

Article

Not peer-reviewed version

---

# Predicting Forex Currency Fluctuations Using a Novel Bio-inspired Modular Neural Network

---

[Christos Bormpotsis](#)\*, Mohamed Sedky, [Asma Patel](#)

Posted Date: 18 August 2023

doi: 10.20944/preprints202308.1351.v1

Keywords: modular neural networks; convolutional neural networks; recurrent neural networks; rational choice theory; price fluctuations; sentiment analysis; Forex prediction



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

*Article*

# Predicting Forex Currency Fluctuations Using a Novel Bio-Inspired Modular Neural Network

Christos Bormpotsis <sup>1,3,\*</sup>, Mohamed Sedky <sup>1</sup> and Asma Patel <sup>1,2</sup>

<sup>1</sup> Staffordshire University, College Road, ST4 2DE Stoke-on-Trent, UK; mhs2@staffs.ac.uk

<sup>2</sup> Aston University, Birmingham B4 7ET, UK; a.patel122@aston.ac.uk

<sup>3</sup> CBS International Business School, Cologne, Germany

\* Correspondence: c.bormpotsis@cbs.de

**Abstract:** Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been utilised to forecast the foreign exchange market (Forex). However, such models usually exhibit unstable behaviour, as data perturbations often downgrade the functionality of the entire network due to their monolithic architecture. Hence, this study proposes a novel neuroscience-informed modular network applied to the closing prices and the sentiments retrieved from Yahoo Finance and Twitter APIs, aiming to anticipate better price fluctuations in Euro to British Pound Sterling (EUR/GBP) rather than monolithic methods. The proposed model is based on a new modular CNN, replacing pooling layers with orthogonal kernel initialisation RNNs coupled with Monte Carlo Dropout (MCD), namely MCoRNNMCD. It combines two modules: i) a convolutional simple RNN and ii) a convolutional Gated Recurrent Unit (GRU), where orthogonality and MCD are added to reduce the overfitting, assessing each module's uncertainty. These parallel feature extraction modules concatenate their outputs to a final three-layer Artificial Neural Network (ANN) decision-making module. A comprehensive comparison viewing objective evaluation metrics such as the Mean Square Error (MSE) proved that the proposed MCoRNNMCD-ANN outperformed single CNN, LSTM, GRU, and the state-of-the-art hybrid BiCuDNNLSTM, CLSTM, CNN-LSTM, and LSTM-GRU in forecasting hourly EUR/GBP closing price fluctuations.

**Keywords:** modular neural networks; convolutional neural networks; recurrent neural networks; rational choice theory; price fluctuations; sentiment analysis; Forex prediction

## 1. Introduction

The foreign exchange (Forex) market, a global and highly liquid financial market for currency exchange, plays a crucial role in international trade and investment. Its continuous operation and substantial trading volume make it an attractive choice for investors, leading to a growing number of individuals transitioning from the stock market to Forex. It substantially influences contemporary international economies concerning economic expansion, global interest rates, and financial equilibrium [1]. Researchers emphasised that due to the substantial magnitude of daily transactions, investors and financial institutions possess the potential to yield significant returns by accurately speculating and signifying fluctuations in Forex exchange rates [2]. Computational advancements, such as Artificial Intelligence (AI) and its machine and deep learning subfields, are utilised in the stock and Forex markets by providing traders with new ways to scrutinise market data and seek to find potentially profitable trading options [3,4]. However, recent AI tendencies have revealed that the synergy between neuroscience, machine and deep learning is necessary for more informed and better-comprehended decision-making [5]. Likewise, neuroscience supplemented with economic theories such as rational choice could be pivotal to developing bio-informed AI models in handling Forex's intricacies [6,7].

Rational Choice Theory (RCT) in financial markets, influencing investors' economic decision-making processes, constitutes a multifaceted cognitive phenomenon intertwined with rational self-

interest. Individuals navigate diverse financial conditions in this intricate landscape to derive optimal net benefits [8]. Furthermore, RCT illuminates how investors assimilate information, exhibit demeanours across various social and economic contexts – notably financial markets like Forex – and formulate trading strategies [9,10]. Nevertheless, while RCT underscores the centrality of rationality in decision-making, it is imperative to recognise that emotions influence investors' choices [11].

Moreover, contemporary insights from neuroscience have contributed to explicating decision-making processes by elucidating the complex connections between rational deliberation and emotional responses mediated by distinct brain regions, such as the insular and prefrontal cortex [12,13]. This emerging understanding highlights the interplay between cognitive rationality and affective elements, providing a more nuanced comprehension of how economic reasoning is constructed. Recent studies indicated that behavioural facilitation in the human brain regions, such as the amygdala and hippocampus, is related to emotions and memory retrieval [14–18]. Amygdala and hippocampus correlate with cortical areas such as the frontal and temporal lobes, including brain parts such as the striatum, insular, and prefrontal cortex [19]. Current neuroscientific investigations imply that these parts of the brain are not only accountable for the individuals' procedural learning, reasoning and emotions, considering them most likely crucial for decision-making under financial risk conditions [20–25].

AI algorithms such as Artificial Neural Networks (ANNs) have emerged as a powerful, innovative mechanism for simulating brain functions, such as self-intuition and Natural Language Processing (NLP) linked with emotions, to comprehend information processing and evaluate the possible contingencies to arrive at optimal decision prospects [26,27]. Moreover, different ANNs types, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, have been employed against traditional methods like Support Vector Machines (SVM) in contemplating the future price direction applied to a non-stationary time series. For example, Sim et al. [28] aimed to predict the Standard and Poor's (S&P) 500 index by considering the closing price and nine technical indicators, including SMA, EMA, ROC, MACD, Fast %K, Slow %D, Upper Band, and Lower Band. In their investigation, a comparison was made between three models: CNN, ANN, and SVM. The researchers concluded that technical indicators were not suitable as input features due to their similarity in behaviour to the moving pattern of the closing price, which resulted in poor performance. Moreover, CNN outperformed ANN and SVM without utilising technical indicators.

Similarly, Lanbouri and Achchab [29] presented a study focusing on predicting the price of Amazon stock using LSTM networks and technical indicators. They conducted two experiments to evaluate the LSTM's performance. The first experiment excluded technical indicators and utilised only the Open, High, Low, and Close (OHLC) prices and volume as input features. The second experiment incorporated five technical indicators (EMA12, EMA25, MACD, Bollinger Up, and Bollinger Down) along with the OHLC prices and volume. Interestingly, their findings indicated that accurate predictions of the closing price could be achieved without the use of technical indicators.

Although ANNs and their advanced techniques, such as CNN and LSTM, have demonstrated the capacity to recognise financial market patterns and trends, their monolithic architectures pose significant challenges such as:

- Limited scalability and Lack of flexibility: Monolithic architectures may be more challenging to scale because they are not easily divided into shorter, independent modules that can be developed and added to the architecture as needed [30].
- Difficulty understanding and modifying the architecture: Monolithic architectures can be challenging to understand, maintain, and modify, especially as their size becomes more extensive. Thus, updating the architecture as data or market conditions can be challenging [31].
- Increased risk of failure: Because monolithic architectures are difficult to understand and modify, there is an increased risk of failure when making changes to the architecture. Hence, fixing it can be computationally costly and time-consuming [32].

The limitations mentioned above result in poor model performance and may increase prediction errors when confronted with even minor changes in the data occurring at a national or global scale [33]. Thus, this study aims to revise the existing monolithic computational models investigating

neural networks to conceivably enhance Forex market price fluctuations forecasts. More specifically, based on recent neuroscientific advancements is critical to comprehend investors' decision-making sufficiently in an effort to improve monolithic architectures [34].

A modular neural model for potentially foreseeing Forex market price fluctuations is proposed in this study to deal with the limitations of existing monolithic approaches. More specifically, the contributions of this study can be described as follows:

1. A novel modular neural network inspired by cognitive neuroscience and RCT is proposed to model human decision-making, enhancing Forex market predictions.
2. A new adaptation mechanism consists of Monte Carlo dropout and orthogonal kernel initialisation, incorporating it into recurrent layers within a convolutional modular network, replacing the standard pooling layer of a typical and conventional CNN.
3. A comparative analysis of the proposed modular network with state-of-the-art hybrid and single monolithic models demonstrates the effectiveness of the proposed method.

The remainder of this paper is structured as follows: Section 2 reviews state-of-the-art machine and deep learning models for neuroscience-informed and price fluctuations forecasting in financial markets. Section 3 presents data collection and a thorough description of the proposed model, emphasising its architecture and usefulness. Section 4 offers the hyperparameters setting, the results from a detailed comparative analysis of the proposed modular neural network against the state-of-the-art hybrid and single monolithic architectures retrieved from the literature, and discussions. Section 5 concludes this research's main findings, limitations and future directions.

## 2. Incorporating Rational Choice Theory with Neuroscience and AI Systems

This study comprehensively investigates diverse sub-fields, including neuroscience, informatics, economics, and machine and deep learning methods. The primary objective is thoroughly synthesising existing conceptual and empirical articles and surveys, encompassing primary research while conducting a meta-narrative review [35]. A semi-systematic review has also proved sufficient to understand better complex areas like NLP and business research [36–38]. A critical literature analysis was performed to foresee Forex hourly price fluctuations, selecting pertinent sources from Yahoo Finance and Twitter Streaming APIs for the EUR/GBP currency pair. Moreover, this study considered 15,796 recovered bibliographic records, focusing on renowned databases, including Scopus (n=11,620) and IEEE Xplore (n=4176). Targeted keyword searches were employed, confining topics such as "brain modularity", "financial decisions under risk", "biologically inspired machine", "rational choice theory for finance", "machine learning for Forex/stock predictions", "deep learning for Forex/stock price predictions", "social media analysis for Forex/stock predictions", "NLP for finance", "neuroeconomics", "artificial neural networks mimic brain", "Twitter sentiment analysis for Forex/stock markets", "CNN for Forex/stock predictions", "RNN for Forex/stock predictions".

By strategically scanning the titles, abstracts, and keywords of the retrieved articles and surveys to identify those that appeared most relevant to the aim of this study, applying the traditional technique of including peer-reviewed reports from reputable publishers, ensuring reliable and high-quality sources. Subsequently, by using exclusion criteria such as non-English language usage and duplicated articles, a subset of 150 articles was picked.

### 2.1. Brain Modularity and Computational Representations

As already discussed in Section 1, RCT could be a beneficial framework for understanding individual decision-making processes in the Forex markets. However, RCT has been criticised that the speculations assembled in this theory fail to consider the reality that the success of the outcome of a decision is affected by conditions that are not within the power of the individual making the decision [7]. One of the components that RCT neglects is the role of emotions in the choices of individuals, which could play an influential role in shaping the financial decision-making of investors [39]. Nevertheless, despite this criticism, the RCT has demonstrated a reasonable basis for defining how economic decisions are affected [7].



Moreover, neuroscience findings provide insights into rational choice's neural mechanisms by highlighting the brain's prefrontal cortex and insula positions [40]. For example, the ventromedial prefrontal cortex (vmPFC), a piece of the prefrontal cortex in the mammalian brain and anterior insula cortex (AIC), could represent distinct modules that influence rational and emotional decision-making, respectively, underscoring the significance of considering cognitive and affective factors that are indicated in the vmPFC and the AIC, including emotions and the ability to plan under risk process [41–44]; observing high modular variability in the insular regions. Researchers also suggested that the cortical brain regions vary fundamentally in their position, having a specific contribution to economic choices, which are mainly determined by the inputs of each area [45]. The modular approach to operating neuroanatomy of financial decision-making confirms the actions of economic choices, such as comparing values, in the regional architecture of the brain [46,47].

Neuroscientists have also examined computational brain modularity to explain brain functionalities. For example, Tzivilivaki et al. [48] investigated that complex, non-linear dendritic computations necessitate the development of a new theory of interneuron arithmetic. Using thorough biophysical models, they foresaw that the dendrites of FS basket cells in both the hippocampus and the prefrontal cortex are supralinear and sublinear. Furthermore, they compared a Linear ANN, in which the input from all dendrites is linearly merged at the cell body, and a two-layer modular ANN, in which the input is fed into two parallel, separated hidden layers. Despite that, the linear ANN exhibited relatively good performance; the two-layer modular ANN surpassed the respective linear ANN, which failed to illustrate the variance assembled by discrepancies in the input area. Based on these findings, the topology of the proposed modular network was selected.

Yao et al. [49] developed a deep learning model for image classification that combines two types of neural networks. More specifically, their model uses a parallel system that combines a CNN and an RNN for image feature extraction and a unique perceptron attention mechanism to unite the features from both networks. Their findings have shown that their suggested method outperforms current state-of-the-art methods based on CNN, demonstrating the benefits of using a parallel structure. Additionally, deep learning using CNNs and RNN can benefit NLP, indicating topic-level representations of sentences in the brain region by capturing intricate relationships of words and sentences. This ability could be crucial for investor sentiment analysis in the frame of the Forex market [50,51].

