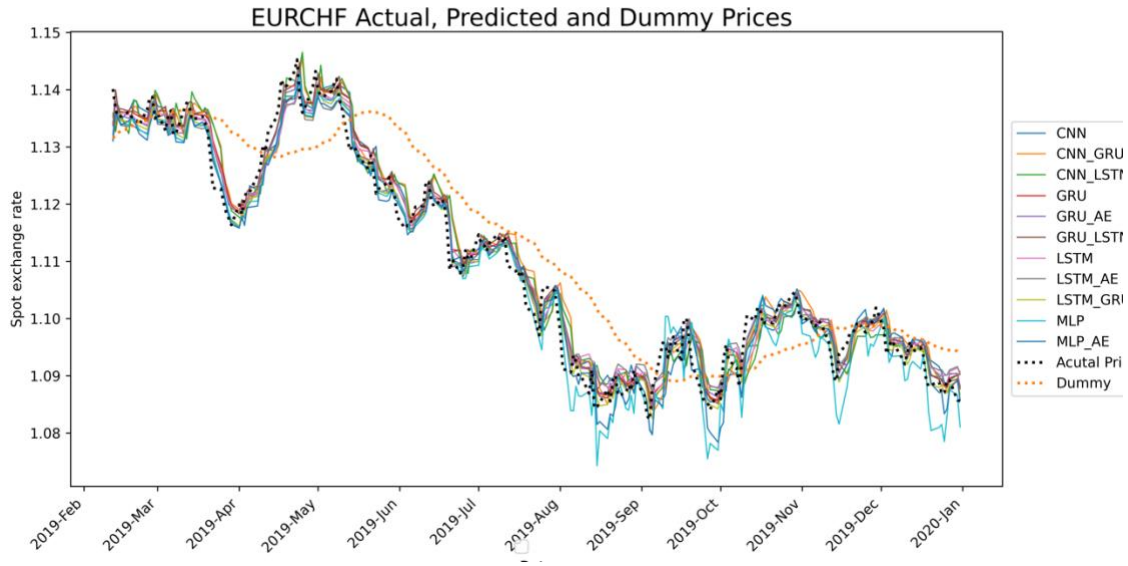

MLP_AE Actual, Predicted and Dummy Prices
MLP_AE Al
MLP_AE Cu
MLP_AE Corn
MLP_AE EURCHF
MLP_AE EURUSD
MLP_AE GBPUSD
MLP_AE Gilt10y
MLP_AE Bund10y
MLP_AE Treasury10y
MLP_AE Amazon
MLP_AE Google
MLP_AE Nvidia
Price (USD/mt)
Price (USd/bushel)
Spot exchange rate
Yield (%)
Price (USD)
Date
Acutal Price
Predicted Price
Dummy Price

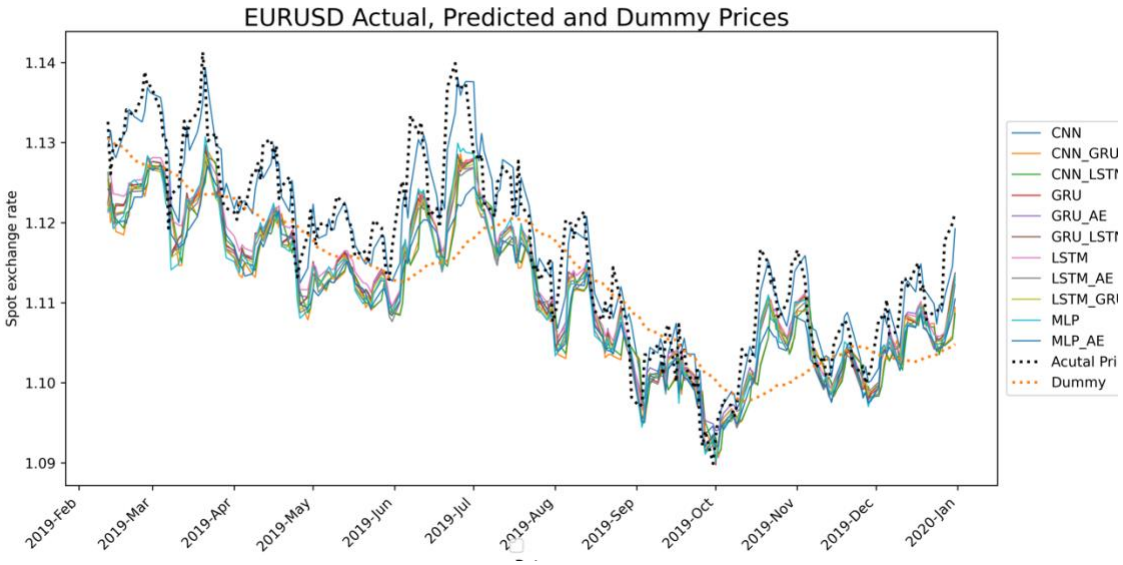
Appendix 5: Plots of model predictions grouped by security

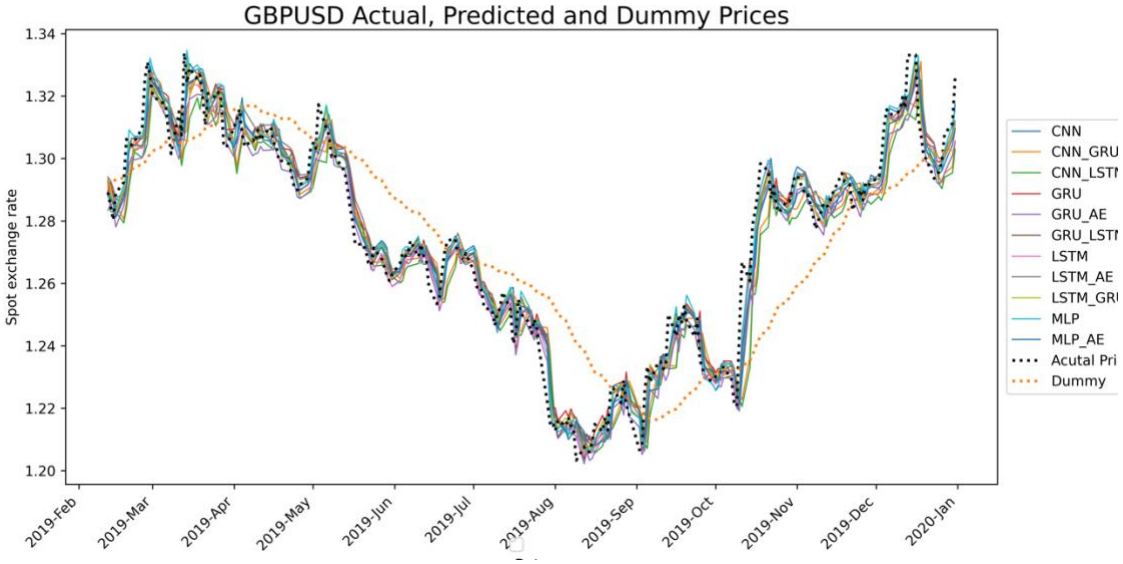


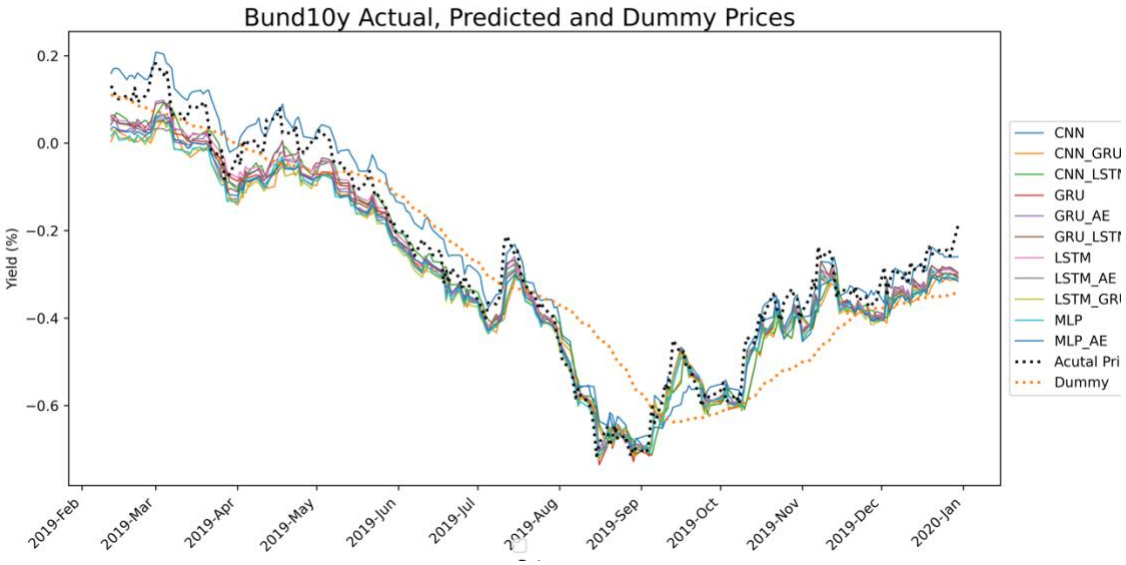


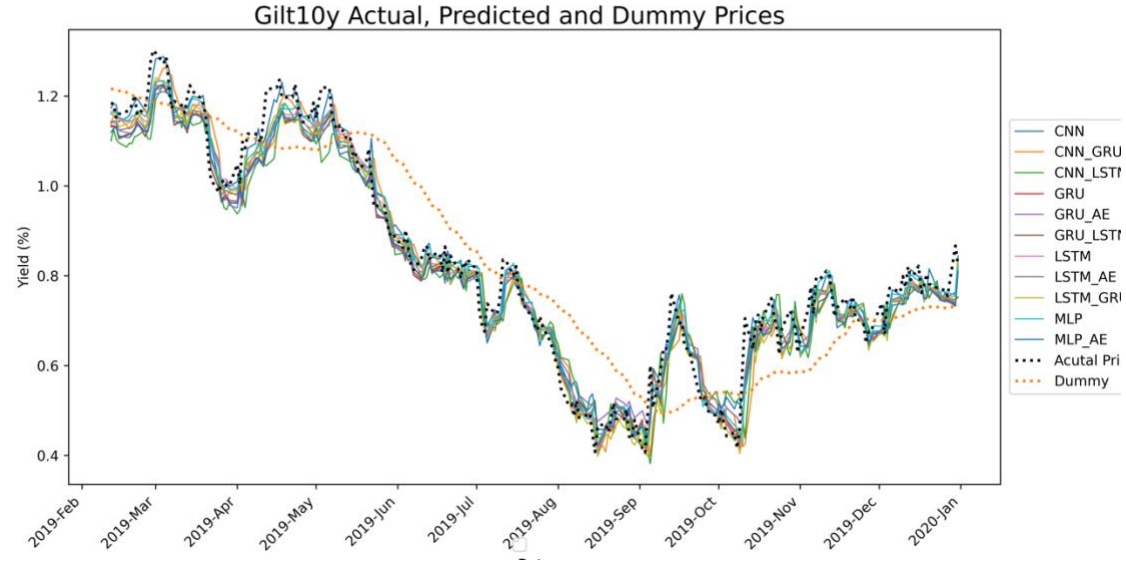

Cu Actual, Predicted and Dummy Prices
Price (USD/mt)
6600
6400
6200
6000
5800
5600
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

