Financial time series forecasting with deep learning : A systematic literature review: 2005–2019

Article *in* Applied Soft Computing · February 2020

DOI: 10.1016/j.asoc.2020.106181

3 authors:

Omer Berat Sezer
TOBB University of Economics and Technology
19 PUBLICATIONS **2,990** CITATIONS

Ugur Gudelek
TOBB University of Economics and Technology
16 PUBLICATIONS **1,993** CITATIONS

Murat Ozbayoglu
TOBB University of Economics and Technology
143 PUBLICATIONS **5,184** CITATIONS

Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019

Omer Berat Sezer[a], M. Ugur Gudelek[a], Ahmet Murat Ozbayoglu[a]

[a]*Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey*

Abstract

Financial time series forecasting is undoubtedly the top choice of computational intelligence for finance researchers in both academia and the finance industry due to its broad implementation areas and substantial impact. Machine Learning (ML) researchers have created various models, and a vast number of studies have been published accordingly. As such, a significant number of surveys exist covering ML studies on financial time series forecasting. Lately, Deep Learning (DL) models have appeared within the field, with results that significantly outperform their traditional ML counterparts. Even though there is a growing interest in developing models for financial time series forecasting, there is a lack of review papers that solely focus on DL for finance. Hence, the motivation of this paper is to provide a comprehensive literature review of DL studies on financial time series forecasting implementation. We not only categorized the studies according to their intended forecasting implementation areas, such as index, forex, and commodity forecasting, but we also grouped them based on their DL model choices, such as Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), and Long-Short Term Memory (LSTM). We also tried to envision the future of the field by highlighting its possible setbacks and opportunities for the benefit of interested researchers.

Keywords: deep learning, finance, computational intelligence, machine learning, time series forecasting, CNN, LSTM, RNN

1. Introduction

The finance industry has always been interested in the successful prediction of financial time series data. Numerous studies have been published on ML models with relatively better performances than classical time series forecasting techniques. Meanwhile, the widespread application of automated electronic trading systems coupled with increasing demand for higher yields keeps forcing researchers and practitioners to continue working on implementing better models. Hence, new publications and implementations keep adding to the finance and computational intelligence literature.

In the last few years, DL has strongly emerged as the best performing predictor class within the ML field in various implementation areas. Financial time series forecasting is no exception, and as such, an increasing number of prediction models based on various DL techniques have been introduced in the appropriate conferences and journals in recent years.

Despite the vast number of survey papers covering financial time series forecasting and trading systems using traditional soft computing techniques, to the best of our knowledge, no reviews have been performed on the literature for DL. Hence, we decided to work on such a comprehensive study, focusing on DL implementations of financial time series forecasting. Our motivation is two-fold; we not only aimed at providing a state-of-the-art snapshot of academic and industry perspectives of developed DL models, but we also pinpoint the important and distinctive characteristics of each studied model to prevent researchers and practitioners from making unsatisfactory choices during their system development. We also wanted to envision where the industry is heading by indicating possible future directions.

Our fundamental motivation was to answer the following research questions:

- Which DL models are used for financial time series forecasting ?
- How does the performance of DL models compare with that of their traditional ML counterparts ?
- What is the future direction of DL research for financial time series forecasting ?

Our focus was solely on DL implementations for financial time series forecasting. For other DL-based financial applications, such as risk assessment and portfolio management, interested readers can refer to another recent survey paper [1]. Because we wanted to single out financial time series prediction studies in our survey, we omitted other time series forecasting studies that were not focused on financial data. Meanwhile, we included time-series research papers that had financial use cases or examples, even if the papers themselves were not directly concerned with financial time series forecasting. Also, we decided to include algorithmic trading papers that were based on financial forecasting but ignore the ones that did not have a time series forecasting component.

We mainly reviewed journals and conferences for our survey, but we also included Masters and PhD theses, book chapters, arXiv papers, and noteworthy technical publications that came up in web searches. We decided to only include articles published in English language.

During our survey, we realized that most of the papers using the term “deep learning" in their description were published in the past five years. However, we also encountered some older studies that implemented deep models, such as Recurrent Neural Networks (RNNs) and Jordan-Elman networks. However, at their time of publication, the term “deep learning" was not in common usage. Therefore, we decided to also include those papers.

According to our findings, this will be one of the first comprehensive “financial time series forecasting" survey papers focusing on DL. Many ML reviews for financial time series forecasting exist in the literature, but we have not encountered any study on DL. Hence, we wanted to fill this gap by analyzing the developed models and applications accordingly. We hope that as a result of this paper, researchers and model developers will have a better idea of how they can implement DL models in their studies.

The remainder of this paper is structured as follows. In Section 2, existing surveys focused on ML and soft computing studies for financial time series forecasting are mentioned. In Section 3, we will cover existing DL models that are used, such as CNN, LSTM, and Deep

Reinforcement Learning (DRL). Section 4 will focus on the various financial time series forecasting implementation areas using DL, namely stock forecasting, index forecasting, trend forecasting, commodity forecasting, volatility forecasting, foreign exchange forecasting, and cryptocurrency forecasting. In each subsection, the problem definition will be given, followed by the particular DL implementations. In Section 5, overall statistical results about our findings will be presented, including histograms related to the annual distributions of different subfields, models, publication types, etc. A state-of-the-art snapshot of financial time series forecasting studies will be given through these statistics. At the same time, they will also show the areas that are already mature in comparison with promising or new areas that still have room for improvement. Section 6 discusses the academic and industrial achievements that have been accomplished and future expectations. The section will include highlights of open areas that require further research. Finally, we conclude this paper in Section 7 by summarizing our findings.