More recently, Flesch et al. [52] uncovered that the “rich” learning approach, which structures the hidden units to prioritise relevant features over irrelevant ones, results in neural coding patterns consistent with how the human brain processes information. Additionally, they found that these patterns evolve as the task progresses. For example, when they trained deep CNNs on the task using the “rich” learning method, they discovered that it induced structured representations that progressively transformed inputs from a grid-like structure to an orthogonal design and eventually to a parallel system. These non-linear, orthogonal and parallel representations demonstrated a vital element of their research, as they suggest that the neural networks can code for multiple, potentially contradicting tasks effectively.

In financial markets, Baek and Kim [53] proposed ModAugNet, a framework integrating a novel data augmentation technique for stock market index forecasting. The model comprises a prediction LSTM module and an overfitting prevention LSTM module. The performance evaluation using S&P500 and KOSPI200 datasets demonstrated ModAugNet-c's superiority over a monolithic deep neural network, an RNN, and SingleNet, a comparable model without the overfitting prevention LSTM module. The test of Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE) errors for S&P 500 decreased to 54.1%, 35.5%, and 32.7%, respectively, and for KOSPI200 decreased to 48%, 23.9%, and 32.7%, respectively. A limitation of their study was the exclusion of other information sources like news and investors' sentiment. Similarly, Lee and Kim [54] proposed NuNet, an end-to-end integrated neural network, to enhance prediction accuracy for S&P 500, KOSPI200, and FTSE100 prices. NuNet's feature extractor modules and trend sampling technique outperformed all baseline models across the three indexes, including SingleNet and the SMA [53].

Below is an overview of financial predictive models in markets, including state-of-the-art methods, which have shown promising performance. These models highlight the potential of incorporating innovative techniques to enhance prognosis accuracy and inform investment decisions.

## 2.2. Overview of Machine and Deep Learning Financial Predictive Models

In order to predict challenging financial markets' fluctuations and accurately forecast them, researchers have proposed several machines and deep learning methods such as the CNNs, the variants of RNNs, namely the GRU and LSTM, and their hybrid and single architectures. For example, Galeshchuk and Mukherjee [55] suggested a CNN for predicting the price change direction in the Forex market. They utilised the daily closing rates of EUR/USD, GBP/USD, and USD/JPY currency pairs. Moreover, they compared the results of CNN with baselines models such as the majority class (MC), autoregressive integrated moving average (ARIMA), exponential smoothing (ETS), ANN and SVM. Their findings showed that the baseline models and SVM yielded an accuracy of around 65%, while their suggested CNN model had an accuracy of about 75%. Deep learning architectures such as the LSTMs were recommended for future investigation in Forex.

Shiao et al. [56] employed the support vector Regression (SVR) and the RNN with LSTM to capture the dynamics of Forex data using the closing price of the USD/JPY exchange rate. The results indicated that their suggested RNN model outperformed the SVR model with a Root Mean Square Error (RMSE) of 0.0816, which achieved an RMSE of 0.1398, respectively. Maneejuk and Srichaikul [57] investigated which ARIMA, ANN, RNN, LSTM, and support vector machines (SVM) models presented better performance to the Forex market predictions. They used the daily closing price of five currencies: the Japanese yen, Great Britain Pound, Euro, Swiss franc, and the Canadian dollar for six years. Each model's performance was evaluated using the RMSE, MAE, MAPE and Theil U. Their findings showed that the ANN outperformed the other models in predicting the CHF/USD currency pair. On the other hand, the LSTM obtains better results than the other methods in predicting EUR/USD, GBP/USD, CAD/USD, and JPY/USD rates. For instance, the LSTM achieved the MAE of 0.0300 in the prediction of the EUR/USD compared to the MAE of 0.0435, 0.0319, 0.0853, 0.0560 obtained from the ARIMA, ANN, RNN, LSTM, and SVM models, respectively.

Hossain et al. [58] suggested a model based on deep learning to forecast the stock price of the Standard & Poor's 500 (S&P 500) from 1950 to 2016; combining LSTM and GRU networks; comparing to a multilayer perceptron (MLP), CNN, RNN, Average Ensemble, Hand-weighted Ensemble and Blended Ensemble. Their findings revealed that the LSTM-GRU model surpassed the other methods, achieving an MSE of 0.00098, with the other models accomplishing MSE of 0.26, 0.2491, 0.2498, 0.23, 0.23, and 0.226, respectively. Similarly, Althelaya et al. [59] investigated LSTM architectures to forecast the closing prices of the S&P500 for eight years. Their findings showed that the Bidirectional LSTM (BLSTM) was the most appropriate model; outperforming the MLP-ANN, the LSTM and the stacked LSTM (SLSTM) models; achieving the lowest error in the short- and long-term predictions. For example, the BLSTM achieved an MAE of 0.00736 in the short-term forecasts compared to MAE of 0.03202, 0.01398 and 0.00987 for the MLP-ANN, LSTM and SLSTM, respectively.

Lu et al. [60] proposed a predicting technique for stock prices employing a combination of CNN and LSTM, which utilises the memory function of LSTM to analyse relationships among time series data and the feature extraction capabilities of CNN. Their CNN-LSTM model uses opening, highest, lowest and closing prices, volume, turnover, ups and downs, and change as input and extracts features from the previous ten days of data. Their method is compared to other forecasting models such as LSTM, MLP, CNN, RNN, and CNN-RNN. The results showed that their CNN-LSTM outperformed the other models by presenting an MAE of 27.564 in contrast to MLP with 37.584, CNN with 30.138, RNN with 29.916, LSTM with 28.712 and CNN-RNN with 28.285. They concluded that their proposed CNN-LSTM could provide a reliable reference for investors' investment decisions. However, their model still needs to improve, as it only considers the effect of stock price data on closing prices rather than combining sentiment analysis and national policies into the predictions.

Alonso-Monsalve et al. [61] considered a convolutional LSTM (CLSTM) as an alternative to the traditional CNN, MLP, and the radial basis function neural networks (RBFNN) for predicting the

price movements of cryptocurrency exchange rates utilising high frequencies. Their study compared the performance of CLSTM against CNN, MLP, and RBFNN on six popular cryptocurrencies: Bitcoin, Dash, Ether, Litecoin, Monero, and Ripple. The results showed that the CLSTM network outperformed all other models significantly and was in place to predict the trends of Dash and Ripple by 4% over the trivial classifier. The CNNs also provided good results, particularly for Bitcoin, Ether, and Litecoin. Their study concludes that CNNs and CLSTM networks are suitable for predicting the trend of cryptocurrency exchange rates. However, a drawback of their study was limited to one year, which indicates that satisfactory outcomes may not be assured for other duration.

Kanwal et al. [62] proposed a hybrid deep learning technique forecasting the prices of Crude Oil, Crude Oil (CL=F1) and Global X DAX Germany ETF (DAX) for the individual stock item, DAX Performance-Index (GDAXI) and Hang Seng Index (HSI). Their Bidirectional Cuda Deep Neural Network Long Short-Term Memory that compounds BiLSTM Neural Networks and a one-dimensional CNN (BiCuDNNLSTM-1dCNN) compared against the LSTM deep neural network (LSTM-DNN), the LSTM-CNN, the Cuda Deep Neural Network Long Short-Term Memory (CuDNNLSTM), and the LSTM. The results from their study showed that the BiCuDNNLSTM-1dCNN outperformed the other models, validating the outcomes by using the RMSE and MAE metrics; for instance, in the DAX predictions, the BiCuDNNLSTM-1dCNN achieved MAE of 0.566; while the LSTM-DNN, the LSTM-CNN, the CuDNNLSTM, and the LSTM achieved MAE of 0.991, 3.694, 2.729, 4.349 in the test dataset, respectively. Features such as sentiment information have not been exploited in their study.

Pokhrel et al. [63] analysed and compared the performance of three deep learning models, LSTM, GRU, and CNN, in predicting the next day's closing price of the Nepal Stock Exchange (NEPSE) index. The study uses fundamental market data, macroeconomic data, technical indicators, and financial text data of the stock market of Nepal. Their models' performances are compared using standard assessment metrics like RMSE, MAPE, and Correlation Coefficient (R). Their results indicated that the LSTM model architecture provides a superior fit with the smallest RMSE 10.4660 MAPE 0.6488 and with R score 0.9874 in contrast to the GRU with RMSE 12.0706, MAPE 0.7350, R 0.9839, and the CNN with RMSE 13.6554, GRU 0.8424, R 0.9782. Their study also suggested that the LSTM model with 30 neurons was the supreme conqueror, followed by GRU with 50 neurons and CNN with 30 neurons. Finally, they proposed developing hybrid predictive models, implementing hybrid optimisation algorithms, and comprising other media sentiments in the model development methodology for future work.

The studies mentioned above have achieved significant results in predicting the financial markets. However, researchers have pointed out that there is still much potential for investigating the use of time series models such as LSTM and GRU in Forex predictions. These models are known for their ability to capture long-term dependencies in time-series data, which can be very useful in the context of Forex forecasts. In the context of the Forex market, they have also indicated that modular neural networks alongside the rising trend of NLP are another approach that has yet to be extensively explored in price fluctuations predictions [64,65]. However, the challenge associated with using modular neural networks is that it can be hard to design and train the individual modules in a way that leads to an effective combination in the final network decision; therefore, more research is needed to determine their effectiveness and practicality in this domain.

### 2.3. Critical Analysis

The multidisciplinary review within this study, incorporating recent neuroscience and financial market insights, underscores the ongoing need to enhance machine and deep learning methods. It also highlights the importance of modular design as a solution to the challenges posed by monolithic architectures [64]. Monolithic neural networks often suffer from catastrophic forgetting when learning new skills, altering their previously acquired knowledge. The study advocates for neural networks inspired by the modular organisation of human and animal brains, capable of integrating new knowledge without erasing existing knowledge—a fundamental consideration [66]. In addition,

the direction of examining investors' sentiment combined with economic indicators like closing prices is a promising trend requiring further investigation [65].

In the realm of computational models, recent studies highlight the significance of techniques like orthogonal initialisation and MCD, which improved the performance of ANNs [67,68]. These techniques diverge from models relying solely on default weights and conventional dropout methods frequently implied in exploring financial predictive models from the literature, conceivably by enhancing predictive performance. Simultaneously, primary data plays a pivotal role in this research, offering a direct path to its aim of forecasting the hourly closing price of EUR/GBP, which is integral to financial analysis [69]. These data, meticulously gathered from Yahoo Finance (closing prices) and Twitter (sentiments) APIs, seamlessly align with the study's context [70,71]. Beyond introducing and comparing baseline models to optimally partition the data, these sources enable a comprehensive assessment of state-of-the-art hybrid and single monolithic architectures selected from the literature, which were relevant to this study's aim and feasible for replication.

2.3.1. Baseline Models

The significance of baselines is crucial in this study as they were created to address research gaps, such as the limited utilisation of MCD and the orthogonal kernel initialisation reducing overfitting and potentially enriching the Forex market's anticipation. These new models provide a starting point for further analysis. They could help researchers identify areas for improvement as an essential tool in designing possible more accurate predictive models concerned further in Section 3. Moreover, baselines are vital for effectively partitioning the input domain in the context of Forex predictions. This partitioning, in turn, optimally allocates inputs, thereby enhancing task performance. This importance is substantiated by primary research leveraging closing prices and sentiment scores from Yahoo Finance and Twitter Streaming APIs as inputs aggregated based on hourly rates within 2018-2019. Tables 1 and 2 present the test performance of the baseline models in anticipating the EUR/GBP hourly closing price based on the MSE, MAE and Mean Squared Logarithmic Error (MSLE) objective evaluation metrics.

The CoRNNMCD and the CoGRUMCD performed better than the other baseline models in Tables 1 and 2, presenting less error in the MSE, MAE and MSLE test sets for the closing prices and sentiment scores, respectively. Moreover, these two baselines will be used to develop the proposed modular neural network model. For instance, CoRNNMCD in closing prices (Table 1) demonstrated fewer errors in the test sets, decreasing the MSE by 1.12%, 1.54%, 1.32%, and 60.68% for the CoRNN, CoGRUMCD, CoGRU and 1D-CNN, respectively. Likewise, sentiment scores (Table 2) presented better arrangement in CoGRUMCD with fewer errors in test sets by decreasing the MSE by 3%, 1.52%, 1.52% and 11.59% for the CoRNNMCD, CoRNN, CoGRU and 1D-CNN, respectively. The typical 1D-CNNs did not employ the orthogonal RNNs and MCD instead of pooling layers and were used as a baseline showing less execution time in closing prices and sentiments. However, their MSE was significantly higher than the other baselines. Also, using MCD increased could increase baselines computational time. However, its application led to significantly better performance in the selected baselines.