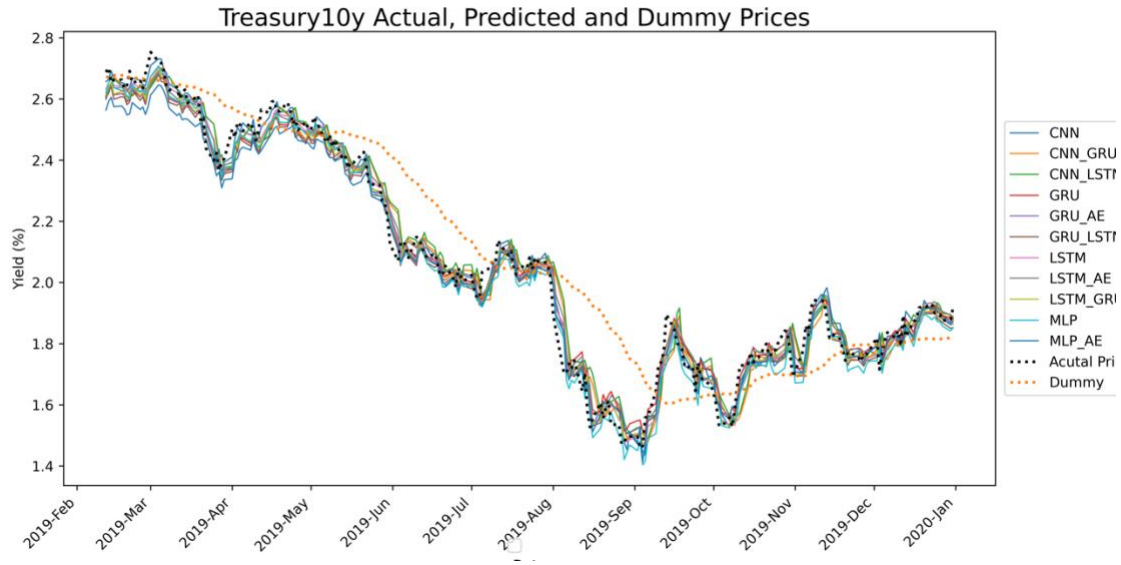

EURCHF Actual, Predicted and Dummy Prices
Spot exchange rate
1.15
1.14
1.13
1.12
1.11
1.10
1.09
1.08
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

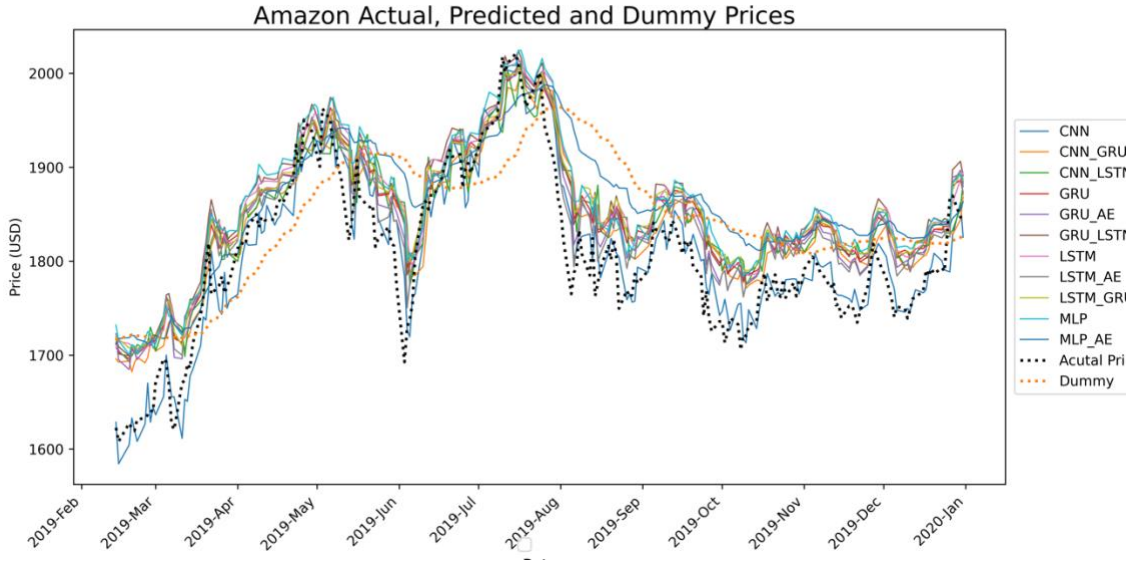

EURUSD Actual, Predicted and Dummy Prices
Spot exchange rate
1.14
1.13
1.12
1.11
1.10
1.09
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

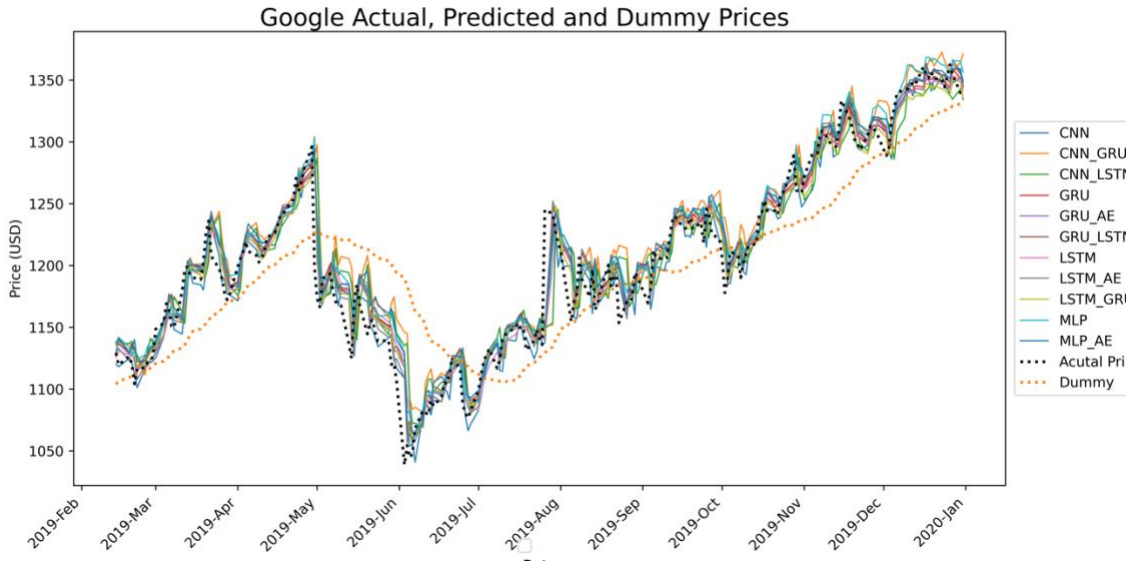

GBPUSD Actual, Predicted and Dummy Prices
Spot exchange rate
1.34
1.32
1.30
1.28
1.26
1.24
1.22
1.20
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

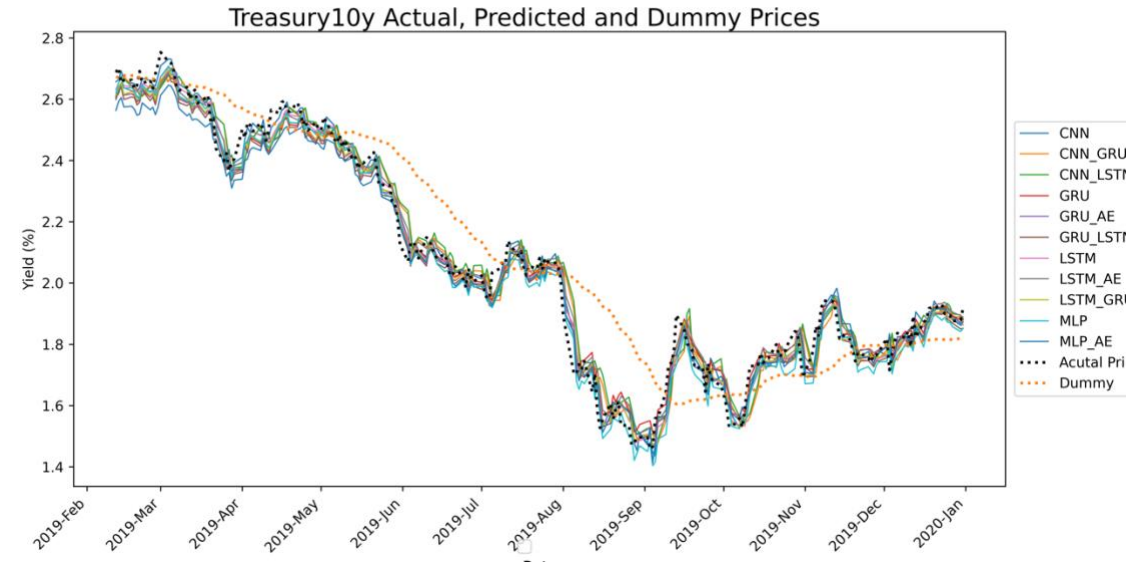

Bund10y Actual, Predicted and Dummy Prices
Yield (%)
0.2
0.0
−0.2
−0.4
−0.6
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTM
GRU
GRU_AE
GRU_LSTM
LSTM
LSTM_AE
LSTM_GRU
MLP
MLP_AE
Acutal Price
Dummy


Gilt10y Actual, Predicted and Dummy Prices
Yield (%)
1.2
1.0
0.8
0.6
0.4
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTM
GRU
GRU_AE
GRU_LSTM
LSTM
LSTM_AE
LSTM_GRU
MLP
MLP_AE
Acutal Price
Dummy

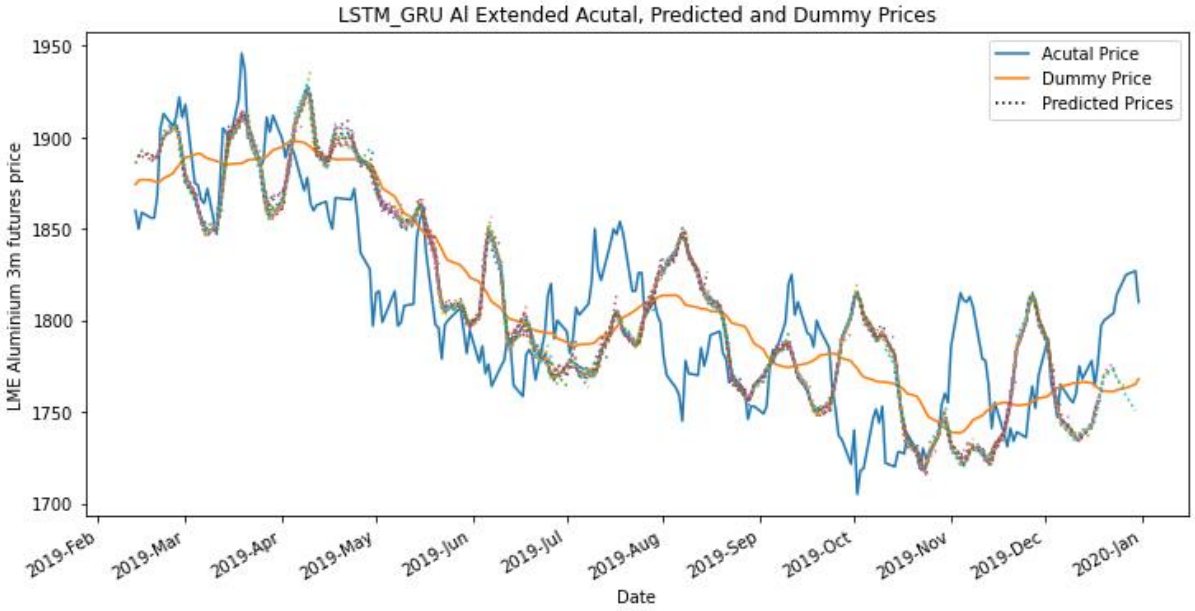

Treasury10y Actual, Predicted and Dummy Prices
Yield (%)
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