2. Financial Time Series Forecasting with ML

Financial time series forecasting and associated applications have been studied extensively for many years. When ML started gaining popularity, financial prediction applications based on soft computing models accordingly also became available. Even though our particular focus is on DL implementations of financial time series prediction studies, it will be beneficial to briefly mention existing surveys covering ML-based financial time series forecasting studies to provide some historical perspective.

In our study, we did not include any survey papers that were focused on specific financial application areas other than forecasting studies. However, we were faced with some review publications that included a mix of financial time-series studies and other financial applications. We decided to include those papers to maintain the comprehensiveness of our coverage.

Examples of these aforementioned publications are provided here. There were published books on stock market forecasting [2], trading system development [3], practical examples of forex and market forecasting applications [4] using ML models, such as Artificial Neural Networks (ANNs), Evolutionary Computations (ECs), and Genetic Programming (GP), and Agent-based models [5].

There were also some existing journal and conference surveys. Bahrammirzaee et al. [6] surveyed financial prediction and planning studies along with other financial applications using various Artificial Intelligence (AI) techniques such as ANN, Expert Systems, and hybrid models. Zhang et al. [7] also compared ML methods in different financial applications, including stock market prediction studies. In Mochon et al. [8], soft computing models for the market, forex prediction, and trading systems were analyzed. Mullainathan and Spies [9] surveyed the prediction process in general from an econometric perspective.

There were also a number of survey papers concentrated on one particular ML model. Even though these papers focused on one technique, the implementation areas generally spanned various financial applications, including financial time series forecasting. Among those soft computing methods, EC and ANN have had the most overall interest.

In terms of EC studies, Chen wrote a book on Genetic Algorithms (GAs) and GP in Computational Finance [10]. Later, Multiobjective Evolutionary Algorithms (MOEAs) were extensively surveyed for various financial applications including financial time series prediction [11, 12, 13]. Meanwhile, Rada reviewed EC applications along with Expert Systems for financial investing models [14].

In terms of ANN studies, Li and Ma reviewed implementations of ANN for stock price forecasting and some other financial applications [15]. Tkac et al. [16] surveyed different implementations of ANN in financial applications, including stock price forecasting. Recently, Elmsili and Outtaj surveyed ANN applications in economics and management research, including economic time series forecasting [17].

There have also been several text mining surveys focused on financial applications, including financial time series forecasting. Mittermayer and Knolmayer compared various text mining implementations that extract market responses to news for prediction [18]. Mitra et al. [19] focused on news analytics studies for prediction of abnormal returns for trading strategies in their survey. Nassirtoussi et al. reviewed text mining studies for stock or forex market prediction [20]. Kearney et al. [21] also surveyed text mining-based time series forecasting and trading strategies using textual sentiment. Similarly, Kumar and Ravi [22] reviewed text mining studies for forex and stock market prediction. Lately, Xing et al. [23] surveyed natural language-based financial forecasting studies.

Finally, there were application-specific survey papers that focused on particular financial time series forecasting implementations. Among these studies, stock market forecasting had the most interest. A number of surveys were published for stock market forecasting studies based on various soft computing methods at different times [24, 25, 26, 27, 28, 29, 30, 31]. Chatterjee et al. [32] and Katarya and Mahajan [33] concentrated on ANN-based financial market prediction studies, whereas Hu et al. [34] focused on EC implementations for stock forecasting and algorithmic trading models. In a different time series forecasting application, researchers surveyed forex prediction studies using ANN [35] and various other soft computing techniques [36].

Although many surveys exist for ML implementations of financial time series forecasting, DL has not yet been surveyed comprehensively despite the emergence of various DL implementations in recent years. This was the main motivation for our survey. In the next section, we cover the various DL models used in financial time series forecasting studies.

3. Deep Learning

DL is a type of ANN that consists of multiple processing layers and enables high-level abstraction to model data. The key advantage of DL models is extracting the good features of input data automatically using a general-purpose learning procedure. Therefore, DL models have been proposed for many applications such as: image, speech, video, and audio reconstruction, natural language understanding (particularly topic classification), sentiment analysis, question answering, and language translation [37]. The historical improvements of DL models are surveyed in Schmidhuber et al. [38].

Financial time series forecasting has been very popular among ML researchers for more than 40 years. The financial community has been boosted by the recent introduction of DL models for financial prediction and their accompanying publications. The success of DL over ML models is the major attractive point for finance researchers. With more financial time series data and different deep architectures, new DL methods will be proposed. In our survey, the vast majority of studies found DL models to be better than their ML counterparts.

In the literature, there are different kinds of DL models: Deep Multilayer Perceptron (DMLP), RNN, LSTM, CNN, Restricted Boltzmann Machines (RBMs), DBN, Autoencoder (AE), and DRL [37, 38]. Throughout the literature, financial time series forecasting is mostly considered as a regression problem. However, there is also a significant number of studies, particularly on trend prediction, that use classification models to tackle financial forecasting problems. In Section 4, different DL implementations are presented along with their model choices.