**Table 1.** Baseline models performance metrics in closing price (EUR/GBP).

Model	Variables	Metrics	Train	Valid	Test	Time Duration
CoRNNMCD	Closing Prices	MSE	6.7184e-05	6.2938e-05	<b>5.8785e-05</b>	2:30
		MAE	0.00549	0.00538	<b>0.00529</b>	
		MSLE	3.1801e-05	2.79863e-05	<b>2.6419e-05</b>	
CoRNN	Closing Prices	MSE	6.8642e-05	6.3699e-05	5.9447e-05	1:27
		MAE	0.00551	0.00541	0.00532	



CoGRUMCD	Closing Prices	MSLE	3.2413e-05	2.8352e-05	2.6773e-05	7:38
		MSE	6.9541e-05	6.4196e-05	5.9700e-05	
		MAE	0.00551	0.00542	0.00532	
CoGRU	Closing Prices	MSLE	3.3150e-05	2.8718e-05	2.7013e-05	3:59
		MSE	6.9597e-05	6.4106e-05	5.9565e-05	
		MAE	0.00552	0.00541	0.00532	
1D-CNN	Closing Prices	MSLE	3.3222e-05	2.8713e-05	2.6914e-05	0:36
		MSE	0.00012	0.00011	0.00011	
		MAE	0.00763	0.00757	0.00747	
		MSLE	5.3941e-05	4.800e-05	4.6293e-05	

**Table 2.** Baseline models performance metrics in sentiments (EUR/GBP).

Model	Variables	Metrics	Train	Valid	Test	Time Duration
CoRNNMCD	Sentiments	MSE	0.00079	0.00076	0.00067	2:21
		MAE	0.01535	0.01512	0.01456	
		MSLE	0.00031	0.00027	0.00026	
CoRNN	Sentiments	MSE	0.00077	0.00076	0.00066	1:24
		MAE	0.01504	0.01489	0.01428	
		MSLE	0.00029	0.00027	0.00025	
CoGRUMCD	Sentiments	MSE	0.00076	0.00074	<b>0.00065</b>	6:09
		MAE	0.01465	0.01439	<b>0.01394</b>	
		MSLE	0.00029	0.00026	<b>0.00024</b>	
CoGRU	Sentiments	MSE	0.00077	0.00075	0.00066	3:47
		MAE	0.01497	0.01479	0.01418	
		MSLE	0.00030	0.00027	0.00025	
1D-CNN	Sentiments	MSE	0.00085	0.00083	0.00073	<b>0:36</b>
		MAE	0.01661	0.01644	0.01591	
		MSLE	0.00034	0.00031	0.00028	

2.3.2. Hybrid Benchmark Models

The choice of hybrid algorithms for this study prioritised adopting the most current, reputable and state-of-the-art techniques available, which can replicate as well, according to the provided information by the authors. This focus on the most recent and state-of-the-art models ensures that the study is grounded in the latest developments and contributes to advancing understanding in the forecast of hourly EUR/GBP price fluctuations.

Table 3 shows that the CNN-LSTM performed better than the other models, presenting more inconsequential errors in the MSE, MAE and MSLE test sets. For example, CNN-LSTM demonstrated fewer errors in the test sets, decreasing the MSE by 72.66%, 66.61%, 194.55%, and 60.68% for the BiCuDNNLSTM, LSTM-GRU and CLSTM, respectively. The BiCuDNNLSTM presented less

execution time, but its MSE was higher than the CNN-LSTM. It is worth mentioning that the BiCuDNNLSTM is running in GPU based on CUDA utilisation, which can boost the speed of training time of deep learning models. Moreover, factors such as the timesteps of each hybrid model can affect its execution time, as discussed in Section 4.

**Table 3.** Hybrid models performance metrics in closing prices and sentiment scores (EUR/GBP).

Model	Metrics	Train	Valid	Test	Time Duration
BiCuDNNLSTM [62]	MSE	0.00013	0.00015	0.00015	<b>1:19</b>
	MAE	0.00848	0.00874	0.00866	
	MSLE	6.4001e-05	6.6003e-05	6.9725e-05	
CNN-LSTM [60]	MSE	6.4471e-05	5.9427e-05	<b>7.005e-05</b>	1:41
	MAE	0.00538	0.00536	<b>0.00552</b>	
	MSLE	3.0273e-05	2.6574e-05	<b>3.2089e-05</b>	
LSTM-GRU [58]	MSE	0.00017	0.00014	0.00014	3:03
	MAE	0.00935	0.00901	0.00887	
	MSLE	7.8049e-05	6.3934e-05	6.4782e-05	
CLSTM [61]	MSE	0.00501	0.00514	0.00507	2:18
	MAE	0.05722	0.05764	0.05743	
	MSLE	0.00229	0.00231	0.00234	

2.3.3. Single Benchmark Models

Likewise, the choice of algorithms for this study strongly emphasised selecting the most recent single methods employed for the possible hourly price fluctuation forecast in the EUR/GBP.

Table 4 revealed that the GRU performed better than the other models, presenting less error in the MSE, MAE and MSLE test sets. For instance, GRU exhibited fewer errors in the test sets, decreasing the MSE by 26.68% and 195.01% for the 2D-CNN and LSTM, respectively. LSTM presented less execution time, implying 30 neurons and an Adam optimiser that can obtain a faster convergence rate. However, the MSE of LSTM was considerably higher than the GRU. Likewise, with the hybrid models, the time step of a single model can affect its execution time, as debated in more detail in Section 4.

**Table 4.** Single models performance metrics in closing prices and sentiment scores (EUR/GBP).

Model	Metrics	Train	Valid	Test	Time Duration
2D-CNN [63]	MSE	0.00012	0.00011	0.00012	0:51
	MAE	0.00760	0.00746	0.00765	
	MSLE	5.3407e-05	4.8839e-05	5.9233e-05	
GRU [63]	MSE	8.7491e-05	7.8116e-05	<b>9.1750e-05</b>	0:47
	MAE	0.00628	0.00595	<b>0.00614</b>	
	MSLE	4.0481e-05	3.5261e-05	<b>4.5704e-05</b>	
LSTM [63]	MSE	0.00732	0.00723	0.00736	<b>0:10</b>
	MAE	0.04911	0.04911	0.04922	
	MSLE	0.00334	0.00327	0.00338	

3. Materials and Methods

This section provides a comprehensive overview of the data collection process, including selecting relevant Forex market data sources and the steps implemented to ensure the consistency of each dataset. Eventually, it describes the proposed modular neural network model for predicting the hourly price fluctuation in EUR/GBP pair.

3.1. Data Collection

The EUR/GBP, exchange rate data consist of the closing price values and sentiment information retrieved from Yahoo Finance API (<https://developer.yahoo.com/api/>) and Twitter Streaming API (<https://developer.twitter.com>) on an hourly rate, respectively.

The predicted hourly intraday trading of the closing price EUR/GBP rate is the defined target from January 2018 to December 2019 for 12,436 hours. However, because the Forex prices incorporate missing values, Twitter's sentiment data utilises the same hourly timeframe to be aligned with the pricing data. This can be achieved by employing a data processing framework in Python, such as Pandas, to construct data entries amalgamating aggregated sentiment scores and stock prices per designated hour/date intervals. Furthermore, a feature-level fusion based on the same hourly timeframe is considered to achieve the data coalition from both APIs. Finally, after the data fusion, the EUR/GBP exchange rate closing prices and the sentiment scores are provided for each module of the MCoRNNMCD-ANN model.

### 3.1.1. Forex Closing Prices

The Forex closing prices of the EUR/GBP rate generated from the Yahoo Finance API from January 2018 to December 2019 for 12,436 hours. The data includes the open, high, low and close values. Only the closing price is taken into consideration as considered the most helpful indicator to foresee Forex markets [72]. One hour is deemed most suitable for better anticipating financial markets because it is shorter than daily or yearly forecasting [73]. Finally, it is worth mentioning that research on data requirements for predicting time series using ANNs revealed that utilising data of one to two years yields the highest accuracy [74].

### 3.1.2. Sentiment Data

The Twitter Streaming API is utilised by tuning the appropriate parameters for the needs of this study. The language parameter indicates whether a user wants to receive tweets only in one or some specific languages in terms of the tweet's text. More specifically, the "language = en" parameter is specified because extracting tweets from an English text was considered more appropriate, as all existing dictionaries support the English language. Using Tweepy enables handling the profile of a user and the data collection by assessing specific keywords; this study uses hashtags such as search words = "#eurgbp", "#forexmarket", and "#forex", referring to EUR/GBP currency pairs. The Twitter data is summarised at an hourly rate. Finally, 3,265,896 have been retrieved from January 2018 to December 2019 for 12,436 hours.

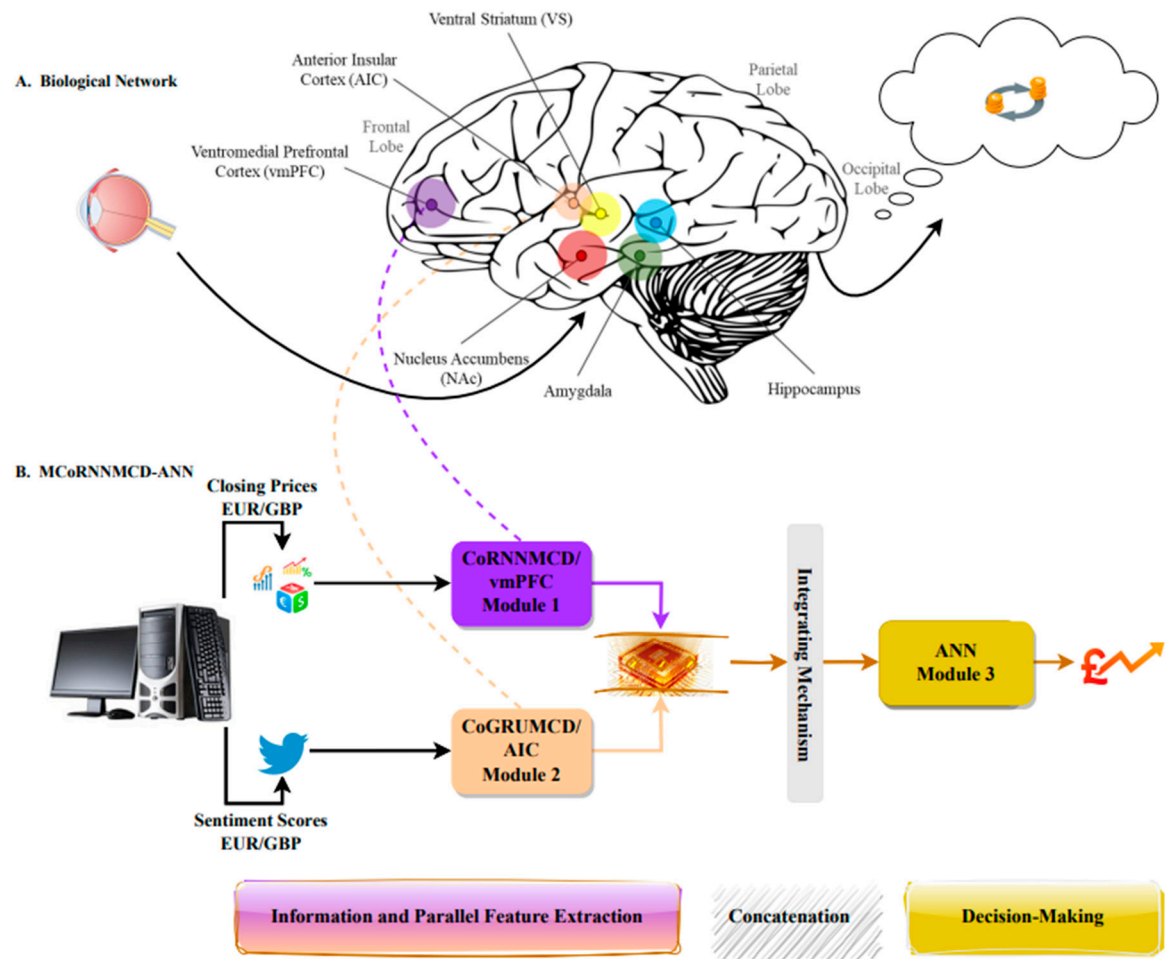
A rule-based sentiment analysis lexicon, namely Valence Aware Dictionary for Sentiment Reasoning (VADER), is utilised to extract each sentiment score from the Twitter data [75]. VADER has yielded enormous results, considering the labelling of a tweet that outperforms even from a human factor rating. VADER delivers a compound ratio, giving the sentiment score comprising negative, positive and neutral classes. For example, from the 3,265,896 tweets of the EUR/GBP exchange rate, VADER yielded the following results: 747,890 (22.9%) negative, 930,780 (28.5%) positive and 1,587,226 neutral (48.6%) tweets, from January 2018 to December 2019 for 12,436 hours of EUR/GBP rate.