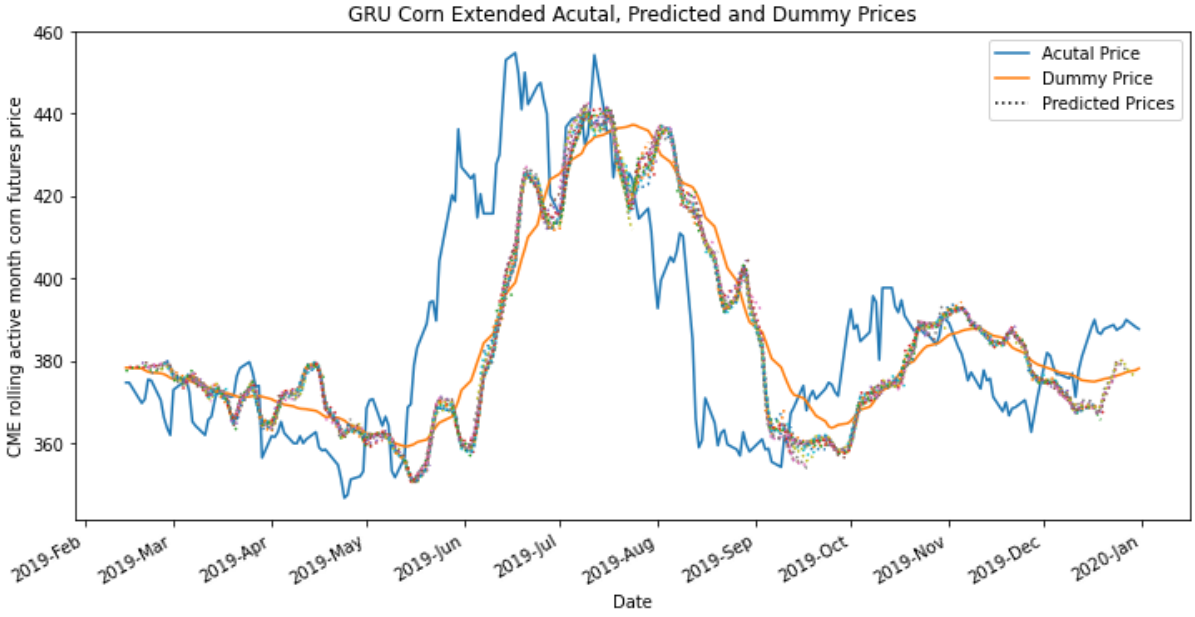

Amazon Actual, Predicted and Dummy Prices
Price (USD)
2000
1900
1800
1700
1600
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

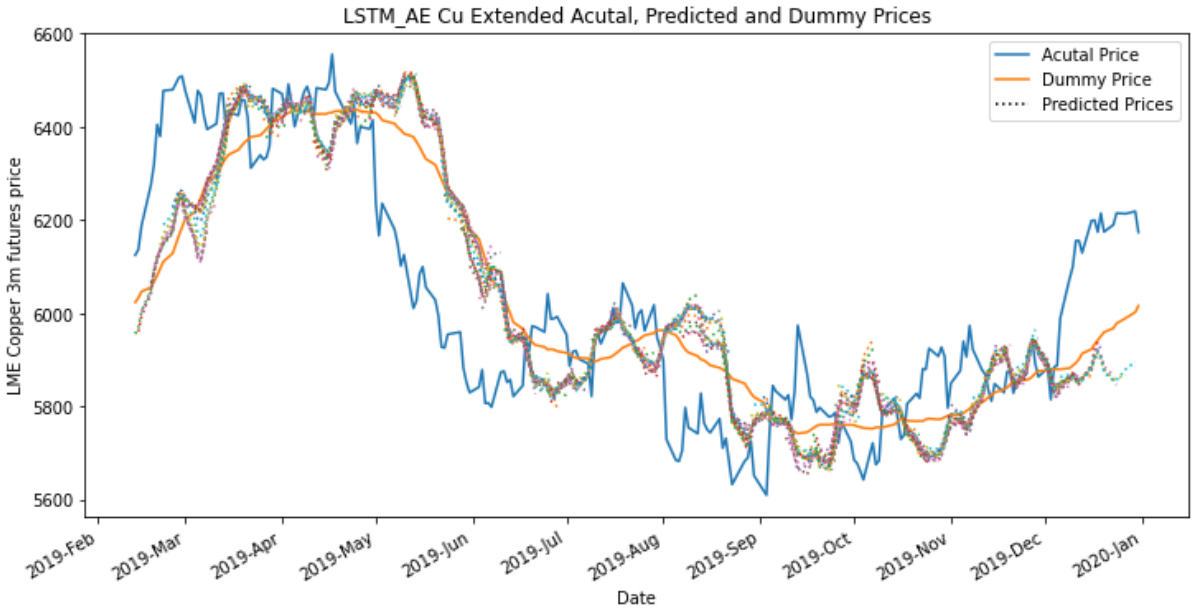

Google Actual, Predicted and Dummy Prices
Price (USD)
1350
1300
1250
1200
1150
1100
1050
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

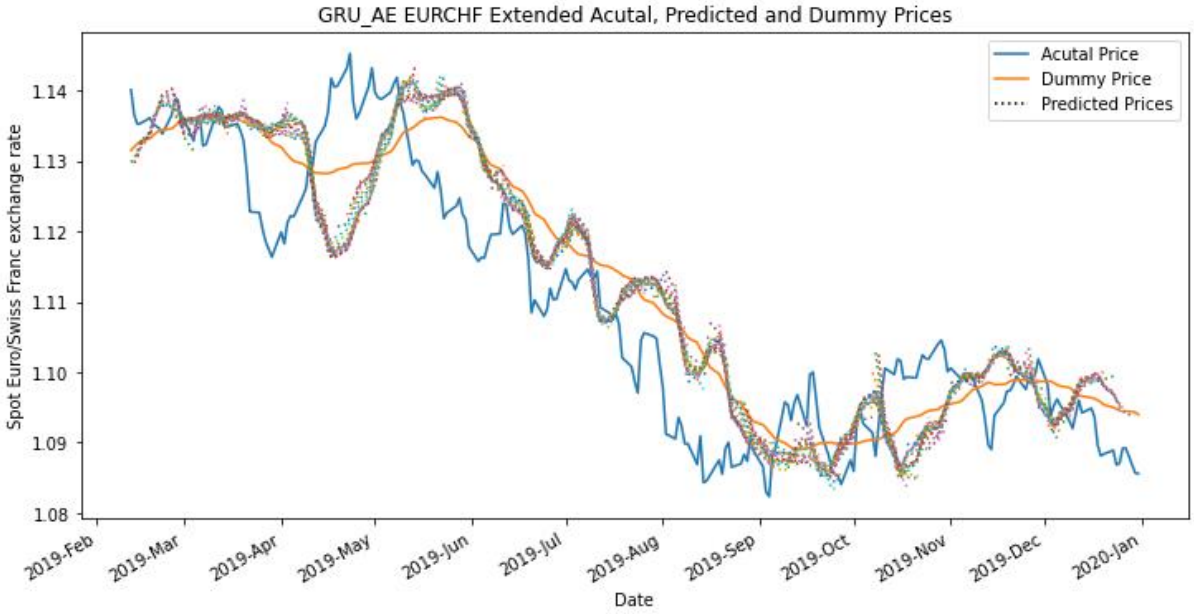

Treasury10y Actual, Predicted and Dummy Prices
Yield (%)
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
CNN
CNN_GRU
CNN_LSTI
GRU
GRU_AE
GRU_LSTI
LSTM
LSTM_AE
LSTM_GRI
MLP
MLP_AE
Acutal Pri
Dummy

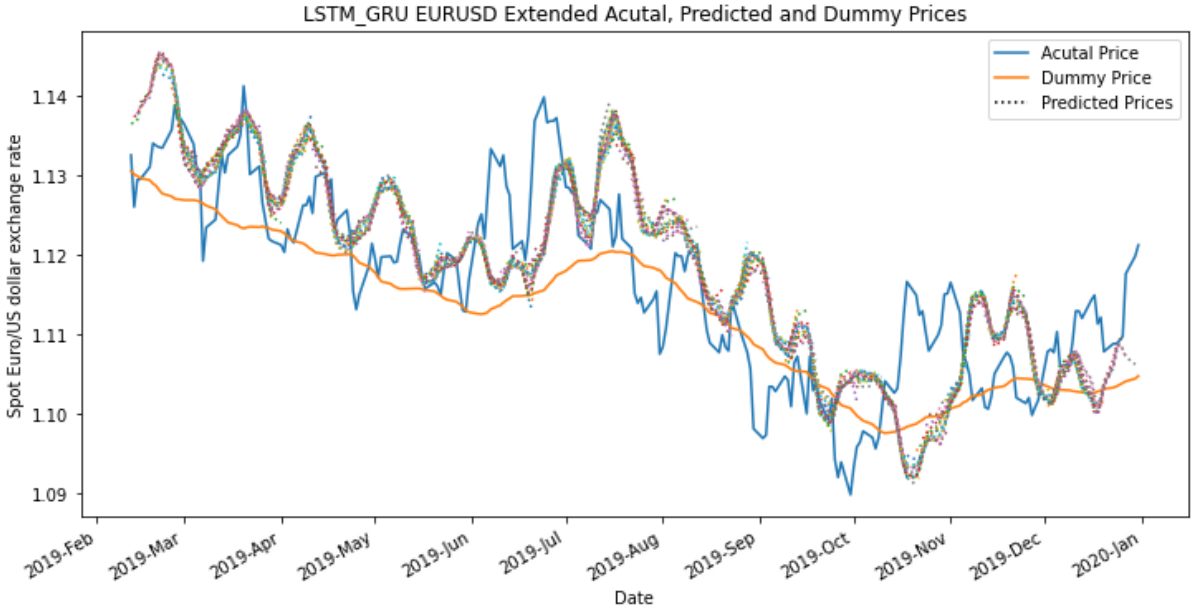
Appendix 6: Plots of extended model predictions