3.1. Deep Multi Layer Perceptron (DMLP)

DMLP was one of the first developed ANNs. Its difference from shallow nets is that DMLP contains more layers. Even though particular model architectures might have variations depending on different problem requirements, DMLP models consist mainly of three layers: input, hidden, and output. The number of neurons in each layer and the number of layers are the hyperparameters of the network. In general, each neuron in the hidden layers has input (x), weight (w), and bias (b) terms. In addition, each neuron has a nonlinear activation function, which produces a cumulative output of the preceding neurons. Equation 1 [39] illustrates the output of a single neuron in the Neural Network (NN). There are different types of nonlinear activation functions. The most commonly used nonlinear activation functions are sigmoid (Equation 2) [40], hyperbolic tangent (Equation 3) [41], Rectified Linear Unit (ReLU) (Equation 4) [42], leaky-ReLU (Equation 5) [43], swish (Equation 6) [44], and softmax (Equation 7) [39]. Non-linear activations were compared in [44].

$$y_i = \sigma(\sum_i W_i x_i + b_i) \tag{1}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2}$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{3}$$

$$R(z) = \max(0, z) \tag{4}$$

$$R(z) = 1(x < 0)(\alpha x) + 1(x >= 0)(x) \tag{5}$$

$$f(x) = x\sigma(\beta x) \tag{6}$$

$$\mathrm{softmax}(z_i) = \frac{\exp z_i}{\sum_j \exp z_j} \tag{7}$$

DMLP models have appeared in various application areas [45, 37] . Using a DMLP model has advantages and disadvantages depending on the problem requirements. Through DMLP models, problems such as regression and classification can be solved by modeling the input data [46]. However, if the number of input features is increased (e.g., image as input), the parameter size in the network will increase accordingly due to the fully connected nature of the model, which will jeopardize the computational performance and create storage problems. To overcome this issue, different types of Deep Neural Network (DNN) methods have been proposed (such as CNN) [37]. With DMLP, much more efficient classification and regression processes can be performed. In Figure 1, a DMLP model's layers, neurons, and weights between neurons are illustrated.

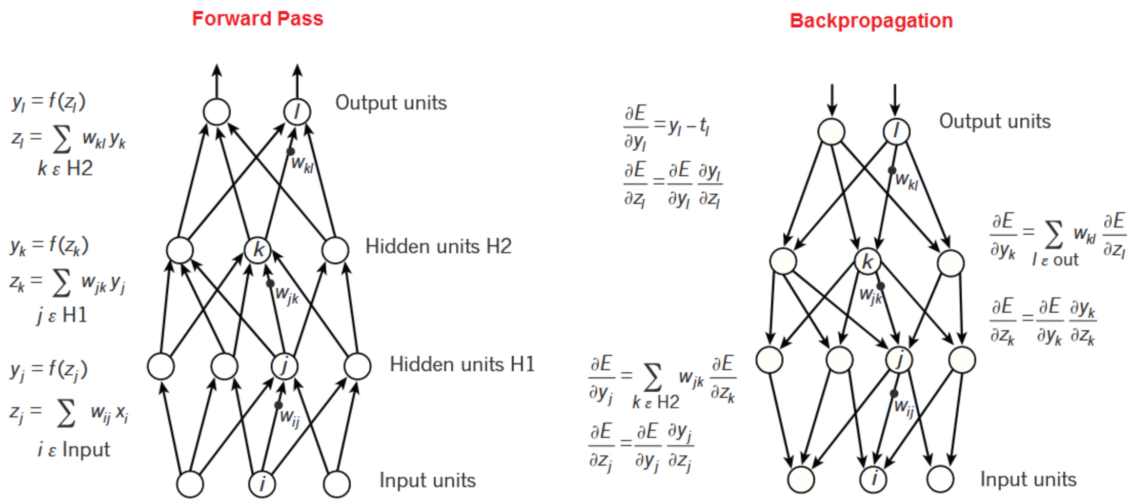

Figure 1: Deep Multi Layer Neural Network Forward Pass and Backpropagation [37]

The DMLP learning stage is implemented through backpropagation. The error in the neurons in the output layer is propagated back to the preceeding layers. Optimization algorithms are used to find the optimum parameters/variables of the NNs. They are used to update the weights of the connections between the layers. Different optimization algorithms have been developed: Stochastic Gradient Descent (SGD), SGD with Momentum, Adaptive Gradient Algorithm (AdaGrad), Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (ADAM) [47, 48, 49, 50, 51]. Gradient descent is an iterative method to find optimum parameters of the function that minimizes the cost function. SGD is an algorithm that randomly selects a few samples for each iteration instead of the whole data set [47]. SGD with Momentum remembers the update in each iteration, which accelerates gradient descent [48]. AdaGrad is a modified SGD that improves on the convergence performance of the standard SGD algorithm [49]. RMSProp is an optimization algorithm that adapts the learning rate for each parameter. In RMSProp, the learning rate is divided by a running average of the magnitudes of recent gradients for that weight [50]. ADAM is an updated version of RMSProp that uses running averages of both the gradients and second moments of the gradients. ADAM combines the advantages of RMSProp (works well in online and non-stationary settings) and AdaGrad (works well with sparse gradients) [51].

As shown in Figure 1, the effect of backpropagation is transferred to the previous layers. If the effect of SGD is gradually lost when the effect reaches the early layers during backpropagation, this problem is called the vanishing gradient problem [52]. In this case, updates between the early layers become unavailable and the learning process stops. The high number of layers in the neural network and the increasing complexity cause the vanishing gradient problem.