## 3.2. Proposed Novel Bio-Inspired Model in Predicted Forex Market Price Fluctuations

This study proposes a novel bio-inspired Modular Convolutional orthogonal Recurrent MCD-ANN (MCoRNNMCD-ANN), aiming to encounter the limitations of the current monolithic architectures presented in the literature. The proposed modular network incorporates a new CNN architecture to address catastrophic forgetting, overfitting, vanishing and exploding gradient problems, and underspecification [76]. Therefore, a proposed new CNN architecture incorporates a modular topology inspired by Tzivilaki et al. [48], formulating a convolutional, orthogonal recurrent MCD replacing the pooling layers, followed by dense layers flattening their outputs. Compared with a typical CNN time series composed of convolutional, pooling, flattened and dense layers, the

proposed new CNN could enhance the robustness and forecasting performance of the Forex market [77].

Consequently, in the proposed MCoRNNMCD-ANN, the modules selected from baselines (Tables 1 and 2) displayed better results in partitioning the input domain in anticipating EUR/GBP price movements. Hence, two separate and parallel features extraction convolutional with orthogonal kernel initialisation applied in simple RNN and a GRU coupled with MCD networks, receiving the closing prices and sentiment scores capture long-term dependencies in the EUR/GBP exchange rate hourly, replacing the pooling layers were considered. The replacement occurs to avoid the down sampling of feature sequences by losing valuable information since the pooling layers capture only the essential features in the data and ignore the less important ones, which can be vital [78]. The dense layers are also placed before the flattening operation in both modules in the proposed novel CNN architecture. This adaptation transpires because the dense layer's preliminary purpose is to increase the model's capacity to learn more complex patterns from the RNN's output. The flatten operation is then applied to reshape the result of the dense layer for each module into a one-dimensional tensor to prepare it for the combined outputs by integrating them into a final concatenation layer. Ultimately, the concatenated features passed in the final decision module consist of a three-layer feed-forward ANN that yields the anticipated hourly closing price of EUR/GBP. Figure 1 displays the proposed MCoRNNMCD-ANN model.



**Figure 1.** The proposed MCoRNNMCD-ANN model simulates the vmPFC and AIC of the brain.

Module 1: Convolutional Orthogonal RNN-MCD (CoRNNMCD)

Let us consider time series data representing the hourly closing prices of the EUR/GBP currency pair. The input data can be characterised as a matrix  $x \in \mathbb{R}^{d_c \times l}$ , where  $d_c$  is the number of channels



(in this case, 1 for a single currency pair) and  $l$  is the length of the time series. In the initial CNN, the convolution operation can be mathematically represented as:

$$y[i] = f((x * w)[i] + b) \quad (1)$$

Here,  $x$  represents the input data,  $w$  denotes the filters or kernels,  $b$  is the bias term, and  $f$  is the activation function. The dot product operation  $(x * w)[i]$  is performed between the filter  $w$  and the portion of the input data  $x$  the filter is currently "looking at". The activation function  $f$  is then applied element-wise to the result of the dot product, adding non-linearity to the output.

Moving on to Module 1, a 1D convolutional layer is applied to the input data. The convolution operation is performed using a set of filters or kernels, denoted as  $w \in \mathbb{R}^{d_c \times r}$ , where  $r$  is the size of the filter. Mathematically, the convolution between a 1D filter  $w$  of size  $r$  and an input signal of length  $l$  can be defined as:

$$(x * w)[i] = \sum_{j=1}^r x[i + j]w[j] \quad (2)$$

Here,  $*$  denotes the convolution operation, and  $i$  ranges from 1 to  $l - r + 1$  to ensure the filter fits entirely within the input signal. The variable  $j$  ranges from 1 to  $r$  and represents the position within the filter and the corresponding elements in length  $l$  input signal. After the convolution, the activation function is applied element-wise to each element of the convolution result, adding non-linearity. Next, the convolution operation generates a new feature representation, denoted as  $W \in \mathbb{R}^{(l-r+1) \times m_c}$ , where  $m_c$  is the number of filters. The output feature map  $C$  of the 1D convolutional layer is defined as the input to the RNN, which directly replaces the max pooling layer. The feature map  $C$  is represented as a matrix  $W$ , where each row corresponds to a window vector  $w_n = [x_n, x_{n+1}, \dots, x_{n+r-1}]$ . To feed the window  $W$  into an RNN, the hidden state is computed as  $h_t \in \mathbb{R}^{m_h}$  where  $m_h$  represents the dimension of the hidden state in the recurrent network at each time step  $t$ . The hidden state  $h_t$  in the equation of the simple RNN is calculated as:

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h). \quad (3)$$

Here,  $x_t \in \mathbb{R}^{m_c}$  represents the input at time step  $t$ ,  $W_{xh} \in \mathbb{R}^{m_c \times m_h}$  and  $W_{hh} \in \mathbb{R}^{m_h \times m_h}$  are weight matrices,  $h_{t-1}$  the previous hidden state, and  $b_h \in \mathbb{R}^{m_h}$  is a bias term. The non-linear activation function  $\phi$ , such as the Rectified Linear Units (ReLU), is applied element-wise to each hidden state  $h_t$  [79]. Replacing the max pooling layer with an RNN allows capturing sequential dependencies in the time series data. In addition, the RNN considers the temporal information and improves the model's performance in predicting future values. After the RNN layer, a dense layer can be added to generate the network's output. The dense layer takes the hidden state  $h_t$  as input and applies the following equation:

$$y = f(W_y h_t + b). \quad (4)$$

Here,  $W_y$  is the weight matrix,  $b$  is the bias term, and  $f$  is the activation function, such as softmax, which converts the output into a likelihood distribution over the possible classes. Finally, a flattened layer takes the output of the dense layer as input, computed as:

$$F[i] = \text{flatten}(y[i]). \quad (5)$$

It is worth mentioning that the dense layer after the RNN can allow the model to learn complex relationships and mappings between the input and the desired output while flattening the outputs of the dense can simplify the data structure by collapsing the dimensions, making it compatible with following layers that expect one-dimensional inputs. The backpropagation technique (BPTT) is utilised to train an RNN. However, RNNs require help to learn long-term dependencies during the BPTT training process since the gradients employed to update the weights increase exponentially. A procedure is known as the vanishing or exploding gradient problem.

In this study, orthogonal initialisation is considered one of the proper mechanisms to address the vanishing gradient issue in the RNNs. Therefore, the kernel weights  $W$  of the RNN will be transformed into orthogonal ( $O$ ) [80]. Furthermore, the parametric rectified linear unit (PReLU)

activation function is utilised instead of the tahn activation function since it is considered one of the keys to deep networks' recent success in time series analysis [81]. Finally, to potentially enhance the performance of the orthogonal RNN receiving the outputs of the 1D-CNN closing price for EUR/GBP as inputs, the MCD is coupled to the RNN layer (CoRNNMCD) in its capability to quantify model uncertainty facilitating more informed decision-making in Forex forecast [82]. The hidden state  $h_t$  in equation (3) presented above is updated and computed as,

$$h_t = \text{PReLU}((Ox_t + W_{hh}h_{t-1} + b_h) \odot \text{MCD}). \quad (6)$$

The output of the CoRNNMCD is fed to the dense layer and computed as,

$$y_t = \text{linear}(W_{hy}h_t + b_y) \quad (7)$$

where  $h_t$  is the hidden state at a time  $t$ ,  $x_t$  is the input at a period  $t$ ,  $W_{hh}$ ,  $W_{hy}$  are weight matrices,  $b_h$ , and  $b_y$  are the bias vectors,  $O$  is an orthogonal matrix used to initialise the input weights,  $\odot$  represents an element-wise multiplication, and MCD is the Monte Carlo Dropout.

Finally, a flattened layer receives as an input the output of the dense layer indicated as,

$$F[t] = \text{flatten}(y[t]). \quad (8)$$

## Module 2: Convolutional Orthogonal GRU-MCD (CoGRUMCD)

Module 2 uses a 1D convolutional layer for sentiment analysis on a time-series task. Nevertheless, first, let us summarise the key components and equations: Given the input data  $x \in \mathbb{R}^{d_s \times l}$ , where  $d_s$  is the number of channels (1 in this matter), and  $l$  is the length of the time series.  $x$  represents the input data at each time step  $t$ . The window  $w_n$  is formed by selecting  $r$  consecutive sentiment scores starting from time step  $n$ , expressed as  $w_n = [x_n, x_{n+1}, \dots, x_{n+r-1}]$ . The 1D convolutional layer processes the window  $w_n$  to extract convolutional features. The output of the convolutional layer, denoted as  $S \in \mathbb{R}^{l-r+1 \times m_s}$ , consists of  $m_s$  feature maps. The parameter  $m_s$  determines the number of feature maps representing the filters used in the convolutional layer. The convolutional features in  $S$  are new window representations, capturing different patterns or representations in the input time series. The output feature maps in  $S$  are then fed into a GRU computed as:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (9)$$

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (10)$$

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h) \quad (11)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (12)$$

where  $h_t$  is the hidden state at time  $t$ ,  $x_t$  is the intake at time  $t$ ,  $r_t$  and  $z_t$  are the reset and update gates, respectively.  $\tilde{h}_t$  is the candidate hidden state,  $W_r$ ,  $W_z$ ,  $W_h$  are weight matrices,  $b_r$ ,  $b_z$ , and  $b_h$  are the bias vectors, and  $\odot$  represents an element-wise multiplication.

The convolutional orthogonal GRU-MCD (CoGRUMCD) updates the initial GRU equations to incorporate the convolutional features and learn temporal dependencies in the sentiment scores as follows:

$$r_t = \sigma(O_r x_t + W_r h_{t-1} + b_r) \quad (13)$$

$$z_t = \sigma(O_z x_t + W_z h_{t-1} + b_z) \quad (14)$$

$$\tilde{h}_t = \text{PReLU}(O_h x_t + W_h(r_t \odot h_{t-1}) + b_h) \odot \text{MCD} \quad (15)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (16)$$

where  $r_t$  is the reset gate at time step  $t$ ,  $z_t$  is the update gate at time step  $t$ ,  $\tilde{h}_t$  is the candidate value for the new hidden state at time step  $t$ ,  $h_t$  is the new hidden state at time step  $t$ ,  $O_r, O_z, O_h$  are orthogonal matrices used to initialise the input weights,  $W_r, W_z, W_h$  are weight matrices,  $b_r, b_z$ , and  $b_h$  are the bias terms and  $\odot$  represents an element-wise multiplication. The output of the CoGRUMCD is fed to the dense layer denoted as,

$$y_t = \text{linear}(W_{hy}h_t + b_y). \quad (17)$$

Finally, a flattened layer receives as an input the output of the dense layer computed as,

$$F[t] = \text{flatten}(y[t]). \quad (18)$$

### Parallel Feature Extraction and Concatenation

The parallel features extraction operation converges the two modules' tasks. The first module consists of a convolutional with orthogonal initialisation RNN that inputs the closing price. The second model consists of a convolutional orthogonal GRU that takes the sentiment scores for the EUR/GBP rate. The two modules are continuous, with hourly time frames. Let us assume that the feature vectors of the first and second inputs are denoted as  $x_1$  and  $x_2$  and contain values of module one and module two, respectively. The vectors then of the first and second outputs are represented as  $y_1$  and  $y_2$  for module one and module two, respectively, and define the output  $y_{1,2}$  of  $y_1$  and  $y_2$ . Let  $M_1$  be the first module (convolutional with orthogonal initialisation RNN) with input feature vector  $x_1$  (closing prices) and output vector  $y_1$ . Let  $M_2$  be the second module (convolutional orthogonal GRU) with input feature vector  $x_2$  (sentiment scores for EUR/GBP rate) and output vector  $y_2$ .

The parallel processing operation can be represented as:

$$y_{1,2} = [M_1(x_1), M_2(x_2)]. \quad (19)$$

The outputs from the parallel processing operating system, module 1 and module 2 that receive the closing price and the sentiment score are merged in the concatenation layer used as an integrating mechanism. The conjunct outputs are connected to the final module of the proposed MCoRNNMCD-ANN model, aiming to yield the anticipated closing price for EUR/GBP rate.