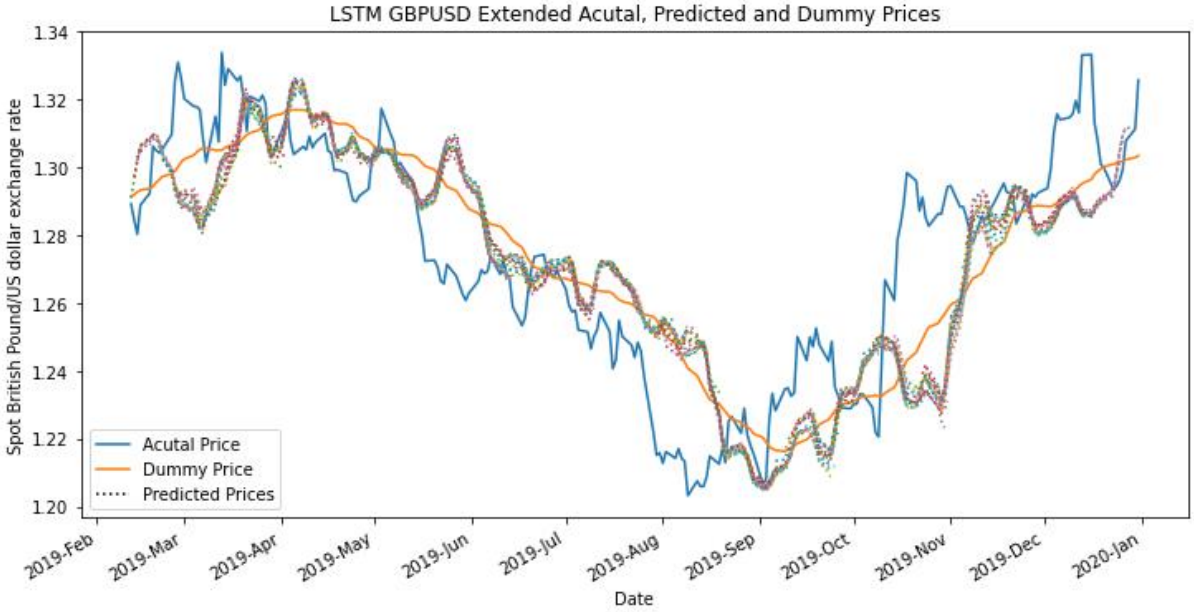


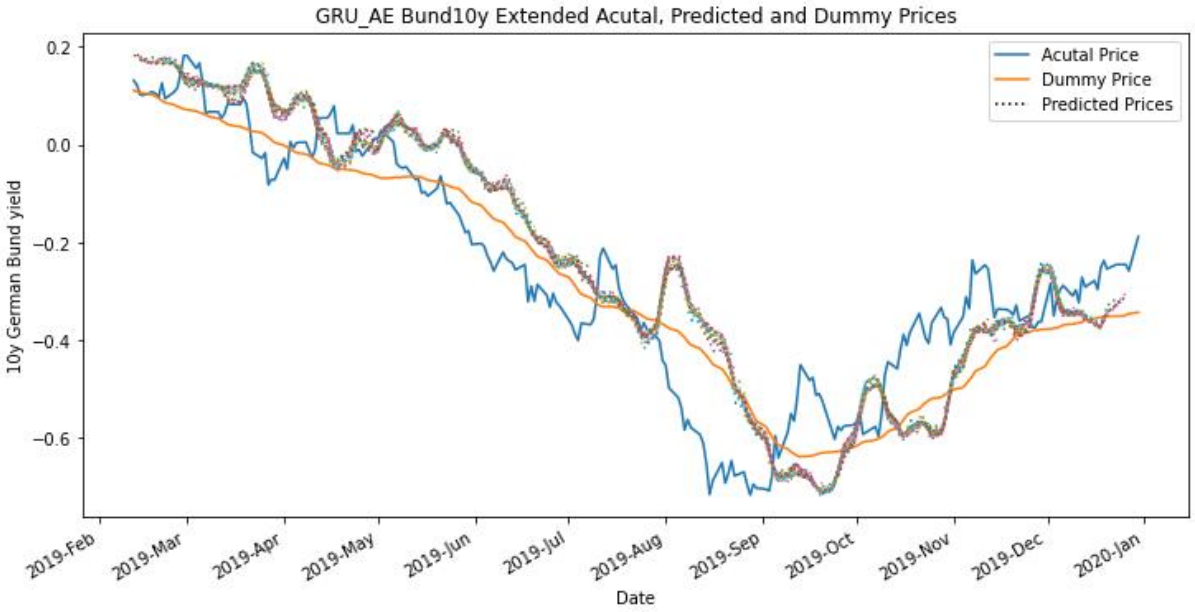


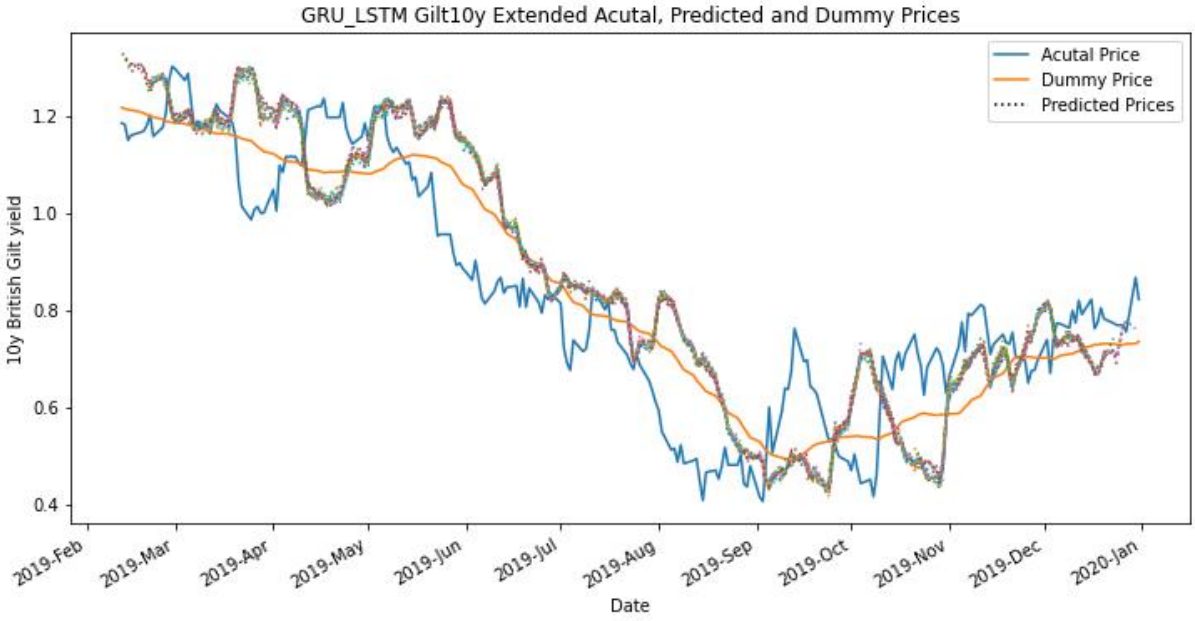

LSTM_AE Cu Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
LME Copper 3m futures price
6600
6400
6200
6000
5800
5600
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date

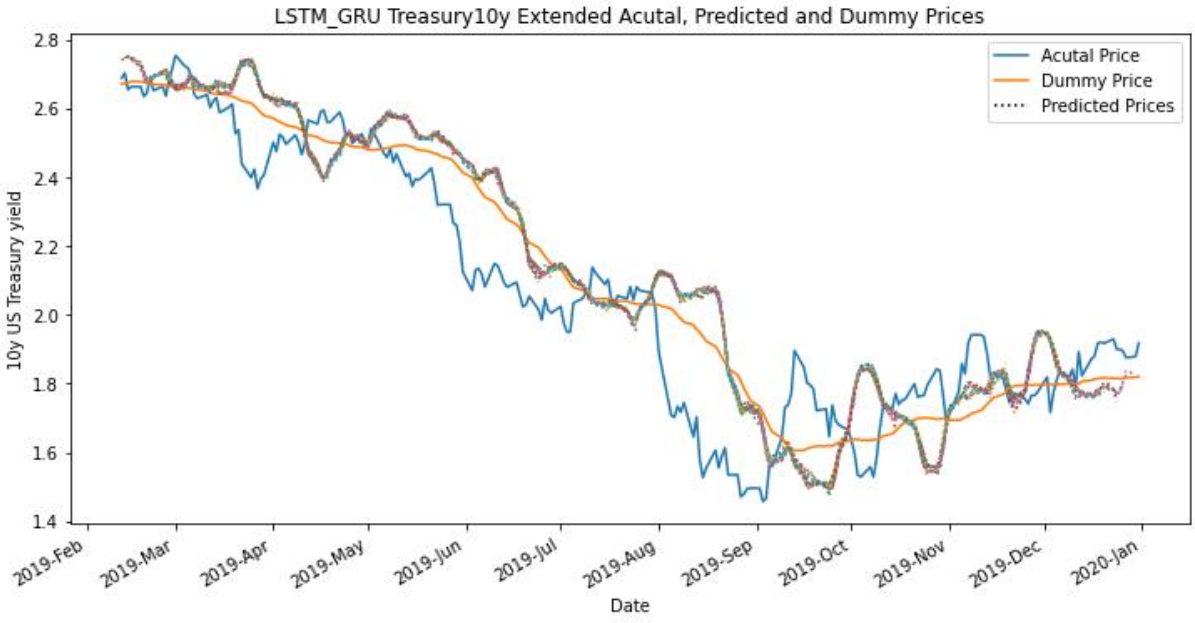

GRU_AE EURCHF Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
Spot Euro/Swiss Franc exchange rate
1.14
1.13
1.12
1.11
1.10
1.09
1.08
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date

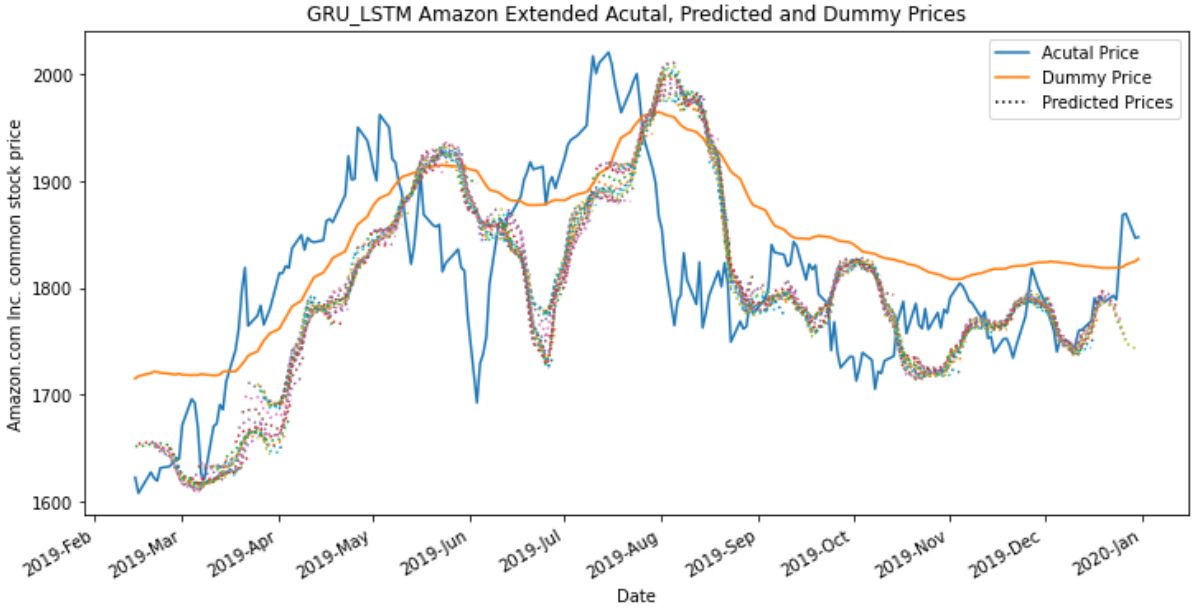

LSTM_GRU EURUSD Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
Spot Euro/US dollar exchange rate
1.14
1.13
1.12
1.11
1.10
1.09
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date

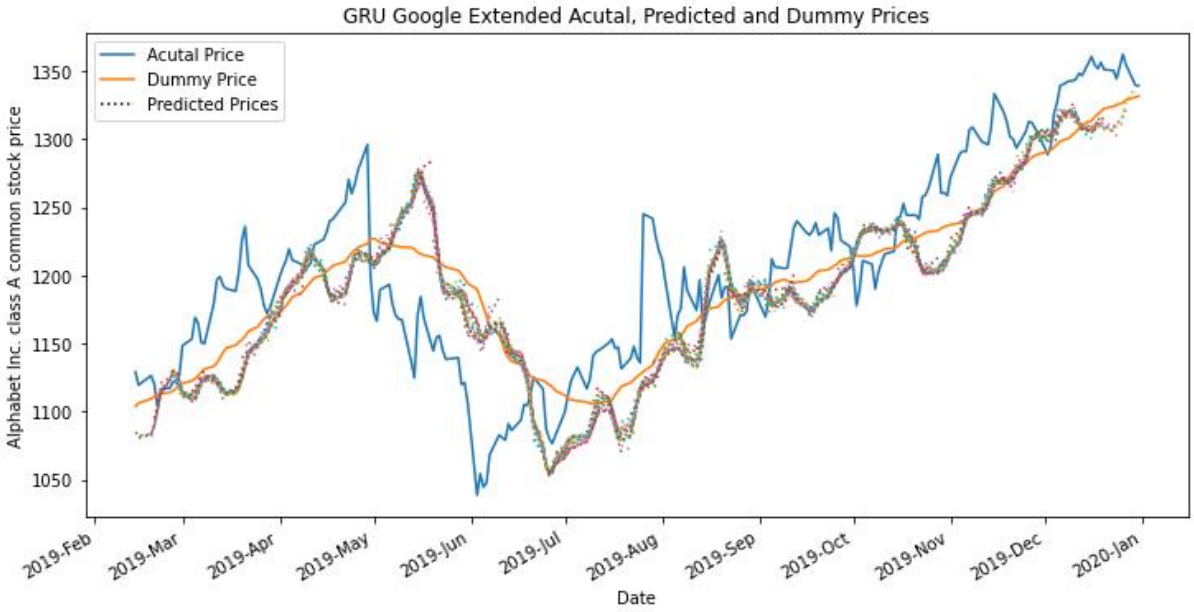

LSTM GBPUSD Extended Acutal, Predicted and Dummy Prices
Spot British Pound/US dollar exchange rate
1.34
1.32
1.30
1.28
1.26
1.24
1.22
1.20
Acutal Price
Dummy Price
Predicted Prices
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date