The important issue in the DMLP are the hyperparameters of the networks and method of tuning these hyperparameters. Hyperparameters are the variables of the network that affect the network architecture and performance of the networks. The number of hidden layers, number of units in each layer, regularization techniques (dropout, L1, L2), network weight initialization (zero, random, He [53], Xavier [54]), activation functions (Sigmoid, ReLU, hyperbolic tangent, etc.), learning rate, decay rate, momentum values, number of epochs, batch size (minibatch size), and optimization algorithms (SGD, AdaGrad, RMSProp, ADAM, etc.) are the hyperparameters of DMLP. Choosing better hyperparameter values/variables for the network results in better performance. Therefore, finding the best hyperparameters for the network is a significant issue. In the literature, there are different methods to find best hyperparameters: Manual Search (MS), Grid Search (GS), RandomSearch (RS), and Bayesian Methods (Sequential Model-Based Global Optimization (SMBGO), The Gaussian Process Approach (GPA), Tree-structured Parzen Estimator Approach (TSPEA)) [55, 56].

3.2. Recurrent Neural Network (RNN)

The RNN is another type of DL network used for time series or sequential data, such as language and speech. RNNs are also used in traditional ML models (Back Propagation Through Time (BPTT), Jordan-Elman networks, etc.); however, the time periods in such models are generally less than those used in deep RNN models. Deep RNNs are preferred due to their ability to include longer time periods. Unlike Fully Connected Neural Networks (FNNs), RNNs use internal memory to process incoming inputs. RNNs are used to analyze time series data in various fields (handwriting recognition, speech recognition, etc.). As stated in the literature, RNNs are good at predicting the next character in text, language translation applications, and sequential data processing [45, 37].

The RNN model architecture consists of different numbers of layers and different types of units in each layer. The main difference between RNN and FNN is that each RNN unit takes the current and previous input data at the same time. The output depends on the previous data in the RNN model. RNNs process input sequences one by one at any given time during their operation. The units in the hidden layer hold information about the history of the input in the “state vector". When the output of the units in the hidden layer is divided into different discrete time steps, an RNN is converted into a DMLP [37]. In Figure 2, the information flow in the RNN's hidden layer is divided into discrete times. The status of the node S at different times of t is shown as s_t, the input value x at different times is x_t, and the output value o at different times is shown as o_t. Parameter values (U, W, V) are always used in the same step.

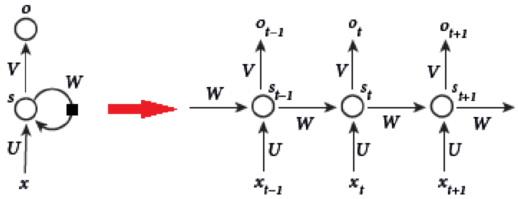

Figure 2: RNN cell through time[37]

RNNs can be trained using the BPTT algorithm. Optimization algorithms (SGD, RMSProp, ADAM) are used for the weight adjustment process. With the BPTT learning method, the error change at time t is reflected in the input and weights of the previous t times. The difficulty of training an RNN is that the RNN structure has a backward dependence over time. Therefore, RNNs become increasingly complex as the learning period increases. Although the main aim of using an RNN is to learn long-term dependencies, studies in the literature show that when knowledge is stored for long time periods, it is not easy to learn with an RNN [57]. To solve this particular problem, LSTMs with different structures of ANN have been developed [37]. Equations 8 and 9 illustrate simpler RNN formulations. Equation 10 shows the total error, which is the sum of the errors from each time iteration[1].

$$h_t = Wf(h_{t-1}) + W^{(hx)}x_{[t]} \tag{8}$$

$$y_t = W^{(S)}f(h_t) \tag{9}$$

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W} \tag{10}$$

Hyperparameters of the RNN also define the network architecture, and the performance of the network is affected by the parameter choices, as in DMLP case. The number of hidden layers, number of units in each layer, regularization techniques, network weight initialization, activation functions, learning rate, momentum values, number of epochs, batch size (minibatch size), decay rate, optimization algorithms, model (Vanilla RNN, Gated-Recurrent Unit (GRU), LSTM), and sequence length are the hyperparameters of RNN. Finding the best hyperparameters for the network is a significant issue. In the literature, there are different methods to find the best hyperparameters: MS, GS, RS, and Bayesian Methods (SMBGO, GPA, TSPEA) [55, 56].

[1]Richard Socher, CS224d: Deep Learning for Natural Language Processing, Lecture Notes

3.3. Long Short Term Memory (LSTM)

LSTM [58] is a type of RNN where the network can remember both short term and long term values. LSTM networks are the preferred choice of many DL model developers when tackling complex problems such as automatic speech and handwritten character recognition. LSTM models are mostly used with time-series data. Their applications include Natural Language Processing (NLP), language modeling, language translation, speech recognition, sentiment analysis, predictive analysis, and financial time series analysis [59, 60]. With attention modules and AE structures, LSTM networks can be more successful in time series data analysis, such as language translation [59].

LSTM networks consist of LSTM units. LSTM units merge to form an LSTM layer. An LSTM unit is composed of cells, each with an input gate, output gate, and forget gate. These gates regulate the information flow. With these features, each cell remembers the desired values over arbitrary time intervals. Equations 11-15 show the form of the forward pass of the LSTM unit [58] (x_t: input vector to the LSTM unit, f_t: forget gate's activation vector, i_t: input gate's activation vector, o_t: output gate's activation vector, h_t: output vector of the LSTM unit, c_t: cell state vector, σ_g: sigmoid function, σ_c , σ_h: hyperbolic tangent function, $*$: element-wise (Hadamard) product, W , U: weight matrices to be learned, b: bias vector parameters to be learned) [60].