The information that merged in the concatenation layer is calculated as,

$$C = M_1 \cup M_2. \quad (20)$$

### Module 3: Decision-Making

The final part of the proposed MCoRNNMCD-ANN model takes place to make the decision consisting of a three-layer feedforward ANN. The first layer of the ANN receives the merged information and can be denoted as:

$$\text{Dense}_{d1} = \text{ReLU}(W_{\text{Dense}_{d1}} C + b_{\text{Dense}_{d1}}). \quad (21)$$

In the second dense layer, a proposed modified version of the Swish activation function, namely HSwish<sub>m</sub>, is applied. The main difference between the Swish activation function and the HSwish<sub>m</sub> function is that it utilised the hard sigmoid instead of the sigmoid in Swish [83,84]. Moreover, HSwish<sub>m</sub> utilised a different scaling factor, as calculated below:

$$\text{Swish}(x, \beta = 1) = x * \text{sigmoid}(\beta x). \quad (22)$$

The HSwish<sub>m</sub> function utilised a  $\beta=0.5$  computed as:

$$\text{HSwish}_m(x, \beta = 0.5) = x * \text{hardsigmoid}(0.5x). \quad (23)$$

The rationale behind  $\text{HSwish}_m$  is to mitigate the issue of exaggerated responses to minor fluctuations in the input. In financial markets like Forex, where prices can exhibit high volatility and noisy fluctuations, prediction models need robustness and stability. Using  $\text{HSwish}_m$  with  $\beta = 0.5$ , the model can potentially introduce a dampening effect on the negative inputs, resulting in smoother and more controlled responses. This dampening effect could also be beneficial in scenarios where the model has to avoid harsh reactions to minor input fluctuations. Hence, the second dense layer receives the output from the first dense layer, estimated as:

$$\text{Dense}_{d2} = \text{HSwish}_m(W_{\text{Dense}_{d2}} \text{Dense}_{d1} + b_{\text{Dense}_{d2}}). \quad (24)$$

The final output layer is denoted as,

$$\text{Dense}_{d3} = \text{linear}(W_{\text{Dense}_{d3}} \text{Dense}_{d2} + b_{\text{Dense}_{d3}}). \quad (25)$$

#### 4. Results and Discussion

This section provides the implementation of hyperparameters and a comprehensive overview of the comparative analysis conducted, where the performance of the proposed modular neural network model is compared against the state-of-the-art and single monolithic architectures in subsection 2.3. The objective evaluation metrics, namely MSE, MAE and MSLE, are used to assess prediction performance. The section also discusses the results obtained from the experiments, showcasing the skill and capabilities of the proposed MCoRNNMCD-ANN model in predicting hourly Forex price fluctuations of EUR/GBP.

##### 4.1. Design and Implementation

In this study, to conduct the experiments, the proposed MCoRNNMCD-ANN model setting is as follows: First, each dataset has been acquired from the Yahoo Finance API and Twitter Streaming API, incorporating the hourly closing price and sentiment data, applying normalisation method, respectively. Second, the datasets are divided into training, validation and testing sets, with the same portion of 60:20:20 used to improve the generalisability of the network. Third, the hyperparameters of the MCoRNNMCD-ANN model are encountered by employing the grid search method. Finally, the parameters below are considered to choose the most optimal for the proposed model receiving closing price and sentiment score evaluated by the MSE. The list of parameters is given below:

- Number of time steps: 20, 30, 40, 50, 60
- Number filters per layer: 32, 64, 128, 256, 512
- Number of nodes per hidden layer: 25, 30, 50, 60, 100
- MCD rates: 10% to 50%
- Batch sizes: 10, 20, 30, 60, 100

It is worth noting that the MSE used as an objective metric of evaluation in grid search is evaluating the performance of its model and not its final predictions that have different calculations in the shake of hourly Forex forecasting closing price. Accordingly, grid search produces the optimal hyperparameters described:

- The look-back window uses a time step of 60. Furthermore, 128 filters are selected as the optimum numbers of the 1D convolutional layer in modules one and two, incorporating the ReLU activation function. Additionally, in the orthogonal kernel initialised RNN and GRU layers coupled with MCD with 0.1 rates, supplanting the max-pooling layer in the initial CNN architecture, 50 neurons have been selected, utilising PReLU as the optimal activation function.
- The dense layers in modules one and two consist of 50 neurons integrating the ReLU activation function connected to the flattened layers. The decision-making module consists of 3 layers receiving the merge features from modules one and two. The first dense layer includes 50 neurons incorporating the ReLU activation function. The second dense layer contains the  $\text{HSwish}_m$ . The output of the decision-making part, receiving one neuron selecting the linear activation function, as it is appropriate for regression tasks, yielding the predicted hourly closing price fluctuations of the EUR/GBP exchange rate.



- A batch size of 20 has been selected. Early stopping is employed to avoid overfitting. The Adam optimiser with a learning rate of  $1e-04$  has been chosen as it proved effective for non-stationary objectives and problems with very noisy gradients, and the MSE as the loss function has been utilised during the proposed MCoRNNMCD-ANN for its training process. Each experiment of the proposed MCoRNNMCD-ANN against benchmarks has been repeated fifty times to be reliable.
- A computer with the following characteristics has been used to execute the experiments: Intel® Core™ i7-9750H (Hyper-Threading Technology), 16 GB RAM, 512 GB PCIe SSD, NVIDIA GeForce RTX 2070 8 GB. The Anaconda computational environment with the Python (version 3.6) programming language has been utilised to conduct the experiments.

After implementing the optimal parameters in the proposed MCoRNNMCD-ANN model, its performance based on the MSE, MAE and MSLE is provided in Table 5. Furthermore, the MCoRNNMCD-ANN outperformed all the baselines.

**Table 5.** MCoRNNMCD-ANN performance metrics in closing prices and sentiment scores (EUR/GBP).

Model	Metrics	Train	Valid	Test	Time Duration
MCoRNNMCDANN	MSE	6.5486e-05	6.1332e-05	5.7488e-05	2:35
	MAE	0.00534	0.00526	0.00518	
	MSLE	3.1117e-05	2.7342e-05	2.6051e-05	

#### 4.2. Benchmark Models

##### Hybrid Benchmark Models

The objective evaluation metrics of the proposed MCoRNNMCD-ANN shown in Table 5 revealed a decline in errors of the hybrid models presented in Table 3. For instance, MCoRNNMCD-ANN decreased to 89.17%, 19.70%, 83.56% and 195.51% for the test MSE of the BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, and CLSTM. The test MAE of MCoRNNMCD-ANN decreased to 50.30%, 6.36%, 52.53%, and 166.91% for the test MAE of the BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, and CLSTM. The test MSLE of MCoRNNMCD-ANN decreased to 91.20%, 20.77%, 85.28%, and 195.59% for the test MSLE of the BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, and CLSTM. The difference in time elapsed in minutes between the proposed MCoRNNMCD-ANN and the hybrid benchmark models presented in Table 3 has also been considered regarding their execution time. As a result, the execution time of MCoRNNMCD-ANN decreased to 28 minutes for the execution time of the LSTM-GRU. The execution time of MCoRNNMCD-ANN was increased to 76, 54 and 17 minutes for the execution time of BiCuDNNLSTM, CNN-LSTM, and CLSTM, respectively. Based on the outcomes, in most cases, the execution time of a model can be tremendously affected by the size of the window length and the complexity of the layers used in each model. It is worth mentioning that the BiCuDNNLSTM with the default parameters need less execution time as it runs in a GPU using CUDA, which accelerates deep learning models. Finally, the LSTM-GRU takes more execution time than the proposed MCoRNNMCD-ANN, even though it utilises a default size window of 30. This effect can result from the more utilised neurons and complex architecture since it employs only LSTM and GRU models. MCoRNNMCD-ANN outperformed benchmarks.

To conduct a fairer comparison, modified versions of benchmarks implemented the parameters from the proposed MCoRNNMCD-ANN model to investigate their performance as below:

- The modified parameters of BiCuDNNLSTM [62] utilise a window length of 60 instead of the default 50 timesteps, a convolution layer with a filter size of 128 instead of its default 64, a dropout layer with a rate of 0.1 instead of 0.2, the **HSwish<sub>m</sub>** activation function in the dense layer after the flattening layer instead of the default ReLu, linear as the output activation function instead of ReLu, MSE as the loss function instead of MAE, a batch size of 20 instead of 64, and early stopping is applied instead of 32 epochs.

- The modified parameters of the CNN-LSTM [60] neural network model are a window length of 60 instead of the default 50 timesteps, a convolution layer with a filter size of 128 instead of its default 32 with a ReLu activation function instead of tanh, an LSTM layer with 50 hidden units instead of 64, and the activation function used in this layer is parametric ReLu instead of that, MSE as the loss function instead of MAE, a batch size of 20 instead of 64, and early stopping is applied instead of 100 epochs.
- The modified parameters of the LSTM-GRU [58] neural network model is a window length of 60 instead of the default 30 timesteps, LSTM and GRU layers with 50 hidden units instead of 100 with the activation function PReLU for both layers instead of a hyperbolic tangent, without the inner activations to be set as hard sigmoid functions, Adam optimiser trains the network with the learning of 0.0001 instead of the rate of 0.001, and early stopping is applied instead of 20 epochs.
- The CLSTM [61] model was adjusted with 128 filters in the 1D convolutional layer, 60 timesteps instead of 15, and 50 neurons instead of 200, 100 and 150 neurons in the dense and LSTM layers. Moreover, LSTM employed MCD with PReLU instead of traditional dropout and ReLu activation function.

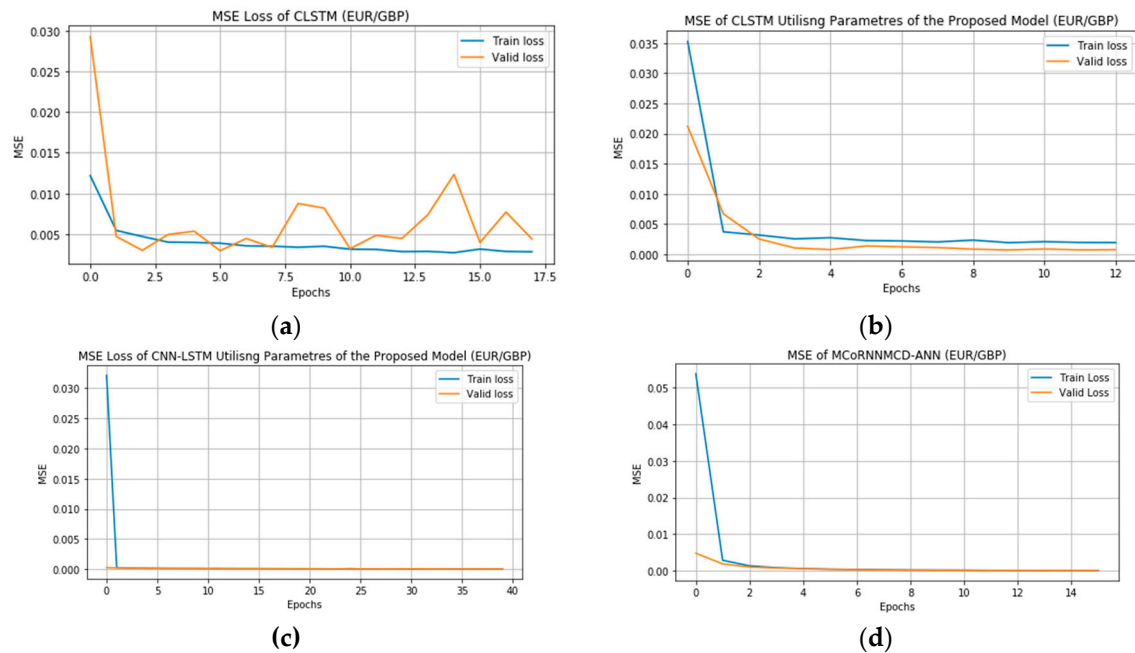
Table 6 confirmed that the MCoRNNMCD-ANN outperforms the state-of-the-art hybrid benchmarks adjusting with the parameters of the MCoRNNMCD-ANN.

**Table 6.** MCoRNNMCD-ANN performance metrics against adjusted (adj.) hybrid benchmarks.

Model	Metrics	Train	Valid	Test	Time Duration
BiCuDNNLSTM adj.	MSE	0.00011	0.00010	0.00010	2:54
	MAE	0.00761	0.00754	0.00746	
	MSLE	5.1970e-05	4.7161e-05	4.6294e-05	
CNN-LSTM adj.	MSE	7.2831e-05	6.7592e-05	6.2947e-05	7:13
	MAE	0.00572	0.00563	0.00551	
	MSLE	3.4659e-05	3.0217e-05	2.8432e-05	
LSTM-GRU adj.	MSE	0.00011	0.00010	0.00010	20:43
	MAE	0.00739	0.00721	0.00721	
	MSLE	5.4301e-05	4.6677e-05	4.6560e-05	
CLSTM adj.	MSE	0.00145	0.00151	0.00134	8:21
	MAE	0.02352	0.02381	0.02353	
	MSLE	0.00061	0.00063	0.00059	
MCoRNNMCDANN	MSE	6.5486e-05	6.1332e-05	<b>5.7488e-05</b>	<b>2:35</b>
	MAE	0.00534	0.00526	<b>0.00518</b>	
	MSLE	3.1117e-05	2.7342e-05	<b>2.6051e-05</b>	

The objective evaluation metrics revealed that the test MSE of MCoRNNMCD-ANN decreased to 53.98%, 9.10%, 53.98%, and 183.54% for the test MSE of the BiCuDNNLSTM, CNN- LSTM, LSTM-GRU, and CLSTM by adjusting their parameters with the parameters of the proposed MCoRNNMCD-ANN. Likewise, the test MAE of MCoRNNMCD-ANN decreased to 36.08%, 6.18%, 32.77%, and 127.83% for the test MAE of the adjusted BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, and CLSTM. The test MSLE of MCoRNNMCD-ANN decreased to 55.96%, 8.74%, 56.49%, and 183.01% for the test MSLE of the BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, and CLSTM, containing the parameters of the MCoRNNMCD-ANN. The difference in time elapsed in minutes between the MCoRNNMCD-ANN and the hybrid benchmark models adjusted with the parameters of the proposed model has shown that the execution time of MCoRNNMCD-ANN decreased to 19, 278, 1088 and 346 minutes for the execution time for the modified BiCuDNNLSTM, CNN- LSTM, LSTM-GRU, and CLSTM. Consequently, the execution time of hybrid benchmarks increased when the window length increased at 60 timesteps incorporating the MCD when usable. That validated the previous assumption that the timesteps play a remarkable role in the execution time of the models. Notably, the predictive error of the benchmarks adjusted with the proposed model parameters was reduced significantly and yielded better outcomes. Finally, the proposed MCoRNNMCD-ANN

significantly outperformed the adjusted benchmarks and was faster. Figure 2 illustrates an example of the MSE's tremendous improvement by utilising the parameters of the proposed MCoRNNMCD-ANN in the CLSTM and the best-performed hybrid benchmark MSE, namely CNN-LSTM adj., and the MSE of the MCoRNNMCD-ANN.