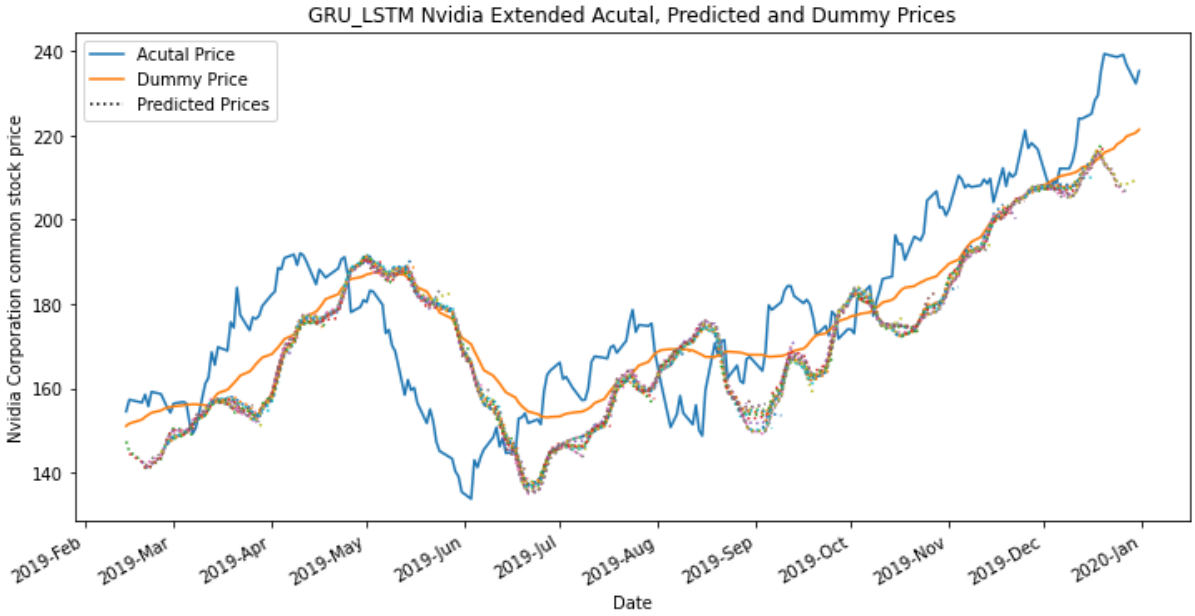

GRU_AE Bund10y Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
10y German Bund yield
0.2
0.0
−0.2
−0.4
−0.6
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date


GRU_LSTM Gilt10y Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
10y British Gilt yield
1.2
1.0
0.8
0.6
0.4
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date


LSTM_GRU Treasury10y Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
10y US Treasury yield
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date


GRU_LSTM Amazon Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
Amazon.com Inc. common stock price
2000
1900
1800
1700
1600
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date


GRU Google Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
Alphabet Inc. class A common stock price
1350
1300
1250
1200
1150
1100
1050
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date


GRU_LSTM Nvidia Extended Acutal, Predicted and Dummy Prices
Acutal Price
Dummy Price
Predicted Prices
Nvidia Corporation common stock price
240
220
200
180
160
140
2019-Feb
2019-Mar
2019-Apr
2019-May
2019-Jun
2019-Jul
2019-Aug
2019-Sep
2019-Oct
2019-Nov
2019-Dec
2020-Jan
Date

Appendix 7: Security Rankings

Security	Model	MSE	MAE	RMSE	Score
Al	LSTM_GRU	0.003619	0.0474	0.060157	1
Al	GRU_AE	0.003719	0.048412	0.060985	2
Al	GRU	0.003817	0.049056	0.061785	3
Al	MLP	0.003844	0.049956	0.062003	4
Al	MLP_AE	0.003956	0.050355	0.062897	5
Al	LSTM	0.004005	0.049639	0.063286	6
Al	GRU_LSTM	0.004271	0.052284	0.065356	7
Al	CNN	0.004282	0.051775	0.065435	8
Al	LSTM_AE	0.004457	0.052758	0.066762	9
Al	CNN_LSTM	0.007341	0.066814	0.085679	10
Al	CNN_GRU	0.007704	0.070362	0.08777	11
Al	Dummy	0.019335	0.116165	0.13905	12

Security	Model	MSE	MAE	RMSE	Score
Amazon	GRU_LSTM	0.002239	0.034979	0.047322	1
Amazon	GRU	0.002329	0.035903	0.04826	2
Amazon	GRU_AE	0.002344	0.037674	0.04841	3
Amazon	LSTM_GRU	0.002523	0.037115	0.050234	4
Amazon	LSTM_AE	0.002684	0.041261	0.051804	5
Amazon	CNN	0.002829	0.041323	0.053184	6
Amazon	MLP	0.003265	0.043473	0.057139	7
Amazon	LSTM	0.003324	0.042733	0.057657	8
Amazon	CNN_LSTM	0.004882	0.05342	0.069871	9
Amazon	CNN_GRU	0.005364	0.05772	0.073239	10
Amazon	MLP_AE	0.00884	0.073717	0.094019	11
Amazon	Dummy	0.034129	0.154633	0.18474	12

Security	Model	MSE	MAE	RMSE	Score
Bund10y	GRU_AE	0.000926	0.023728	0.030425	1
Bund10y	MLP	0.001024	0.024575	0.032007	2
Bund10y	LSTM_AE	0.001069	0.025644	0.032694	3
Bund10y	LSTM_GRU	0.001071	0.025267	0.03272	4
Bund10y	GRU_LSTM	0.001109	0.027032	0.0333	5
Bund10y	GRU	0.001162	0.027262	0.034085	6
Bund10y	LSTM	0.001273	0.028714	0.035682	7
Bund10y	MLP_AE	0.001553	0.030137	0.039414	8
Bund10y	CNN_GRU	0.001824	0.033069	0.042712	9
Bund10y	CNN	0.001973	0.0367	0.044422	10
Bund10y	CNN_LSTM	0.002084	0.036352	0.045656	11
Bund10y	Dummy	0.011925	0.088719	0.109202	12

Security	Model	MSE	MAE	RMSE	Score
Corn	GRU	0.003249	0.04053	0.056996	1
Corn	LSTM_GRU	0.003417	0.04121	0.058451	2
Corn	MLP	0.003452	0.041842	0.058751	3
Corn	GRU_LSTM	0.003472	0.041208	0.058925	4
Corn	LSTM	0.00356	0.042177	0.059668	5
Corn	MLP_AE	0.003571	0.043118	0.059757	6
Corn	LSTM_AE	0.003585	0.042656	0.059879	7
Corn	GRU_AE	0.004501	0.051096	0.067092	8
Corn	CNN	0.004581	0.048404	0.067683	9
Corn	CNN_GRU	0.007381	0.063775	0.085911	10
Corn	CNN_LSTM	0.007593	0.064278	0.087137	11
Corn	Dummy	0.041447	0.151605	0.203585	12

Security	Model	MSE	MAE	RMSE	Score
Cu	LSTM_AE	0.003	0.042771	0.054775	1
Cu	GRU_AE	0.003008	0.043516	0.054844	2
Cu	GRU_LSTM	0.003014	0.043444	0.054896	3
Cu	LSTM_GRU	0.003088	0.043208	0.055566	4
Cu	MLP_AE	0.003325	0.045312	0.057662	5
Cu	GRU	0.003451	0.047591	0.058741	6
Cu	LSTM	0.003458	0.047292	0.058807	7
Cu	CNN	0.0036	0.047846	0.060004	8
Cu	MLP	0.004758	0.054926	0.068977	9
Cu	CNN_LSTM	0.005842	0.059432	0.076432	10
Cu	CNN_GRU	0.006104	0.060759	0.078131	11
Cu	Dummy	0.028899	0.132173	0.169997	12

Security	Model	MSE	MAE	RMSE	Score
EURCHF	GRU_AE	0.00189	0.033785	0.04348	1
EURCHF	LSTM_GRU	0.00207	0.035561	0.045495	2
EURCHF	GRU_LSTM	0.002328	0.038214	0.048252	3
EURCHF	GRU	0.00241	0.039341	0.049093	4
EURCHF	LSTM_AE	0.002736	0.040903	0.052309	5
EURCHF	CNN	0.002924	0.04302	0.054078	6
EURCHF	MLP_AE	0.002997	0.042516	0.054743	7
EURCHF	LSTM	0.003026	0.043352	0.055006	8
EURCHF	CNN_LSTM	0.004222	0.052007	0.064979	9
EURCHF	MLP	0.004309	0.05003	0.065642	10
EURCHF	CNN_GRU	0.004397	0.052347	0.066309	11
EURCHF	Dummy	0.016968	0.10864	0.130262	12

Security	Model	MSE	MAE	RMSE	Score

EURUSD	LSTM_GRU	0.002662	0.039449	0.051592	1
EURUSD	GRU_LSTM	0.002684	0.039162	0.051803	2
EURUSD	GRU	0.00277	0.039854	0.052631	3
EURUSD	GRU_AE	0.002799	0.0406	0.052909	4
EURUSD	CNN	0.002823	0.041446	0.053128	5
EURUSD	MLP	0.002828	0.041666	0.053181	6
EURUSD	LSTM_AE	0.002982	0.042022	0.05461	7
EURUSD	LSTM	0.003673	0.04679	0.060608	8
EURUSD	MLP_AE	0.004121	0.050867	0.064191	9
EURUSD	CNN_GRU	0.005758	0.05989	0.075883	10
EURUSD	CNN_LSTM	0.005811	0.060148	0.07623	11
EURUSD	Dummy	0.023689	0.119703	0.153911	12

Security	Model	MSE	MAE	RMSE	Score
GBPUSD	LSTM	0.002516	0.036615	0.05016	1
GBPUSD	LSTM_GRU	0.002553	0.037499	0.050528	2
GBPUSD	GRU_LSTM	0.002556	0.037807	0.050553	3
GBPUSD	MLP	0.002642	0.038993	0.051399	4
GBPUSD	GRU	0.002909	0.041625	0.053939	5
GBPUSD	MLP_AE	0.003023	0.041234	0.054978	6
GBPUSD	CNN	0.003404	0.042789	0.058341	7
GBPUSD	LSTM_AE	0.00342	0.044443	0.05848	8
GBPUSD	GRU_AE	0.00353	0.043521	0.059413	9
GBPUSD	CNN_GRU	0.005522	0.055139	0.074308	10
GBPUSD	CNN_LSTM	0.006242	0.058181	0.079007	11
GBPUSD	Dummy	0.021934	0.119493	0.148101	12

Security	Model	MSE	MAE	RMSE	Score
Gilt10y	GRU_LSTM	0.00174	0.032736	0.041712	1
Gilt10y	GRU	0.001749	0.033293	0.041823	2
Gilt10y	LSTM_AE	0.001806	0.033712	0.042501	3
Gilt10y	CNN	0.001809	0.033311	0.042531	4
Gilt10y	LSTM_GRU	0.001941	0.03439	0.044058	5
Gilt10y	MLP_AE	0.002039	0.035479	0.045157	6
Gilt10y	GRU_AE	0.002122	0.036199	0.046068	7
Gilt10y	LSTM	0.002128	0.036908	0.046134	8
Gilt10y	MLP	0.002279	0.038613	0.047734	9
Gilt10y	CNN_GRU	0.003639	0.04503	0.060323	10
Gilt10y	CNN_LSTM	0.004042	0.047708	0.063574	11
Gilt10y	Dummy	0.013892	0.09978	0.117863	12

Security	Model	MSE	MAE	RMSE	Score
Google	GRU	0.002602	0.034234	0.051013	1

Google	GRU_LSTM	0.002627	0.034244	0.05125	2
Google	GRU_AE	0.002627	0.034243	0.051254	3
Google	LSTM_AE	0.002874	0.037039	0.053609	4
Google	LSTM_GRU	0.003053	0.039521	0.055251	5
Google	CNN	0.003091	0.038765	0.0556	6
Google	MLP_AE	0.003189	0.03906	0.056475	7
Google	MLP	0.003212	0.039945	0.056679	8
Google	LSTM	0.003404	0.041219	0.058341	9
Google	CNN_LSTM	0.005741	0.055641	0.075771	10
Google	CNN_GRU	0.006321	0.056461	0.079504	11
Google	Dummy	0.017869	0.107493	0.133676	12

Security	Model	MSE	MAE	RMSE	Score
Nvidia	GRU_LSTM	0.001341	0.027575	0.036617	1
Nvidia	GRU	0.001371	0.028062	0.037027	2
Nvidia	GRU_AE	0.001522	0.029688	0.03901	3
Nvidia	LSTM	0.001718	0.031567	0.041453	4
Nvidia	CNN	0.00175	0.031841	0.041832	5
Nvidia	MLP	0.001751	0.032292	0.041849	6
Nvidia	LSTM_GRU	0.002004	0.035316	0.044768	7
Nvidia	MLP_AE	0.002345	0.038853	0.048421	8
Nvidia	CNN_LSTM	0.002951	0.042897	0.054322	9
Nvidia	CNN_GRU	0.003424	0.046397	0.058518	10
Nvidia	LSTM_AE	0.005039	0.057601	0.070989	11
Nvidia	Dummy	0.016238	0.103206	0.127429	12