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{11}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{12}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{13}$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{14}$$

$$h_t = o_t * \sigma_h(c_t) \tag{15}$$

LSTM is a specialized version of RNN. Therefore, the weight updates and preferred optimization methods are the same. In addition, the hyperparameters of LSTM are just like those of RNN: number of hidden layers, number of units in each layer, network weight initialization, activation functions, learning rate, momentum values, number of epochs, batch size (minibatch size), decay rate, optimization algorithms, sequence length for LSTM, gradient clipping, gradient normalization, and dropout[60, 61]. To find the best hyperparameters of LSTM, the hyperparameter optimization methods used for RNN are also applicable to LSTM [55, 56].

3.4. Convolutional Neural Networks (CNNs)

The CNN is a type of DNN that consists of convolutional layers based on the convolutional operation. It is the most common model used for vision and image processing-based classification problems (image classification, object detection, image segmentation, etc.) [62, 63, 64]. The advantage of the CNN is the number of parameters compared to vanilla DL models, such as DMLP. Filtering with the kernel window function gives the advantage of image processing to CNN architectures with fewer parameters, which is beneficial for computing and storage. In CNN architectures, there are different layers: convolutional, max-pooling, dropout, and fully connected Multilayer Perceptron (MLP) layer. The convolutional layer consists of a convolution (filtering) operation. A basic convolution operation is shown in Equation 16, where t denotes time, s denotes feature map, w denotes kernel, x denotes input, and a denotes variable. In addition, the convolution operation is implemented on two-dimensional images. Equation 17 shows the convolution operation for a two-dimensional image, where I denotes input image, K denotes the kernel, (m, n) denotes image dimensions, and i and j denote variables. Consecutive convolutional and max-pooling layers construct the deep network. Equation 18 describes the NN architecture, where W denotes weights, x denotes input, b denotes bias, and z denotes the output of neurons. At the end of the network, the softmax function is used to obtain the output. Equations 19 and 20 illustrate the softmax function, where y denotes output [39].

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{16}$$

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n). \tag{17}$$

$$z_i = \sum_{j} W_{i,j}\, x_j + b_i. \tag{18}$$

$$y = \mathrm{softmax}(z) \tag{19}$$

$$\mathrm{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j} \exp(z_j)} \tag{20}$$

The backpropagation process is used for CNN model learning. Most commonly used optimization algorithms (SGD, RMSProp) are used to find optimum CNN parameters. Hyperparameters of the CNN are similar to other DL model hyperparameters: number of hidden layers, number of units in each layer, network weight initialization, activation functions, learning rate, momentum values, number of epochs, batch size (minibatch size), decay rate, optimization algorithms, dropout, kernel size, and filter size. To find the best CNN hyperparameters, the following search algorithms are commonly used: MS, GS, RS, and Bayesian methods. [55, 56].

3.5. Restricted Boltzmann Machines (RBMs)

An RBM is a productive stochastic ANN that can learn a probability distribution on the input set [65]. RBMs are mostly used for unsupervised learning [66]. RBMs are used in applications such as dimension reduction, classification, feature learning, and collaborative filtering [67]. The advantage of RBMs is their ability to find hidden patterns in an unsupervised manner. The disadvantage of RBMs is its difficult training process. "RBMs are tricky because although there are good estimators of the log-likelihood gradient, there are no known cheap ways of estimating the log-likelihood itself" [68].

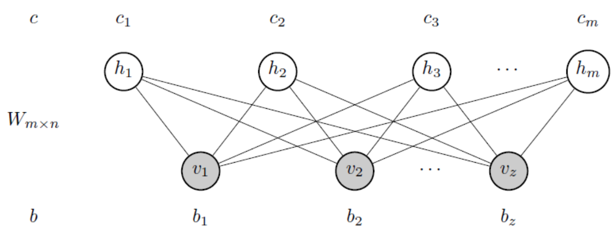

Figure 3: RBM Visible and Hidden Layers [65]

An RBM is a two-layer, bipartite, and undirected graphical model that consists of two layers: visible and hidden (Figure 3). The layers are not connected among themselves. Each cell is a computational point that processes the input and makes stochastic decisions about whether this nerve node will transmit the input. Inputs are multiplied by specific weights, certain threshold values (bias) are added to input values, and then the calculated values are passed through an activation function. In the reconstruction stage, the results from the outputs re-enter the network as input before finally exiting the visible layer as output. The values of the previous input and values after the processes are compared. The purpose of this comparison is to reduce the difference.