**Figure 2.** MSEs of CLSTMs, CNN-LSTM and the MCoRNNMCD-ANN model: (a) CLSTM default (b) CLSTM adjusted (c) CNN-LSTM adjusted (d) Proposed MCoRNNMCD-ANN.

### Single Benchmark Models

In Table 4, the results of the single benchmark models have been shown. The objective evaluation metrics demonstrated that the test MSE of MCoRNNMCD-ANN decreased to 70.44%, 45.91%, and 196.90% for the test MSE of the 2D-CNN, GRU, and LSTM, respectively. The test MAE of MCoRNNMCD-ANN decreased to 38.50%, 16.96%, and 161.91% for the 2D-CNN, GRU, and LSTM test MAE, respectively. The test MSLE of MCoRNNMCD-ANN decreased to 77.81%, 54.77%, and 196.94% for the test MSLE of the 2D-CNN, GRU, and LSTM, respectively. The difference in time elapsed between the proposed MCoRNNMCD-ANN and the benchmark-single models in minutes has also been considered regarding their execution time. As a result, the execution time of MCoRNNMCD-ANN was increased to 104, 108, and 145 minutes for the execution time of 2D-CNN, GRU, and LSTM with default parameters. The execution time of the models can again be tremendously affected by the size of the window length and the complicatedness of each model. For instance, when the window length of the 2D-CNN, GRU, and LSTM single model incorporates a smaller time window length equal to 5 timesteps decreasing the execution time training. On the other hand, even though the LSTM is more complex than the 2D-CNN and GRU, it took significantly less time to be trained since it utilised fewer neurons (30) than GRU (50 neurons) and an Adam optimiser that can obtain a faster convergence rate leading to being faster against Adagrad for CNN [85]. However, LSTM has the highest predictive error. Finally, MCoRNNMCD-ANN presented a minor prediction error by significantly outperforming the single models.

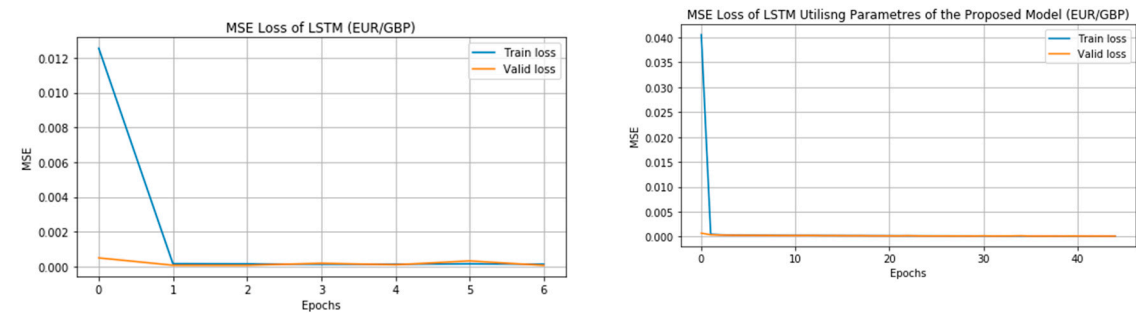
Similarly to the hybrid benchmark models, the single benchmarks [63] will be adjusted with the proposed MCoRNNMCD-ANN parameters for a fairer comparison. The modified parameters of the CNN, LSTM and GRU models are a window length of 60 instead of the default 5 timesteps, a convolution layer with a filter size of 128 instead of its default 30 for CNN, an LSTM with 50 hidden units instead of 30 utilising the activation function of PReLU. Furthermore, a batch size of 20 is used. Finally, Adam is employed instead of Adagrad for GRU and CNN, while the Adam learning rate is set to 1e-4 instead of 0.1 in LSTM.

Table 7 proved that the error in the test performance of the MCoRNNMCD-ANN based on the evaluation metrics MSE, MAE, and MSLE was the smallest one in hourly EUR/GBP closing forecasting price outperforming the single benchmarks adjusting with the parameters of the proposed model.

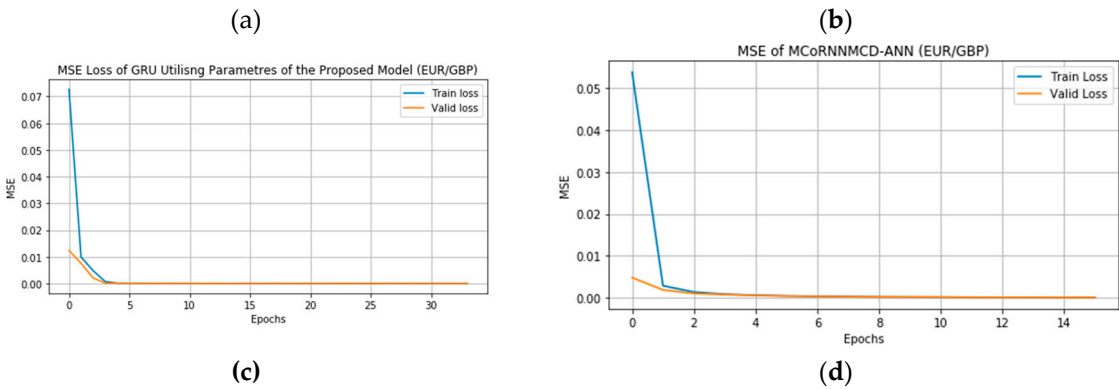
**Table 7.** MCoRNNMCD-ANN performance metrics against adjusted (adj.) single benchmarks.

Model	Metrics	Train	Valid	Test	Time Duration
2D-CNN adj.	MSE	0.00017	0.00016	0.00017	<b>0:39</b>
	MAE	0.00901	0.00897	0.00912	
	MSLE	8.3038e-05	7.3716e-05	7.6177e-05	
GRU adj.	MSE	7.3054e-05	6.6247e-05	6.2578e-05	4:54
	MAE	0.00558	0.00544	0.00536	
	MSLE	3.4814e-05	2.9724e-05	2.8312e-05	
LSTM adj.	MSE	8.7955e-05	8.0258e-05	7.6894e-05	8:43
	MAE	0.00635	0.00621	0.00612	
	MSLE	4.1941e-05	3.6171e-05	3.4807e-05	
MCoRNNMCDANN	MSE	6.5486e-05	6.1332e-05	<b>5.7488e-05</b>	2:35
	MAE	0.00534	0.00526	<b>0.00518</b>	
	MSLE	3.1117e-05	2.7342e-05	<b>2.6051e-05</b>	

More specifically, the test MSE of MCoRNNMCD-ANN decreased to 98.91%, 8.48%, and 28.88% for the test MSE of the 2D-CNN, GRU and LSTM adjusted with the parameters of the proposed MCoRNNMCD-ANN. The test MAE of MCoRNNMCD-ANN decreased to 55.10%, 3.41%, and 16.63% for the test MAE of the 2D-CNN, GRU and LSTM adjusted with the full parameters of the proposed MCoRNNMCD-ANN. The test MSLE of MCoRNNMCD-ANN decreased to 98.06%, 8.32%, and 28.77% for the test MSE of the 2D-CNN, GRU and LSTM adjusted with the parameters of the proposed MCoRNNMCD-ANN. The difference in time elapsed between the proposed MCoRNNMCD-ANN and the benchmark-single models in minutes has also been considered regarding their execution time. As a result, the execution time of MCoRNNMCD-ANN decreased to 139 and 368 for the execution time of the GRU and LSTM modified with full parameters of the proposed MCoRNNMCD-ANN, respectively. The execution time of MCoRNNMCD-ANN was increased to 116 minutes for the time of 2D-CNN utilising the parameters of the proposed MCoRNNMCD-ANN. Based on the results, it has been observed that the 2D-CNN performs faster adjusted with the proposed model parameters despite the timestep being increased to 60. This could result from the Adam optimiser that led to a faster training process instead of the Adagrad in CNN. However, the modified 2D-CNN performed worse than the default 2D-CNN. However, networks using 1D convolutions with MCoRNNMCD-ANN parameters show improvement, displaying lower MSE and confirming the effectiveness of 1D-CNNs for time-series tasks. For the adjusted LSTM and GRU with 60 timesteps, execution time is increased due to retaining information from previous steps and slowing training. Figure 3 shows substantial MSE improvement with proposed MCoRNNMCD-ANN parameters in LSTM and the best GRU-utilised MCoRNNMCD-ANN parameters and MSE of the proposed model's MSE.

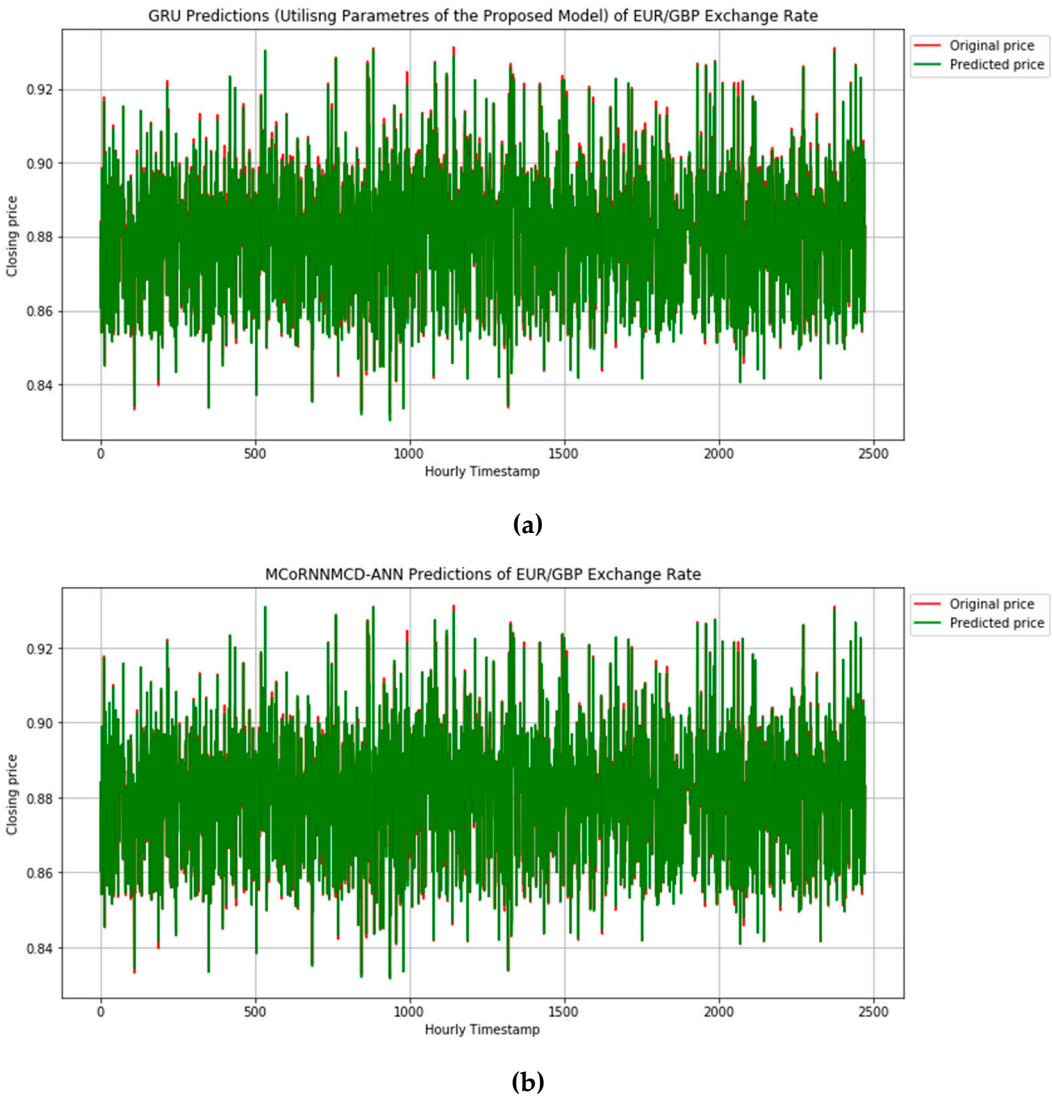






**Figure 3.** MSEs of LSTMs, GRU and the MCoRNNMCD-ANN model: (a) LSTM default (b) LSTM adjusted (c) GRU adjusted (d) Proposed MCoRNNMCD-ANN.