Security	Model	MSE	MAE	RMSE	Score
Treasury10y	LSTM_GRU	0.001056	0.024796	0.032497	1
Treasury10y	GRU_AE	0.001066	0.02515	0.032642	2
Treasury10y	CNN	0.001113	0.025451	0.033363	3
Treasury10y	GRU_LSTM	0.00115	0.026558	0.033905	4
Treasury10y	LSTM	0.001199	0.026206	0.034631	5
Treasury10y	GRU	0.001261	0.027422	0.035505	6
Treasury10y	MLP	0.001414	0.029093	0.037607	7
Treasury10y	MLP_AE	0.001517	0.031135	0.038943	8
Treasury10y	LSTM_AE	0.001563	0.030144	0.03953	9
Treasury10y	CNN_GRU	0.002303	0.036542	0.047994	10
Treasury10y	CNN_LSTM	0.002611	0.038968	0.051095	11
Treasury10y	Dummy	0.011908	0.082874	0.109124	12

Appendix 7: Model Rankings

Model	Security	MSE	MAE	RMSE	Score
CNN	Treasury10y	0.001113	0.025451	0.033363	1
CNN	Nvidia	0.00175	0.031841	0.041832	2
CNN	Gilt10y	0.001809	0.033311	0.042531	3
CNN	Bund10y	0.001973	0.0367	0.044422	4
CNN	EURUSD	0.002823	0.041446	0.053128	5
CNN	Amazon	0.002829	0.041323	0.053184	6
CNN	EURCHF	0.002924	0.04302	0.054078	7
CNN	Google	0.003091	0.038765	0.0556	8
CNN	GBPUSD	0.003404	0.042789	0.058341	9
CNN	Cu	0.0036	0.047846	0.060004	10
CNN	Al	0.004282	0.051775	0.065435	11
CNN	Corn	0.004581	0.048404	0.067683	12

Model	Security	MSE	MAE	RMSE	Score
CNN_GRU	Bund10y	0.001824	0.033069	0.042712	1
CNN_GRU	Treasury10y	0.002303	0.036542	0.047994	2
CNN_GRU	Nvidia	0.003424	0.046397	0.058518	3
CNN_GRU	Gilt10y	0.003639	0.04503	0.060323	4
CNN_GRU	EURCHF	0.004397	0.052347	0.066309	5
CNN_GRU	Amazon	0.005364	0.05772	0.073239	6
CNN_GRU	GBPUSD	0.005522	0.055139	0.074308	7
CNN_GRU	EURUSD	0.005758	0.05989	0.075883	8
CNN_GRU	Cu	0.006104	0.060759	0.078131	9
CNN_GRU	Google	0.006321	0.056461	0.079504	10
CNN_GRU	Corn	0.007381	0.063775	0.085911	11
CNN_GRU	Al	0.007704	0.070362	0.08777	12

Model	Security	MSE	MAE	RMSE	Score
CNN_LSTM	Bund10y	0.002084	0.036352	0.045656	1
CNN_LSTM	Treasury10y	0.002611	0.038968	0.051095	2
CNN_LSTM	Nvidia	0.002951	0.042897	0.054322	3
CNN_LSTM	Gilt10y	0.004042	0.047708	0.063574	4
CNN_LSTM	EURCHF	0.004222	0.052007	0.064979	5
CNN_LSTM	Amazon	0.004882	0.05342	0.069871	6
CNN_LSTM	Google	0.005741	0.055641	0.075771	7
CNN_LSTM	EURUSD	0.005811	0.060148	0.07623	8
CNN_LSTM	Cu	0.005842	0.059432	0.076432	9
CNN_LSTM	GBPUSD	0.006242	0.058181	0.079007	10
CNN_LSTM	Al	0.007341	0.066814	0.085679	11
CNN_LSTM	Corn	0.007593	0.064278	0.087137	12

Model	Security	MSE	MAE	RMSE	Score
Dummy	Treasury10y	0.011908	0.082874	0.109124	1
Dummy	Bund10y	0.011925	0.088719	0.109202	2
Dummy	Gilt10y	0.013892	0.09978	0.117863	3
Dummy	Nvidia	0.016238	0.103206	0.127429	4
Dummy	EURCHF	0.016968	0.10864	0.130262	5
Dummy	Google	0.017869	0.107493	0.133676	6
Dummy	Al	0.019335	0.116165	0.13905	7
Dummy	GBPUSD	0.021934	0.119493	0.148101	8
Dummy	EURUSD	0.023689	0.119703	0.153911	9
Dummy	Cu	0.028899	0.132173	0.169997	10
Dummy	Amazon	0.034129	0.154633	0.18474	11
Dummy	Corn	0.041447	0.151605	0.203585	12

Model	Security	MSE	MAE	RMSE	Score
GRU	Bund10y	0.001162	0.027262	0.034085	1
GRU	Treasury10y	0.001261	0.027422	0.035505	2
GRU	Nvidia	0.001371	0.028062	0.037027	3
GRU	Gilt10y	0.001749	0.033293	0.041823	4
GRU	Amazon	0.002329	0.035903	0.04826	5
GRU	EURCHF	0.00241	0.039341	0.049093	6
GRU	Google	0.002602	0.034234	0.051013	7
GRU	EURUSD	0.00277	0.039854	0.052631	8
GRU	GBPUSD	0.002909	0.041625	0.053939	9
GRU	Corn	0.003249	0.04053	0.056996	10
GRU	Cu	0.003451	0.047591	0.058741	11
GRU	Al	0.003817	0.049056	0.061785	12

Model	Security	MSE	MAE	RMSE	Score
GRU_AE	Bund10y	0.000926	0.023728	0.030425	1
GRU_AE	Treasury10y	0.001066	0.02515	0.032642	2
GRU_AE	Nvidia	0.001522	0.029688	0.03901	3
GRU_AE	EURCHF	0.00189	0.033785	0.04348	4
GRU_AE	Gilt10y	0.002122	0.036199	0.046068	5
GRU_AE	Amazon	0.002344	0.037674	0.04841	6
GRU_AE	Google	0.002627	0.034243	0.051254	7
GRU_AE	EURUSD	0.002799	0.0406	0.052909	8
GRU_AE	Cu	0.003008	0.043516	0.054844	9
GRU_AE	GBPUSD	0.00353	0.043521	0.059413	10
GRU_AE	Al	0.003719	0.048412	0.060985	11
GRU_AE	Corn	0.004501	0.051096	0.067092	12

Model	Security	MSE	MAE	RMSE	Score
GRU_LSTM	Bund10y	0.001109	0.027032	0.0333	1

GRU_LSTM	Treasury10y	0.00115	0.026558	0.033905	2
GRU_LSTM	Nvidia	0.001341	0.027575	0.036617	3
GRU_LSTM	Gilt10y	0.00174	0.032736	0.041712	4
GRU_LSTM	Amazon	0.002239	0.034979	0.047322	5
GRU_LSTM	EURCHF	0.002328	0.038214	0.048252	6
GRU_LSTM	GBPUSD	0.002556	0.037807	0.050553	7
GRU_LSTM	Google	0.002627	0.034244	0.05125	8
GRU_LSTM	EURUSD	0.002684	0.039162	0.051803	9
GRU_LSTM	Cu	0.003014	0.043444	0.054896	10
GRU_LSTM	Corn	0.003472	0.041208	0.058925	11
GRU_LSTM	Al	0.004271	0.052284	0.065356	12

Model	Security	MSE	MAE	RMSE	Score
LSTM	Treasury10y	0.001199	0.026206	0.034631	1
LSTM	Bund10y	0.001273	0.028714	0.035682	2
LSTM	Nvidia	0.001718	0.031567	0.041453	3
LSTM	Gilt10y	0.002128	0.036908	0.046134	4
LSTM	GBPUSD	0.002516	0.036615	0.05016	5
LSTM	EURCHF	0.003026	0.043352	0.055006	6
LSTM	Amazon	0.003324	0.042733	0.057657	7
LSTM	Google	0.003404	0.041219	0.058341	8
LSTM	Cu	0.003458	0.047292	0.058807	9
LSTM	Corn	0.00356	0.042177	0.059668	10
LSTM	EURUSD	0.003673	0.04679	0.060608	11
LSTM	Al	0.004005	0.049639	0.063286	12

Model	Security	MSE	MAE	RMSE	Score
LSTM_AE	Bund10y	0.001069	0.025644	0.032694	1
LSTM_AE	Treasury10y	0.001563	0.030144	0.03953	2
LSTM_AE	Gilt10y	0.001806	0.033712	0.042501	3
LSTM_AE	Amazon	0.002684	0.041261	0.051804	4
LSTM_AE	EURCHF	0.002736	0.040903	0.052309	5
LSTM_AE	Google	0.002874	0.037039	0.053609	6
LSTM_AE	EURUSD	0.002982	0.042022	0.05461	7
LSTM_AE	Cu	0.003	0.042771	0.054775	8
LSTM_AE	GBPUSD	0.00342	0.044443	0.05848	9
LSTM_AE	Corn	0.003585	0.042656	0.059879	10
LSTM_AE	Al	0.004457	0.052758	0.066762	11
LSTM_AE	Nvidia	0.005039	0.057601	0.070989	12

Model	Security	MSE	MAE	RMSE	Score
LSTM_GRU	Treasury10y	0.001056	0.024796	0.032497	1
LSTM_GRU	Bund10y	0.001071	0.025267	0.03272	2
LSTM_GRU	Gilt10y	0.001941	0.03439	0.044058	3

LSTM_GRU	Nvidia	0.002004	0.035316	0.044768	4
LSTM_GRU	EURCHF	0.00207	0.035561	0.045495	5
LSTM_GRU	Amazon	0.002523	0.037115	0.050234	6
LSTM_GRU	GBPUSD	0.002553	0.037499	0.050528	7
LSTM_GRU	EURUSD	0.002662	0.039449	0.051592	8
LSTM_GRU	Google	0.003053	0.039521	0.055251	9
LSTM_GRU	Cu	0.003088	0.043208	0.055566	10
LSTM_GRU	Corn	0.003417	0.04121	0.058451	11
LSTM_GRU	Al	0.003619	0.0474	0.060157	12

Model	Security	MSE	MAE	RMSE	Score
MLP	Bund10y	0.001024	0.024575	0.032007	1
MLP	Treasury10y	0.001414	0.029093	0.037607	2
MLP	Nvidia	0.001751	0.032292	0.041849	3
MLP	Gilt10y	0.002279	0.038613	0.047734	4
MLP	GBPUSD	0.002642	0.038993	0.051399	5
MLP	EURUSD	0.002828	0.041666	0.053181	6
MLP	Google	0.003212	0.039945	0.056679	7
MLP	Amazon	0.003265	0.043473	0.057139	8
MLP	Corn	0.003452	0.041842	0.058751	9
MLP	Al	0.003844	0.049956	0.062003	10
MLP	EURCHF	0.004309	0.05003	0.065642	11
MLP	Cu	0.004758	0.054926	0.068977	12

Model	Security	MSE	MAE	RMSE	Score
MLP_AE	Treasury10y	0.001517	0.031135	0.038943	1
MLP_AE	Bund10y	0.001553	0.030137	0.039414	2
MLP_AE	Gilt10y	0.002039	0.035479	0.045157	3
MLP_AE	Nvidia	0.002345	0.038853	0.048421	4
MLP_AE	EURCHF	0.002997	0.042516	0.054743	5
MLP_AE	GBPUSD	0.003023	0.041234	0.054978	6
MLP_AE	Google	0.003189	0.03906	0.056475	7
MLP_AE	Cu	0.003325	0.045312	0.057662	8
MLP_AE	Corn	0.003571	0.043118	0.059757	9
MLP_AE	Al	0.003956	0.050355	0.062897	10
MLP_AE	EURUSD	0.004121	0.050867	0.064191	11
MLP_AE	Amazon	0.00884	0.073717	0.094019	12