Equation 21 illustrates the probabilistic semantics for an RBM using its energy function, where P denotes the probabilistic semantics for an RBM, Z denotes the partition function, E denotes the energy function, h denotes hidden units, and v denotes visible units. Equation 22 illustrates the partition function or normalizing constant. Equation 23 shows the energy of a configuration (in matrix notation) of a standard RBM with binary-valued hidden and visible units, where a denotes bias weights (offsets) for the visible units, b denotes bias weights for the hidden units, W denotes matrix weight of the connection between hidden and visible units, T denotes the transpose of matrix, v denotes visible units, and h denotes hidden units [69, 70].

$$P(v,h) = \frac{1}{Z} \exp(-E(v,h)) \tag{21}$$

$$Z = \sum_{v} \sum_{h} \exp(-E(v,h)) \tag{22}$$

$$E(v, h) = -a^T v - b^T h - v^T W h \tag{23}$$

The learning is performed multiple times on the network [65]. The training of RBMs is implemented by minimizing the negative log-likelihood of the model and data. The Contrastive Divergence (CD) algorithm is used as the stochastic approximation algorithm, which replaces the model expectation using an estimation using Gibbs Sampling with a limited number of iterations [66]. In the CD algorithm, the Kullback Leibler Divergence (KL-Divergence) algorithm is used to measure the distance between its reconstructed probability distribution and the original probability distribution of the input [71].

Momentum, learning rate, weight-cost (decay rate), batch size (minibatch size), regularization method, number of epochs, number of layers, initialization of weights, size of visible units, size of hidden units, type of activation units (sigmoid, softmax, ReLU, Gaussian units), loss function, and optimization algorithms are the hyperparameters of RBMs. Similar to other deep networks, the hyperparameters are searched with MS, GS, RS, and Bayesian methods (Gaussian process). In addition to these, Annealed Importance Sampling (AIS) is used to estimate the partition function. The CD algorithm is also used for the optimization of RBMs [55, 56, 72, 73].

3.6. Deep Belief Networks (DBNs)

A DBN is a type of deep ANN consisting of a stack of RBM networks (Figure 4). A DBN is a probabilistic generative model that consists of latent variables. In a DBN, there is no link between units in each layer. DBNs are used to find discriminate independent features in the input set using unsupervised learning [69]. The ability to encode higher-order network structures and fast inference are the advantages of DBNs [74]. DBNs have the same disadvantages as RBMs because DBNs are composed of RBMs.

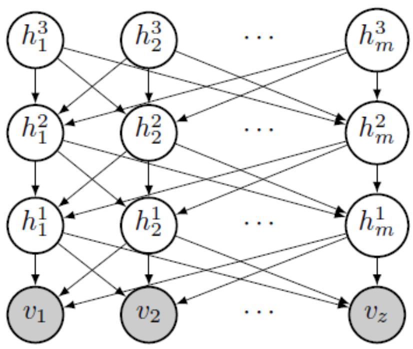

Figure 4: Deep Belief Network [65]

When a DBN is trained in an unsupervised manner, it can learn to reconstruct the input set in a probabilistic manner. Then, the layers in the network begin to detect discriminating features in the input. After this learning step, supervised learning is conducted for classification [75]. Equation 24 illustrates the probability of generating a visible vector (W: matrix weight of connection between hidden unit h and visible unit v, $p(h|W)$: prior distribution over hidden vectors) [69].

$$p(v) = \sum_{h} p(h|W)p(v|h, W) \tag{24}$$

The DBN training process can be divided into two steps: stacked RBM learning and backpropagation learning. In stacked RBM learning, an iterative CD algorithm is used [66]. In backpropagation learning, optimization algorithms (SGD, RMSProp, ADAM) are used to train the network [74]. The hyperparameters of a DBNs are similar to thoseof an RBM. Momentum, learning rate, weight-cost (decay rate), regularization method, batch size (minibatch size), number of epochs, number of layers, initialization of weights, number of RBM stacks, size of visible units in RBMs' layers, size of hidden units in RBMs' layers, type of units (sigmoid, softmax, rectified, Gaussian units, etc.), network weight initialization, and optimization algorithms are the hyperparameters of DBNs. Similar to other deep networks, the hyperparameters are searched with MS, GS, RS, and Bayesian methods. The CD algorithm is also used for the optimization of DBNs [55, 56, 72, 73].

3.7. Autoencoders (AEs)

AE networks are ANNs used as unsupervised learning models. In addition, AE networks are commonly used in DL models, wherein they remap the inputs (features) such that the inputs are more representative for classification. In other words, AE networks perform an unsupervised feature learning process, which fits very well with the DL framework. A representation of a data set is learned by reducing the dimensionality with AEs. AEs are similar to Feedforward Neural Networks (FFNNs)' in their architecture. They consist of an input layer, an output layer, and one or more hidden layers that connect them together. The number of nodes in the input layer and the number of nodes in the output layer are equal to each other in AEs, and they have a symmetrical structure. The most notable advantages of AEs are dimensionality reduction and feature learning. Meanwhile, reducing dimensionality and feature extraction in AEs cause some drawbacks. Focusing on minimizing the loss of data relationship in the encoding of AEs causes the loss of some significant data relationships. Hence, this may be considered as a drawback of AEs[76].

In general, AEs contain two components: encoder and decoder. The input $x \in [0, 1]^d$ is converted through function $f(x)$ (W_1 denotes a weight matrix, b_1 denotes a bias vector, σ_1 element-wise sigmoid activation function of the encoder). Output h is the encoded part of the AEs (code), latent variables, or latent representation. The inverse of function $f(x)$, called function $g(h)$, produces the reconstruction of output r (W_2 denotes a weight matrix, b_2 denotes a bias vector, and σ_2 is an element-wise sigmoid activation function of the decoder). Equations 25 and 26 illustrate the simple AE process [77]. Equation 27 shows the loss function of the AE, the Mean Squared Error (MSE). In the literature, AEs have been used for feature extraction and dimensionality reduction [39, 77].