The best predictive method after the MCoRNNMCD-ANN model from the hybrid and single monolithic architectures based on the experimental results that emerged was the GRU adjusted with the parameters of the proposed model. Figure 4 displays the predictions of MCoRNNMCD-ANN showed better performance than the adjusted GRU.



**Figure 4.** Predictions of hourly closing price fluctuation in EUR/GBP of GRU with adjusted parameters of the proposed model and the MCoRNNMCD-ANN. (a) GRU predictions (b) MCoRNNMCD-ANN predictions.

## 5. Conclusions

The study presented a novel neuroscience-informed supplement with the RCT, modular neural network, designed to improve Forex price fluctuation prediction compared to conventional monolithic architectures. This approach divides decision-making into parallel feature extraction modules by integrating modularity, effectively handling complexness within the Forex market. The proposed MCoRNNMCD-ANN combines two modules, CoRNNMCD and CoGRUMCD, featuring a new CNN architecture with a unique adaptation mechanism that combines MCD and orthogonal kernel initialisation to address issues of catastrophic forgetting and vanishing gradients. These modules encompass convolutional layers, orthogonal-initialised RNNs, GRUs coupled with MCD replacing pooling layers, and dense layers concatenating outputs into a final decision-making ANN module. It is worth noting that a proposed modified version of Swish, namely HSwish<sub>m</sub> employed to avoid harsh reactions to minor input Forex price fluctuations.

Experiments revealed MCoRNNMCD-ANN's superior performance with reduced MSE versus hybrid BiCuDNNLSTM, CNN-LSTM, LSTM-GRU, CLSTM and single models 2D-CNN, GRU, LSTM, forecasting EUR/GBP hourly closing price fluctuations. MCoRNNMCD-ANN executed faster than hybrid models using the same parameters. MCoRNNMCD-ANN was also faster than the adjusted single models, except for the time 2D-CNN, which lessened the data's dimensionality due to pooling that retains essential information but can ignore vital data. Proposed parameters boosted benchmarks, except 2D-CNN, which could be suboptimal for time series like 1D convolutional networks.

However, a study limitation is its exclusive focus on EUR/GBP pair. Future research will encompass broader Forex pairs to assess the proposed model's generalizability. Additionally, employing transfer learning using the proposed MCoRNNMCD-ANN to fine-tune ANNs with limited data in the Forex domain holds promise for enhancing predictive capabilities in complex non-linear environments like Forex trading.

**Author Contributions:** Christos Bormpotsis conceptualized, designed, performed the experiments and developed the proposed Modular Convolutional orthogonal Recurrent Neural Network with Monte Carlo Dropout - Artificial Neural Network (MCoRNNMCD-ANN) and the baselines and replicated benchmark algorithms. Mohamed Sedky and Asma Patel provided guidance and direction for the research and evaluation. All authors significantly contributed to the writing and review and agreed to the published version of the manuscript.

**Funding:** Not applicable.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are partly available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mai, Y.; Chen, H.; Zou, J.-Z.; Li, S.-P. Currency Co-Movement and Network Correlation Structure of Foreign Exchange Market. *Physica A: Statistical Mechanics and its Applications* **2018**, *492*, 65–74. <https://doi.org/10.1016/j.physa.2017.09.068>.
2. Hayward, R. Foreign Exchange Speculation: An Event Study. *IJFS* **2018**, *6*, 22. <https://doi.org/10.3390/ijfs6010022>.
3. Ray, R.; Khandelwal, P.; Baranidharan, B. *A Survey on Stock Market Prediction Using Artificial Intelligence Techniques*. In Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT); IEEE: Tirunelveli, India, December 2018; pp. 594–598.

4. Berradi, Z.; Lazaar, M.; Mahboub, O.; Omara, H. *A Comprehensive Review of Artificial Intelligence Techniques in Financial Market*. In Proceedings of the 2020 6th IEEE Congress on Information Science and Technology (CiSt); IEEE: Agadir - Essaouira, Morocco, June 5 2020; pp. 367–371.
5. J. Russin, R. C. O'Reilly, and Y. Bengio. *Deep learning needs a prefrontal cortex*. *Work Bridging AI Cogn Sci*, 107:603–616, 2020.
6. Pujara, M.S.; Wolf, R.C.; Baskaya, M.K.; Koenigs, M. Ventromedial Prefrontal Cortex Damage Alters Relative Risk Tolerance for Prospective Gains and Losses. *Neuropsychologia* **2015**, *79*, 70–75. <https://doi.org/10.1016/j.neuropsychologia.2015.10.026>.
7. Awunyo-Vitor, D. Theoretical and Conceptual Framework of Access to Financial Services by Farmers in Emerging Economies: Implication for Empirical Analysis. *Acta Universitatis Sapientiae, Economics and Business* **2018**, *6*, 43–59. <https://doi.org/10.1515/auseb-2018-0003>.
8. Arnott, D.; Gao, S. Behavioral Economics for Decision Support Systems Researchers. *Decision Support Systems* **2019**, *122*, 113063. <https://doi.org/10.1016/j.dss.2019.05.003>.
9. Buskens, V. Rational Choice Theory in Sociology. In *International Encyclopedia of the Social & Behavioral Sciences*; Elsevier, 2015; pp. 901–906 ISBN 978-0-08-097087-5.
10. Zey, M.A. Rational Choice and Organization Theory. In *International Encyclopedia of the Social & Behavioral Sciences*; Elsevier, 2015; pp. 892–895 ISBN 978-0-08-097087-5.
11. Lerner, J.S.; Li, Y.; Valdesolo, P.; Kassam, K.S. Emotion and Decision Making. *Annu. Rev. Psychol.* **2015**, *66*, 799–823. <https://doi.org/10.1146/annurev-psych-010213-115043>.
12. Rilling, J.K.; Sanfey, A.G. The Neuroscience of Social Decision-Making. *Annu. Rev. Psychol.* **2011**, *62*, 23–48. <https://doi.org/10.1146/annurev.psych.121208.131647>.
13. Lamm, C.; Singer, T. The Role of Anterior Insular Cortex in Social Emotions. *Brain Struct Funct* **2010**, *214*, 579–591. <https://doi.org/10.1007/s00429-010-0251-3>.
14. Eichenbaum, H. Hippocampus. *Neuron* **2004**, *44*, 109–120. <https://doi.org/10.1016/j.neuron.2004.08.028>.
15. LaBar, K.S.; Cabeza, R. Cognitive Neuroscience of Emotional Memory. *Nat Rev Neurosci* **2006**, *7*, 54–64. <https://doi.org/10.1038/nrn1825>.
16. Olsen, R.K.; Moses, S.N.; Riggs, L.; Ryan, J.D. The Hippocampus Supports Multiple Cognitive Processes through Relational Binding and Comparison. *Front. Hum. Neurosci.* **2012**, *6*. <https://doi.org/10.3389/fnhum.2012.00146>.
17. Phelps, E.A.; LeDoux, J.E. Contributions of the Amygdala to Emotion Processing: From Animal Models to Human Behavior. *Neuron* **2005**, *48*, 175–187. <https://doi.org/10.1016/j.neuron.2005.09.025>.
18. Roozendaal, B.; McEwen, B.S.; Chattarji, S. Stress, Memory and the Amygdala. *Nat Rev Neurosci* **2009**, *10*, 423–433. <https://doi.org/10.1038/nrn2651>.
19. Pizzo, F.; Roehri, N.; Medina Villalon, S.; Trébuchon, A.; Chen, S.; Lagarde, S.; Carron, R.; Gavaret, M.; Giusiano, B.; McGonigal, A.; et al. Deep Brain Activities Can Be Detected with Magnetoencephalography. *Nat Commun* **2019**, *10*, 971. <https://doi.org/10.1038/s41467-019-08665-5>.
20. Grossmann, T. The Role of Medial Prefrontal Cortex in Early Social Cognition. *Front. Hum. Neurosci.* **2013**, *7*. <https://doi.org/10.3389/fnhum.2013.00340>.
21. McEwen, B.S.; Bowles, N.P.; Gray, J.D.; Hill, M.N.; Hunter, R.G.; Karatsoreos, I.N.; Nasca, C. Mechanisms of Stress in the Brain. *Nat Neurosci* **2015**, *18*, 1353–1363. <https://doi.org/10.1038/nn.4086>.
22. Price, J.L.; Drevets, W.C. Neurocircuitry of Mood Disorders. *Neuropsychopharmacol* **2010**, *35*, 192–216. <https://doi.org/10.1038/npp.2009.104>.
23. Tsukiura, T.; Shigemune, Y.; Nouchi, R.; Kambara, T.; Kawashima, R. Insular and Hippocampal Contributions to Remembering People with an Impression of Bad Personality. *Social Cognitive and Affective Neuroscience* **2013**, *8*, 515–522. <https://doi.org/10.1093/scan/nss025>.
24. Loued-Khenissi, L.; Pfeuffer, A.; Einhäuser, W.; Preuschoff, K. Anterior Insula Reflects Surprise in Value-Based Decision-Making and Perception. *NeuroImage* **2020**, *210*, 116549. <https://doi.org/10.1016/j.neuroimage.2020.116549>.
25. Ruissen, M.I.; Overgaauw, S.; De Bruijn, E.R.A. Being Right, but Losing Money: The Role of Striatum in Joint Decision Making. *Sci Rep* **2018**, *8*, 6711. <https://doi.org/10.1038/s41598-018-24617-3>.
26. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-Art in Artificial Neural Network Applications: A Survey. *Heliyon* **2018**, *4*, e00938. <https://doi.org/10.1016/j.heliyon.2018.e00938>.
27. Fermin, A.S.R.; Friston, K.; Yamawaki, S. An Insula Hierarchical Network Architecture for Active Interoceptive Inference. *R. Soc. open sci.* **2022**, *9*, 220226. <https://doi.org/10.1098/rsos.220226>.
28. Sim, H.S.; Kim, H.I.; Ahn, J.J. Is Deep Learning for Image Recognition Applicable to Stock Market Prediction? *Complexity* **2019**, *2019*, 1–10. <https://doi.org/10.1155/2019/4324878>.
29. Lanbouri, Z.; Achchab, S. Stock Market Prediction on High Frequency Data Using Long-Short Term Memory. *Procedia Computer Science* **2020**, *175*, 603–608. <https://doi.org/10.1016/j.procs.2020.07.087>.