Appendix 8: Full model outputs

Model	Security	Epochs	Trainable Parameters	Activation	Optimizer	Batch Size	Learning Rate	MSE	RMSE	MAE
CNN	Al	200	17729	relu	adam	32	0.001	0.004282	0.065435	0.051775
CNN	Cu	50	17729	selu	rmsprop	32	0.0005	0.0036	0.060004	0.047846
CNN	Corn	200	17729	selu	nadam	32	0.001	0.004581	0.067683	0.048404
CNN	EURCHF	200	17729	selu	adam	64	0.001	0.002924	0.054078	0.04302

CNN	EURUSD	100	17729	relu	adam	32	0.001	0.002823	0.053128	0.041446
CNN	GBPUSD	50	17921	selu	adam	32	0.0005	0.003404	0.058341	0.042789
CNN	Gilt10y	200	17729	selu	nadam	64	0.001	0.001809	0.042531	0.033311
CNN	Bund10y	50	17729	selu	nadam	32	0.005	0.001973	0.044422	0.0367
CNN	Treasury10y	200	17729	selu	adam	128	0.001	0.001113	0.033363	0.025451
CNN	Amazon	200	17729	selu	nadam	64	0.001	0.002829	0.053184	0.041323
CNN	Google	200	17729	selu	adam	128	0.001	0.003091	0.0556	0.038765
CNN	Nvidia	200	17729	selu	adam	128	0.001	0.00175	0.041832	0.031841
CNN_GRU	Al	100	738561	selu	adam	128	0.005	0.007704	0.08777	0.070362
CNN_GRU	Cu	50	738561	selu	adam	64	0.001	0.006104	0.078131	0.060759
CNN_GRU	Corn	100	738561	relu	adam	128	0.001	0.007381	0.085911	0.063775
CNN_GRU	EURCHF	50	738561	selu	adam	64	0.001	0.004397	0.066309	0.052347
CNN_GRU	EURUSD	200	738561	selu	adam	32	0.0005	0.005758	0.075883	0.05989
CNN_GRU	GBPUSD	50	738561	selu	adam	64	0.005	0.005522	0.074308	0.055139
CNN_GRU	Gilt10y	100	738561	selu	adam	128	0.005	0.003639	0.060323	0.04503
CNN_GRU	Bund10y	50	738561	selu	adam	64	0.005	0.001824	0.042712	0.033069
CNN_GRU	Treasury10y	100	738561	selu	adam	64	0.0005	0.002303	0.047994	0.036542
CNN_GRU	Amazon	50	738561	selu	adam	32	0.0005	0.005364	0.073239	0.05772
CNN_GRU	Google	100	738561	selu	rmsprop	32	0.0005	0.006321	0.079504	0.056461
CNN_GRU	Nvidia	50	738561	selu	adam	32	0.0005	0.003424	0.058518	0.046397
CNN_LSTM	Al	50	984065	selu	adam	32	0.001	0.007341	0.085679	0.066814
CNN_LSTM	Cu	200	984065	selu	adam	32	0.001	0.005842	0.076432	0.059432
CNN_LSTM	Corn	100	984065	relu	adam	32	0.0005	0.007593	0.087137	0.064278
CNN_LSTM	EURCHF	100	984065	selu	adam	32	0.005	0.004222	0.064979	0.052007
CNN_LSTM	EURUSD	50	984065	selu	adam	32	0.001	0.005811	0.07623	0.060148
CNN_LSTM	GBPUSD	50	984065	selu	adam	64	0.005	0.006242	0.079007	0.058181
CNN_LSTM	Gilt10y	200	984065	selu	adam	32	0.005	0.004042	0.063574	0.047708
CNN_LSTM	Bund10y	100	984065	selu	adam	64	0.0005	0.002084	0.045656	0.036352
CNN_LSTM	Treasury10y	50	984065	selu	adam	32	0.001	0.002611	0.051095	0.038968
CNN_LSTM	Amazon	200	984065	selu	adam	64	0.005	0.004882	0.069871	0.05342
CNN_LSTM	Google	100	984065	selu	adam	32	0.001	0.005741	0.075771	0.055641
CNN_LSTM	Nvidia	200	984065	selu	nadam	64	0.0005	0.002951	0.054322	0.042897
GRU	Al	200	23301	tanh	adam	32	0.0005	0.003817	0.061785	0.049056
GRU	Cu	200	23301	tanh	adam	64	0.005	0.003451	0.058741	0.047591
GRU	Corn	100	40451	selu	adam	64	0.005	0.003249	0.056996	0.04053
GRU	EURCHF	200	40451	selu	Nadam	32	0.001	0.00241	0.049093	0.039341
GRU	EURUSD	200	53751	tanh	Nadam	32	0.0005	0.00277	0.052631	0.039854
GRU	GBPUSD	50	91601	tanh	Nadam	32	0.005	0.002909	0.053939	0.041625
GRU	Gilt10y	200	53751	tanh	Nadam	32	0.0005	0.001749	0.041823	0.033293
GRU	Bund10y	200	53651	tanh	Nadam	32	0.001	0.001162	0.034085	0.027262
GRU	Treasury10y	200	53751	selu	adam	32	0.001	0.001261	0.035505	0.027422
GRU	Amazon	200	23301	tanh	adam	32	0.001	0.002329	0.04826	0.035903
GRU	Google	200	91601	selu	adam	64	0.001	0.002602	0.051013	0.034234
GRU	Nvidia	100	91601	selu	adam	64	0.001	0.001371	0.037027	0.028062
GRU_AE	Al	100	187137	tanh	adam	32	0.0005	0.003719	0.060985	0.048412

GRU_AE	Cu	200	187137	tanh	nadam	64	0.001	0.003008	0.054844	0.043516
GRU_AE	Corn	100	187137	relu	rmsprop	64	0.005	0.004501	0.067092	0.051096
GRU_AE	EURCHF	100	187137	tanh	rmsprop	32	0.0005	0.00189	0.04348	0.033785
GRU_AE	EURUSD	50	187137	relu	nadam	32	0.005	0.002799	0.052909	0.0406
GRU_AE	GBPUSD	100	187137	tanh	adam	32	0.005	0.00353	0.059413	0.043521
GRU_AE	Gilt10y	200	187137	relu	nadam	64	0.005	0.002122	0.046068	0.036199
GRU_AE	Bund10y	200	187137	tanh	adam	128	0.0005	0.000926	0.030425	0.023728
GRU_AE	Treasury10y	100	187137	tanh	adam	32	0.0005	0.001066	0.032642	0.02515
GRU_AE	Amazon	200	187137	tanh	nadam	32	0.0005	0.002344	0.04841	0.037674
GRU_AE	Google	100	187137	tanh	adam	32	0.0005	0.002627	0.051254	0.034243
GRU_AE	Nvidia	100	187137	tanh	adam	32	0.001	0.001522	0.03901	0.029688
GRU_LSTM	Al	100	45953	tanh	rmsprop	32	0.001	0.004271	0.065356	0.052284
GRU_LSTM	Cu	200	45953	selu	adam	128	0.005	0.003014	0.054896	0.043444
GRU_LSTM	Corn	100	45953	selu	adam	32	0.005	0.003472	0.058925	0.041208
GRU_LSTM	EURCHF	200	45953	tanh	adam	64	0.001	0.002328	0.048252	0.038214
GRU_LSTM	EURUSD	200	45953	selu	adam	32	0.0005	0.002684	0.051803	0.039162
GRU_LSTM	GBPUSD	200	45953	selu	adam	32	0.001	0.002556	0.050553	0.037807
GRU_LSTM	Gilt10y	50	45953	tanh	adam	32	0.005	0.00174	0.041712	0.032736
GRU_LSTM	Bund10y	200	45953	selu	nadam	32	0.001	0.001109	0.0333	0.027032
GRU_LSTM	Treasury10y	100	45953	selu	nadam	32	0.001	0.00115	0.033905	0.026558
GRU_LSTM	Amazon	200	45953	selu	adam	32	0.0005	0.002239	0.047322	0.034979
GRU_LSTM	Google	200	45953	tanh	nadam	32	0.0005	0.002627	0.05125	0.034244
GRU_LSTM	Nvidia	200	45953	tanh	nadam	32	0.001	0.001341	0.036617	0.027575
LSTM	Al	100	63626	tanh	adam	64	0.005	0.004005	0.063286	0.049639
LSTM	Cu	200	111451	tanh	adam	64	0.005	0.003458	0.058807	0.047292
LSTM	Corn	200	58351	tanh	nadam	32	0.0005	0.00356	0.059668	0.042177
LSTM	EURCHF	50	63626	tanh	adam	32	0.005	0.003026	0.055006	0.043352
LSTM	EURUSD	100	231701	tanh	adam	64	0.005	0.003673	0.060608	0.04679
LSTM	GBPUSD	100	282101	tanh	nadam	32	0.005	0.002516	0.05016	0.036615
LSTM	Gilt10y	200	71051	tanh	rmsprop	32	0.0005	0.002128	0.046134	0.036908
LSTM	Bund10y	200	63626	tanh	rmsprop	64	0.001	0.001273	0.035682	0.028714
LSTM	Treasury10y	200	18026	tanh	nadam	32	0.001	0.001199	0.034631	0.026206
LSTM	Amazon	200	282101	tanh	adam	32	0.001	0.003324	0.057657	0.042733
LSTM	Google	200	111451	tanh	nadam	128	0.005	0.003404	0.058341	0.041219
LSTM	Nvidia	100	111451	tanh	adam	32	0.005	0.001718	0.041453	0.031567
LSTM_AE	Al	200	247937	tanh	nadam	64	0.001	0.004457	0.066762	0.052758
LSTM_AE	Cu	100	247937	tanh	adam	32	0.005	0.003	0.054775	0.042771
LSTM_AE	Corn	200	247937	tanh	nadam	32	0.005	0.003585	0.059879	0.042656
LSTM_AE	EURCHF	100	247937	tanh	nadam	64	0.005	0.002736	0.052309	0.040903
LSTM_AE	EURUSD	100	247937	tanh	adam	32	0.001	0.002982	0.05461	0.042022
LSTM_AE	GBPUSD	100	247937	tanh	nadam	32	0.0005	0.00342	0.05848	0.044443
LSTM_AE	Gilt10y	200	247937	tanh	nadam	128	0.005	0.001806	0.042501	0.033712
LSTM_AE	Bund10y	200	247937	tanh	nadam	32	0.001	0.001069	0.032694	0.025644
LSTM_AE	Treasury10y	100	247937	tanh	adam	32	0.0005	0.001563	0.03953	0.030144
LSTM_AE	Amazon	200	247937	tanh	adam	32	0.005	0.002684	0.051804	0.041261