$$h = f(x) = \sigma_1(W_1 x + b_1) \tag{25}$$

$$r = g(h) = \sigma_2(W_2 h + b_2) \tag{26}$$

$$L(x, r) = ||x - r||^2 \tag{27}$$

AEs are a specialized version of FFNNs. The backpropagation learning is used for updating the weights in the network [39]. Optimization algorithms (SGD, RMSProp, ADAM) are used for the learning process of AEs. MSE is used as a loss function in AEs. In addition, recirculation algorithms may also be used for the training of AEs [39]. AEs' hyperparameters are similar to those of DL hyperparameters. Learning rate, weight-cost (decay rate), dropout fraction, batch size (minibatch size), number of epochs, number of layers, number of nodes in each encoder layer, type of activation functions, number of nodes in each decoder layers, network weight initialization, optimization algorithms, and number of nodes in the code layer (size of latent representation) are the hyperparameters of AEs. Similar to other deep networks, the hyperparameters are searched with MS, GS, RS, and Bayesian methods [55, 56].

3.8. Deep Reinforcement Learning (DRL)

Reinforcement learning (RL) is a type of learning that differs from supervised and unsupervised learning models. It does not require a preliminary data set that has been labeled or clustered before. RL is an ML approach inspired by learning action/behavior, which deals with what actions should be taken by subjects to achieve the highest reward in an environment. There are different areas in which it is used: game theory, control theory, multi-agent systems, operations research, robotics, information theory, investment portfolio management, simulation-based optimization, playing Atari games, and statistics [78]. Some advantages of using RL for control problems are that an agent can be easily re-trained to adapt to changes in the environment and that the system is continually improved while training is constantly performed. An RL agent learns by interacting with its surroundings and observing the results of these interactions. This learning method mimics the basics of how humans learn.

RL is mainly based on a Markov Decision Process (MDP). A MDP is used to formalize the RL environment. A MDP consists of five tuples: state (finite set of states), action (finite set of actions), reward function (scalar feedback signal), state transition probability matrix ($p(s', r|s, a)$, where s' denotes next state, r denotes reward function, s denotes state, and a denotes action), and discount factor (γ, present value of future rewards). The aim of the agent is to maximize the cumulative reward. The return (G_t) is the total discounted reward. Equation 28 illustrates the total return, where R denotes rewards, t denotes time, and k denotes a variable in time.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{28}$$

The value function is the prediction of future values. It provides information on how good the state/action is. Equation 29 illustrates the formulation of the value function, where $v(s)$ denotes the value function, $E[.]$ denotes the expectation function, G_t denotes the total discounted reward, s denotes the given state, R denotes the rewards, S denotes the set of states, and t denotes time.

$$v(s) = E[G_t|S_t = s] = E[R_{t+1} + \gamma v(S_{t+1})|S_t = s] \tag{29}$$

Policy (π) is the agent's behavior strategy. It is like a map from state to action. There are two types of value functions to express the actions in the policy: state-value function ($v_\pi(s)$) and action-value function ($q_\pi(s, a)$). The state-value function (Equation 30) is the expected return of starting from s to following policy π ($E_\pi[.]$ denotes expectation function). The action-value function (Equation 31) is the expected return of starting from s and taking action a to following policy π (A denotes the set of actions and a denotes the given action).

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \tag{30}$$

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] \tag{31}$$

The optimal state-value function (Equation 32) is the maximum value function over all policies. The optimal action-value function (Equation 33) is the maximum action-value function over all policies.

$$v_*(s) = \max(v_\pi(s)) \tag{32}$$

$$q_*(s, a) = \max(q_\pi(s, a)) \tag{33}$$

The RL solutions and methods in the literature are too broad to review in this paper. Therefore, we summarize the important issues of RL and important RL solutions and methods. RL methods can mainly be divided into two types: model-based methods and model-free methods. Model-based methods use a model that is known by the agent before, value/policy, and experience. The experience can be real (sample from the environment) or simulated (sample from the model). Model-based methods are mostly used in the applications of robotics and control algorithms [79]. Model-free methods are mainly divided into two groups: value-based and policy-based. In value-based methods, a policy is produced directly from the value function (e.g., epsilon-greedy). In policy-based methods, the policy is parametrized directly. In value-based methods, there are three main solutions for MDP problems: Dynamic Programming (DP), Monte Carlo (MC), and Temporal Difference (TD).

In the DP method, problems are solved with optimal substructure and overlapping subproblems. The full model is known and is used for planning in MDP. There are two iterations (learning algorithms) in DP: policy iteration and value iteration. The MC method learns experience directly by running an episode of game/simulation. MC is a type of model-free method that does not require MDP transitions/rewards. It collects states and takes the mean of returns for the value function. TD is also a model-free method that learns the experience directly by running the episode. In addition, TD learns incomplete episodes like the DP method by using bootstrapping. The TD method combines the MC and DP methods. SARSA (state, action, reward, state, action; S_t, A_t, R_t, S_{t+1}, A_{t+1}) is a type of TD control algorithm. Q-value (action-value function) is updated with the agent actions. It is an on-policy learning model that learns from actions according to the current policy π. Equation 34 illustrates the update of the action-value function in the SARSA algorithm, where S_t denotes current state, A_t denotes current action, t denotes time, R denotes reward, α denotes learning rate, γ denotes discount factor. Q-learning is another TD control algorithm. It is an off-policy learning model that learns from different actions that do not require the policy π at all. Equation 35 illustrates the update of the action-value function in the Q-Learning algorithm (the whole algorithm is described in [78], a' denotes action).