30. Amer, M.; Maul, T. A Review of Modularization Techniques in Artificial Neural Networks. *Artif Intell Rev* **2019**, *52*, 527–561. <https://doi.org/10.1007/s10462-019-09706-7>.
31. Ali, M.; Sarwar, A.; Sharma, V.; Suri, J. Artificial Neural Network Based Screening of Cervical Cancer Using a Hierarchical Modular Neural Network Architecture (HMNNA) and Novel Benchmark Uterine Cervix Cancer Database. *Neural Comput & Applic* **2019**, *31*, 2979–2993. <https://doi.org/10.1007/s00521-017-3246-7>.
32. Yarushev, S.A.; Averkin, A.N. Time Series Analysis Based on Modular Architectures of Neural Networks. *Procedia Computer Science* **2018**, *123*, 562–567. <https://doi.org/10.1016/j.procs.2018.01.085>.
33. Thakkar, A.; Chaudhari, K. A Comprehensive Survey on Deep Neural Networks for Stock Market: The Need, Challenges, and Future Directions. *Expert Systems with Applications* **2021**, *177*, 114800. <https://doi.org/10.1016/j.eswa.2021.114800>.
34. Sengupta, S.; Basak, S.; Saikia, P.; Paul, S.; Tsalavoutis, V.; Atiah, F.; Ravi, V.; Peters, A. A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends. *Knowledge-Based Systems* **2020**, *194*, 105596. <https://doi.org/10.1016/j.knsys.2020.105596>.
35. Wong, G.; Greenhalgh, T.; Westhorp, G.; Buckingham, J.; Pawson, R. RAMESES Publication Standards: Meta-Narrative Reviews. *BMC Med* **2013**, *11*, 20. <https://doi.org/10.1186/1741-7015-11-20>.
36. Snyder, H. Literature Review as a Research Methodology: An Overview and Guidelines. *Journal of Business Research* **2019**, *104*, 333–339. <https://doi.org/10.1016/j.jbusres.2019.07.039>.
37. Chaddad, A.; Li, J.; Lu, Q.; Li, Y.; Okuwobi, I.P.; Tanougast, C.; Desrosiers, C.; Niazi, T. Can Autism Be Diagnosed with Artificial Intelligence? A Narrative Review. *Diagnostics* **2021**, *11*, 2032. <https://doi.org/10.3390/diagnostics11112032>.
38. Zhang, T.; Schoene, A.M.; Ji, S.; Ananiadou, S. Natural Language Processing Applied to Mental Illness Detection: A Narrative Review. *npj Digit. Med.* **2022**, *5*, 46. <https://doi.org/10.1038/s41746-022-00589-7>.
39. Lerner, J.S.; Li, Y.; Valdesolo, P.; Kassam, K.S. Emotion and Decision Making. *Annu. Rev. Psychol.* **2015**, *66*, 799–823. <https://doi.org/10.1146/annurev-psych-010213-115043>.
40. Butler, M.J.R.; O'Broin, H.L.R.; Lee, N.; Senior, C. How Organizational Cognitive Neuroscience Can Deepen Understanding of Managerial Decision-making: A Review of the Recent Literature and Future Directions. *International Journal of Management Reviews* **2016**, *18*, 542–559. <https://doi.org/10.1111/ijmr.12071>.
41. Adolphs, R. Social Cognition and the Human Brain. *Trends in Cognitive Sciences* **1999**, *3*, 469–479. [https://doi.org/10.1016/S1364-6613\(99\)01399-6](https://doi.org/10.1016/S1364-6613(99)01399-6).
42. Christopoulos, G.I.; Tobler, P.N.; Bossaerts, P.; Dolan, R.J.; Schultz, W. Neural Correlates of Value, Risk, and Risk Aversion Contributing to Decision Making under Risk. *J. Neurosci.* **2009**, *29*, 12574–12583. <https://doi.org/10.1523/JNEUROSCI.2614-09.2009>.
43. Mohr, P.N.C.; Biele, G.; Heekeren, H.R. Neural Processing of Risk. *J. Neurosci.* **2010**, *30*, 6613–6619. <https://doi.org/10.1523/JNEUROSCI.0003-10.2010>.
44. Nieuwenhuis, I.L.C.; Takashima, A. The Role of the Ventromedial Prefrontal Cortex in Memory Consolidation. *Behavioural Brain Research* **2011**, *218*, 325–334. <https://doi.org/10.1016/j.bbr.2010.12.009>.
45. Yoo, S.B.M.; Hayden, B.Y. Economic Choice as an Untangling of Options into Actions. *Neuron* **2018**, *99*, 434–447. <https://doi.org/10.1016/j.neuron.2018.06.038>.
46. Padoa-Schioppa, C. Neurobiology of Economic Choice: A Good-Based Model. *Annu. Rev. Neurosci.* **2011**, *34*, 333–359. <https://doi.org/10.1146/annurev-neuro-061010-113648>.
47. Rangel, A.; Hare, T. Neural Computations Associated with Goal-Directed Choice. *Current Opinion in Neurobiology* **2010**, *20*, 262–270. <https://doi.org/10.1016/j.conb.2010.03.001>.
48. Tzivilaki, A.; Kastellakis, G.; Poirazi, P. Challenging the Point Neuron Dogma: FS Basket Cells as 2-Stage Nonlinear Integrators. *Nat Commun* **2019**, *10*, 3664. <https://doi.org/10.1038/s41467-019-11537-7>.
49. Yao, H.; Zhang, X.; Zhou, X.; Liu, S. Parallel Structure Deep Neural Network Using CNN and RNN with an Attention Mechanism for Breast Cancer Histology Image Classification. *Cancers* **2019**, *11*, 1901. <https://doi.org/10.3390/cancers11121901>.
50. Acunzo, D.J.; Low, D.M.; Fairhall, S.L. Deep Neural Networks Reveal Topic-Level Representations of Sentences in Medial Prefrontal Cortex, Lateral Anterior Temporal Lobe, Precuneus, and Angular Gyrus. *NeuroImage* **2022**, *251*, 119005. <https://doi.org/10.1016/j.neuroimage.2022.119005>.
51. Sadiq, S.; Umer, M.; Ullah, S.; Mirjalili, S.; Rupapara, V.; Nappi, M. Discrepancy Detection between Actual User Reviews and Numeric Ratings of Google App Store Using Deep Learning. *Expert Systems with Applications* **2021**, *181*, 115111. <https://doi.org/10.1016/j.eswa.2021.115111>.
52. Flesch, T.; Juechems, K.; Dumbalska, T.; Saxe, A.; Summerfield, C. Orthogonal Representations for Robust Context-Dependent Task Performance in Brains and Neural Networks. *Neuron* **2022**, *110*, 1258–1270.e11. <https://doi.org/10.1016/j.neuron.2022.01.005>.
53. Baek, Y.; Kim, H.Y. ModAugNet: A New Forecasting Framework for Stock Market Index Value with an Overfitting Prevention LSTM Module and a Prediction LSTM Module. *Expert Systems with Applications* **2018**, *113*, 457–480. <https://doi.org/10.1016/j.eswa.2018.07.019>.



54. Lee, S.W.; Kim, H.Y. Stock Market Forecasting with Super-High Dimensional Time-Series Data Using ConvLSTM, Trend Sampling, and Specialized Data Augmentation. *Expert Systems with Applications* **2020**, *161*, 113704. <https://doi.org/10.1016/j.eswa.2020.113704>.
55. Galeshchuk, S.; Mukherjee, S. Deep Networks for Predicting Direction of Change in Foreign Exchange Rates. *Intell Sys Acc Fin Mgmt* **2017**, *24*, 100–110. <https://doi.org/10.1002/isaf.1404>.
56. Shiao, Y.C.; Chakraborty, G.; Chen, S.F.; Hua Li, L.; Chen, R.C. Modeling and Prediction of Time-Series-A Case Study with Forex Data. In Proceedings of the 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST); IEEE: Morioka, Japan, October 2019; pp. 1–5.
57. Maneejuk, P.; Srichaikul, W. Forecasting Foreign Exchange Markets: Further Evidence Using Machine Learning Models. *Soft Comput* **2021**, *25*, 7887–7898. <https://doi.org/10.1007/s00500-021-05830-1>.
58. Hossain, M.A.; Karim, R.; Thulasiram, R.; Bruce, N.D.B.; Wang, Y. Hybrid Deep Learning Model for Stock Price Prediction. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI); IEEE: Bangalore, India, November 2018; pp. 1837–1844.
59. Althelaya, K.A.; El-Alfy, E.-S.M.; Mohammed, S. Evaluation of Bidirectional LSTM for Short-and Long-Term Stock Market Prediction. In Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS); IEEE: Irbid, April 2018; pp. 151–156.
60. Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity* **2020**, *2020*, 1–10. <https://doi.org/10.1155/2020/6622927>.
61. Alonso-Monsalve, S.; Suárez-Cetrulo, A.L.; Cervantes, A.; Quintana, D. Convolution on Neural Networks for High-Frequency Trend Prediction of Cryptocurrency Exchange Rates Using Technical Indicators. *Expert Systems with Applications* **2020**, *149*, 113250. <https://doi.org/10.1016/j.eswa.2020.113250>.
62. Kanwal, A.; Lau, M.F.; Ng, S.P.H.; Sim, K.Y.; Chandrasekaran, S. BiCuDNNLSTM-1dCNN — A Hybrid Deep Learning-Based Predictive Model for Stock Price Prediction. *Expert Systems with Applications* **2022**, *202*, 117123. <https://doi.org/10.1016/j.eswa.2022.117123>.
63. Pokhrel, N.R.; Dahal, K.R.; Rimal, R.; Bhandari, H.N.; Khatri, R.K.C.; Rimal, B.; Hahn, W.E. Predicting NEPSE Index Price Using Deep Learning Models. *Machine Learning with Applications* **2022**, *9*, 100385. <https://doi.org/10.1016/j.mlwa.2022.100385>.
64. Islam, Md.S.; Hossain, E.; Rahman, A.; Hossain, M.S.; Andersson, K. A Review on Recent Advancements in FOREX Currency Prediction. *Algorithms* **2020**, *13*, 186. <https://doi.org/10.3390/a13080186>.
65. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. *Applied Soft Computing* **2020**, *90*, 106181. <https://doi.org/10.1016/j.asoc.2020.106181>.
66. Ellefsen, K.O.; Mouret, J.-B.; Clune, J. Neural Modularity Helps Organisms Evolve to Learn New Skills without Forgetting Old Skills. *PLoS Comput Biol* **2015**, *11*, e1004128. <https://doi.org/10.1371/journal.pcbi.1004128>.
67. Duan, H.; Wang, X. Echo State Networks With Orthogonal Pigeon-Inspired Optimization for Image Restoration. *IEEE Trans. Neural Netw. Learning Syst.* **2016**, *27*, 2413–2425. <https://doi.org/10.1109/TNNLS.2015.2479117>.
68. Miok, K.; Nguyen-Doan, D.; Škrlj, B.; Zaharie, D.; Robnik-Šikonja, M. Prediction Uncertainty Estimation for Hate Speech Classification. In *Statistical Language and Speech Processing*; Martín-Vide, C., Purver, M., Pollak, S., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2019; Vol. 11816, pp. 286–298 ISBN 978-3-030-31371-5.
69. Barcellos, S.H.; Zamarro, G. Unbanked Status and Use of Alternative Financial Services among Minority Populations. *Journal of Pension Economics and Finance* **2021**, *20*, 468–481. <https://doi.org/10.1017/S1474747219000052>.
70. Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; Chang, Y. Abusive Language Detection in Online User Content. In Proceedings of the Proceedings of the 25th International Conference on World Wide Web; International World Wide Web Conferences Steering Committee: Montréal Québec Canada, April 11 2016; pp. 145–153.
71. Yang, J.; Counts, S. Predicting the Speed, Scale, and Range of Information Diffusion in Twitter. *ICWSM* **2010**, *4*, 355–358. <https://doi.org/10.1609/icwsml.v4i1.14039>.
72. Chen, C.-C.; Kuo, C.; Kuo, S.-Y.; Chou, Y.-H. Dynamic Normalization BPN for Stock Price Forecasting. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics; IEEE: Kowloon Tong, Hong Kong, October 2015; pp. 2855–2860.
73. Almasri, E.; Arslan, E. Predicting Cryptocurrencies Prices with Neural Networks. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT); IEEE: Istanbul, Turkey, October 2018; pp. 1–5.
74. Walczak, S. An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. *Journal of Management Information Systems* **2001**, *17*, 203–222. <https://doi.org/10.1080/07421222.2001.11045659>.

75. Hutto, C.; Gilbert, E. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *ICWSM* **2014**, *8*, 216–225. <https://doi.org/10.1609/icwsml.v8i1.14550>.
76. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J Big Data* **2021**, *8*, 53. <https://doi.org/10.1186/s40537-021-00444-8>.
77. Aryal, S.; Nadarajah, D.; Kasthurirathna, D.; Rupasinghe, L.; Jayawardena, C. Comparative Analysis of the Application of Deep Learning Techniques for Forex Rate Prediction. In Proceedings of the 2019 International Conference on Advancements in Computing (ICAC); IEEE: Malabe, Sri Lanka, December 2019; pp. 329–333.
78. Liu, S.; Ji, H.; Wang, M.C. Nonpooling Convolutional Neural Network Forecasting for Seasonal Time Series With Trends. *IEEE Trans. Neural Netw. Learning Syst.* **2020**, *31*, 2879–2888. <https://doi.org/10.1109/TNNLS.2019.2934110>.
79. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the Icml; 2010.
80. Golmohammadi, M.; Ziyabari, S.; Shah, V.; Von Weltin, E.; Campbell, C.; Obeid, I.; Picone, J. Gated Recurrent Networks for Seizure Detection. In Proceedings of the 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB); IEEE, 2017; pp. 1–5.
81. Dong, Y.; Wang, J.; Guo, Z. Research and Application of Local Perceptron Neural Network in Highway Rectifier for Time Series Forecasting. *Applied Soft Computing* **2018**, *64*, 656–673. <https://doi.org/10.1016/j.asoc.2017.12.022>.
82. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. **2015**. <https://doi.org/10.48550/ARXIV.1506.02142>.
83. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for Activation Functions. *arXiv preprint arXiv:1710.05941* **2017**.
84. Courbariaux, M.; Bengio, Y.; David, J.-P. BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations. **2015**. <https://doi.org/10.48550/ARXIV.1511.00363>.
85. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. **2014**. <https://doi.org/10.48550/ARXIV.1412.6980>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.