LSTM_AE	Google	100	247937	tanh	nadam	32	0.005	0.002874	0.053609	0.037039
LSTM_AE	Nvidia	50	247937	tanh	rmsprop	32	0.0005	0.005039	0.070989	0.057601
LSTM_GRU	Al	100	41921	selu	adam	32	0.001	0.003619	0.060157	0.0474
LSTM_GRU	Cu	200	41921	selu	adam	64	0.005	0.003088	0.055566	0.043208
LSTM_GRU	Corn	100	41921	tanh	nadam	32	0.005	0.003417	0.058451	0.04121
LSTM_GRU	EURCHF	50	41921	tanh	adam	32	0.005	0.00207	0.045495	0.035561
LSTM_GRU	EURUSD	200	41921	selu	nadam	32	0.0005	0.002662	0.051592	0.039449
LSTM_GRU	GBPUSD	200	41921	tanh	nadam	32	0.001	0.002553	0.050528	0.037499
LSTM_GRU	Gilt10y	200	41921	selu	adam	32	0.005	0.001941	0.044058	0.03439
LSTM_GRU	Bund10y	200	41921	tanh	rmsprop	32	0.001	0.001071	0.03272	0.025267
LSTM_GRU	Treasury10y	200	41921	tanh	adam	32	0.001	0.001056	0.032497	0.024796
LSTM_GRU	Amazon	200	41921	tanh	adam	32	0.001	0.002523	0.050234	0.037115
LSTM_GRU	Google	100	41921	selu	nadam	32	0.001	0.003053	0.055251	0.039521
LSTM_GRU	Nvidia	100	41921	tanh	nadam	64	0.005	0.002004	0.044768	0.035316
MLP	Al	100	44705	selu	adam	32	0.005	0.003844	0.062003	0.049956
MLP	Cu	50	44705	selu	adam	32	0.005	0.004758	0.068977	0.054926
MLP	Corn	100	44705	selu	adam	64	0.005	0.003452	0.058751	0.041842
MLP	EURCHF	100	44705	relu	adam	32	0.005	0.004309	0.065642	0.05003
MLP	EURUSD	200	44705	selu	adam	32	0.0005	0.002828	0.053181	0.041666
MLP	GBPUSD	200	44705	selu	adam	32	0.0005	0.002642	0.051399	0.038993
MLP	Gilt10y	100	44705	relu	adam	32	0.0005	0.002279	0.047734	0.038613
MLP	Bund10y	100	44705	selu	adam	64	0.001	0.001024	0.032007	0.024575
MLP	Treasury10y	100	44705	selu	adam	64	0.005	0.001414	0.037607	0.029093
MLP	Amazon	100	44705	selu	adam	64	0.001	0.003265	0.057139	0.043473
MLP	Google	100	44705	selu	adam	128	0.005	0.003212	0.056679	0.039945
MLP	Nvidia	100	44705	selu	adam	128	0.001	0.001751	0.041849	0.032292
MLP_AE	Al	100	88769	relu	adam	32	0.0005	0.003956	0.062897	0.050355
MLP_AE	Cu	100	88769	relu	adam	32	0.001	0.003325	0.057662	0.045312
MLP_AE	Corn	50	88769	selu	adam	64	0.0005	0.003571	0.059757	0.043118
MLP_AE	EURCHF	100	88769	selu	adam	64	0.0005	0.002997	0.054743	0.042516
MLP_AE	EURUSD	100	88769	relu	adam	128	0.0005	0.004121	0.064191	0.050867
MLP_AE	GBPUSD	100	88769	relu	adam	64	0.001	0.003023	0.054978	0.041234
MLP_AE	Gilt10y	50	88769	relu	adam	32	0.005	0.002039	0.045157	0.035479
MLP_AE	Bund10y	50	88769	relu	adam	32	0.005	0.001553	0.039414	0.030137
MLP_AE	Treasury10y	100	88769	selu	adam	64	0.0005	0.001517	0.038943	0.031135
MLP_AE	Amazon	100	88769	selu	adam	64	0.0005	0.00884	0.094019	0.073717
MLP_AE	Google	100	88769	selu	adam	128	0.0005	0.003189	0.056475	0.03906
MLP_AE	Nvidia	100	88769	selu	adam	64	0.001	0.002345	0.048421	0.038853
Dummy	Al	n/a	n/a	n/a	n/a	n/a	n/a	0.019335	0.13905	0.116165
Dummy	Cu	n/a	n/a	n/a	n/a	n/a	n/a	0.028899	0.169997	0.132173
Dummy	Corn	n/a	n/a	n/a	n/a	n/a	n/a	0.041447	0.203585	0.151605
Dummy	EURCHF	n/a	n/a	n/a	n/a	n/a	n/a	0.016968	0.130262	0.10864
Dummy	EURUSD	n/a	n/a	n/a	n/a	n/a	n/a	0.023689	0.153911	0.119703
Dummy	GBPUSD	n/a	n/a	n/a	n/a	n/a	n/a	0.021934	0.148101	0.119493
Dummy	Gilt10y	n/a	n/a	n/a	n/a	n/a	n/a	0.013892	0.117863	0.09978

Dummy	Bund10y	n/a	n/a	n/a	n/a	n/a	n/a	0.011925	0.109202	0.088719
Dummy	Treasury10y	n/a	n/a	n/a	n/a	n/a	n/a	0.011908	0.109124	0.082874
Dummy	Amazon	n/a	n/a	n/a	n/a	n/a	n/a	0.034129	0.18474	0.154633
Dummy	Google	n/a	n/a	n/a	n/a	n/a	n/a	0.017869	0.133676	0.107493
Dummy	Nvidia	n/a	n/a	n/a	n/a	n/a	n/a	0.016238	0.127429	0.103206

Appendix 9: Full extended model outputs

Model	Security	MSE	RMSE	MAE
LSTM_GRU	Al	0.00124051	0.03522093	0.02825647
LSTM_AE	Cu	0.00135829	0.03685493	0.02878544
GRU	Corn	0.00208747	0.04568883	0.03192325
GRU_AE	EURCHF	0.00113994	0.03376295	0.02659384
LSTM_GRU	EURUSD	0.00111012	0.03331839	0.02582744
LSTM	GBPUSD	0.00153209	0.03914187	0.02925361
GRU_LSTM	Gilt10y	0.00047727	0.02184658	0.01740288
GRU_AE	Bund10y	0.00036044	0.01898521	0.01485619
LSTM_GRU	Treasury10y	0.00035478	0.01883548	0.01491477
GRU_LSTM	Amazon	0.00132712	0.03642961	0.0276995
GRU	Google	0.00120894	0.03476975	0.02545886
GRU_LSTM	Nvidia	0.00060798	0.02465728	0.01941261

Appendix 10: Average model trainable parameters and points scored, ranked by number of trainable parameters

Model	Average Trainable Parameters	Points
CNN_LSTM	984065	123
CNN_GRU	738561	123
LSTM_AE	247937	72
GRU_AE	187137	45
LSTM	122380	76
MLP_AE	88769	86
GRU	53376	41
GRU_LSTM	45953	36
MLP	44705	75
LSTM_GRU	41921	38
CNN	17745	77

Appendix 11: Glossary of terms

AE	*Autoencoder*
CNN	*Convolutional neural network*
DL	*Deep learning*
DNN	*Deep neural network*
EMH	*Efficient market hypothesis*
GRU	*Gated recurrent unit*
LSTM	*Long short-term memory network*
MAE	*Mean absolute error*
MLP	*Deep multi-layer perceptron*
MLP	*Multi-layer perception*
MSE	*Mean squared error*
OTC	*Over the counter*
RELU	*Rectified linear unit*
RL	*Reinforcement learning*
RMSE	*Root mean squared error*
RMSprop	*RMS propagation*
RNN	*Recurrent neural network*
SELU	*Scaled exponential linear unit*
SGD	*Stochastic gradient descent*
SK Learn	*Sci-kit Learn*
tanh	*Hyperbolic tangent*
wandb	*Weights and Biases*

References

Bengio, Y. (2012) Deep Learning of Representations for Unsupervised and Transfer Learning. In: Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, et al. (eds.). *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. Proceedings of Machine Learning Research. [Online]. June 2012 Bellevue, Washington, USA, PMLR. pp. 17–36. Available from: http://proceedings.mlr.press/v27/bengio12a.html.

Bengio, Y., Boulanger-Lewandowski, N. & Pascanu, R. (2013) Advances in optimizing recurrent networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [Online]. 2013 pp. 8624–8628. Available from: doi:10.1109/ICASSP.2013.6639349.

Bergstra, J. & Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*. [Online] 13 (10), 281–305. Available from: http://jmlr.org/papers/v13/bergstra12a.html.

Bontempi, G., Taieb, S., & le Borgne, Y., (2013) Machine Learning Strategies for Time Series Forecasting. In: Esteban Aufaure Marie-Aude and Zimányi (ed.). *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*. [Online]. Berlin, Heidelberg, Springer Berlin Heidelberg. pp. 62–77. Available from: doi:10.1007/978-3-642-36318-4_3.

Cao, Q., Leggio, K.B. & Schniederjans, M.J. (2005) A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. *Computers & Operations Research*. [Online] 32 (10), 2499–2512. Available from: doi:https://doi.org/10.1016/j.cor.2004.03.015.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014) *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*.

Dahl, G.E., Sainath, T.N. & Hinton, G.E. (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [Online]. 2013 pp. 8609–8613. Available from: doi:10.1109/ICASSP.2013.6639346.

Deng, L. & Yu, D. (2014) Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*. [Online] 7 (3–4), 197–387. Available from: doi:10.1561/2000000039.

Fukushima, K. (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*. [Online] 36 (4), 193–202. Available from: doi:10.1007/BF00344251.

Galbraith, J.K. (1955) *The Great Crash, 1929.* Boston, Houghton Mifflin.

Gardner, M.W. & Dorling, S.R. (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*. [Online] 32 (14), 2627–2636. Available from: doi:https://doi.org/10.1016/S1352-2310(97)00447-0.

Gers, F.A., Schmidhuber, J. & Cummins, F. (1999) Learning to forget: continual prediction with LSTM. In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. [Online]. 1999 pp. 850–855 vol.2. Available from: doi:10.1049/cp:19991218.

Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. MIT Press.

Graham, Benjamin., Dodd, D.L. & Buffett, Warren. (2009) *Security analysis : principles and technique*. New York, McGraw-Hill.

Harding, D. (1994) Making Money from Mathematical Models. *Philosophical Transactions: Physical Sciences and Engineering*. [Online] 347 (1684), 511–515. Available from: http://www.jstor.org/stable/54361.

Hinton, G.E. & Salakhutdinov, R.R. (2006) Reducing the Dimensionality of Data with Neural Networks. *Science*. [Online] 313 (5786), 504–507. Available from: doi:10.1126/science.1127647.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., & Soman, K.P., (2018) NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science*. [Online] 132, 1351–1362. Available from: doi:https://doi.org/10.1016/j.procs.2018.05.050.

Hochreiter, S. & Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Comput.* [Online] 9 (8), 1735–1780. Available from: doi:10.1162/neco.1997.9.8.1735.

Hull J (2012) *Options, futures, and other derivatives*. [Online]. Eighth edition. Boston : Prentice Hall, [2012] ©2012. Available from: https://search.library.wisc.edu/catalog/9910112878402121.

Jayanth, A., Harish Ram, D.S. & Nair, B. (2018) Applicability of Deep Learning Models for Stock Price Forecasting An Empirical Study on BANKEX Data. *Procedia Computer Science*. [Online] 143, 947–953. Available from: doi:https://doi.org/10.1016/j.procs.2018.10.340.

Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. (2017) *Self-Normalizing Neural Networks*.

Lewis, M. (2014) *Flash boys: a Wall Street revolt*. First Edition. New York, W.W. Norton & Company.

Lo, A.W. (2017) *Financial Evolution at the Speed of Thought*. [Online]. Princeton University Press. Available from: doi:10.2307/j.ctvc77k3n.

Malkiel, B.G. & Fama, E.F. (1970) EFFICIENT CAPITAL MARKETS: A REVIEW OF THEORY AND EMPIRICAL WORK*. *The Journal of Finance*. [Online] 25 (2), 383–417. Available from: doi:https://doi.org/10.1111/j.1540-6261.1970.tb00518.x.

Malkiel, Burton.G. (1973) *A Random Walk Down Wall Street*. Norton, New York.

Meng, Q., Catchpoole, D., Skillicom, D. & Kennedy, P.J. (2017) Relational autoencoder for feature extraction. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. [Online]. 2017 pp. 364–371. Available from: doi:10.1109/IJCNN.2017.7965877.

Pascanu, R., Mikolov, T. & Bengio, Y. (2013) On the Difficulty of Training Recurrent Neural Networks. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. 2013 JMLR.org. pp. III–1310–III–1318.

Rezaei, H., Faaljou, H. & Mansourfar, G. (2021) Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*. [Online] 169, 114332. Available from: doi:https://doi.org/10.1016/j.eswa.2020.114332.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA, MIT Press. pp. 318–362.

Sezer, O.B., Gudelek, M.U. & Ozbayoglu, A.M. (2020) Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing*. [Online] 90, 106181. Available from: doi:https://doi.org/10.1016/j.asoc.2020.106181.

Taleb, N. (1997) *Dynamic Hedging: Managing Vanilla and Exotic Options*. 1st edition. New York, John Wiley & Sons.

Taleb, N.N. (2004) *Fooled by Randomness*. 2nd edition. New York, Texere.

Ziemba, W.T. (2020) The five investor camps that try to beat the stock market. *Borsa Istanbul Review*. [Online] 20 (4), 301–306. Available from: doi:https://doi.org/10.1016/j.bir.2020.09.008.