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R(t+1) + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \tag{34}$$

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R(t+1) + \gamma max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)] \tag{35}$$

In the value-based methods, a policy can be generated directly from the value function (e.g., using epsilon-greedy). Policy-based methods use the policy directly instead of using the value function. It has both advantages and disadvantages over value-based methods. The policy-based methods are more effective in high-dimensional or continuous action spaces and have better convergence properties than value-based methods. It can also learn stochastic policies. On the other hand, policy-based methods evaluate a policy that is typically inefficient and has high variance. It typically converges to a local rather than global optimum. In the policy-based methods, there are also different solutions: Policy gradient, Reinforce (Monte-Carlo Policy Gradient), and Actor-Critic [78] (details of policy-based methods can be found in [78]).

DRL methods contain NNs. Therefore, DRL hyperparameters are similar to the DL hyperparameters. Learning rate, weight-cost (decay rate), dropout fraction, regularization method, batch size (minibatch size), number of epochs, number of layers, number of nodes in each layer, type of activation functions, network weight initialization, optimization algorithms, discount factor, and number of episodes are the hyperparameters of DRL. Similar to other deep networks, the hyperparameters are searched with MS, GS, RS and Bayesian methods [55, 56].

4. Financial Time Series Forecasting

The most widely studied financial application area is forecasting of a given financial time series, particularly asset price forecasting. Even though some variations exist, the main focus is on predicting the next movement of the underlying asset. More than half of the existing implementations of DL are focused on this area. Even though there are several subtopics of this general problem, including stock price forecasting, index prediction, forex price prediction, commodity (oil, gold, etc.) price prediction, bond price forecasting, volatility forecasting, cryptocurrency price forecasting, the underlying dynamics are the same in all of these applications.

Studies can also be clustered into two main groups based on their expected outputs: price prediction and price movement (trend) prediction. Although price forecasting is essentially a regression problem, in most financial time series forecasting applications, correct price prediction of the price is not perceived to be as important as correctly identifying the directional movement. As a result, researchers consider trend prediction, i.e., forecasting which way the price will change, a more crucial study area compared with exact price prediction. In that sense, trend prediction becomes a classification problem. In some studies, only up or down movements are taken into consideration (2-class problem), although 3-class problems also exist (up, down, or neutral movements).

LSTM and its variations along with some hybrid models dominate the financial time series forecasting domain. LSTM, by its nature, utilizes the temporal characteristics of any time series signal; hence, forecasting financial time series is a well-studied and successful implementation of LSTM. However, some researchers prefer to either extract appropriate features from the time series or transform the time series such that the resulting financial data become stationary from a temporal perspective, meaning even if we shuffle the data order, we will still be able to properly train the model and achieve successful out-of-sample test performance. For those implementations, CNN and Deep Feedforward Neural Network (DFNN) are the most commonly chosen DL models.

Various financial time series forecasting implementations using DL models exist in literature. We will cover each of them in the following subsections. In this survey paper, we examine the papers using the following criteria: First, we group articles according to their subjects. Then, we group related papers according to their feature set. Finally, we group each subgroup according to DL models/methods.

For each implementation area, the related papers are subgrouped and tabulated. Each table contains the following fields to provide information about the implementation details for the papers within the group: Article (Art.) and Data Set are trivial, Period refers to the time period for training and testing. Feature Set lists the input features used in the study. Lag is the time length of the input vector (e.g., 30d means the input vector has a 30 day window), and horizon shows how far into the future the model predicts. Some abbreviations are used for the two aforementioned fields: min is minutes, h is hours, d is days, w is weeks, m is months, y is years, s is steps, and * is mixed. Method shows the DL models that are used in the study. Performance criteria provides the evaluation metrics, and Environment (Env.) lists the development framework/software/tools. Some column values might be empty, indicating there was no relevant information in the paper for the corresponding field.

4.1. Stock Price Forecasting

Price prediction of any given stock is the most studied financial application of all. We observed the same trend within DL implementations. Depending on the prediction time horizon, different input parameters are chosen, varying from High Frequency Trading (HFT) and intraday price movements to daily, weekly, or even monthly stock close prices. Also, technical, fundamental analysis, social media feeds, and sentiment are among the different parameters used for the prediction models.

Table 1: Stock Price Forecasting Using Only Raw Time Series Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[80]	38 stocks in KOSPI	2010-2014	Lagged stock returns	50min	5min	DNN	NMSE, RMSE, MAE, MI	-
[81]	China stock market, 3049 Stocks	1990-2015	OCHLV	30d	3d	LSTM	Accuracy	Theano, Keras
[82]	Daily returns of 'BRD' stock in Romanian Market	2001-2016	OCHLV	-	1d	LSTM	RMSE, MAE	Python, Theano
[83]	297 listed companies of CSE	2012-2013	OCHLV	2d	1d	LSTM, SRNN, GRU	MAD, MAPE	Keras
[84]	5 stock in NSE	1997-2016	OCHLV, Price data, turnover and number of trades.	200d	1..10d	LSTM, RNN, CNN, MLP	MAPE	-
[85]	Stocks of Infosys, TCS and CIPLA from NSE	2014	Price data	-	-	RNN, LSTM and CNN	Accuracy	-
[86]	10 stocks in S&P500	1997-2016	OCHLV, Price data	36m	1m	RNN, LSTM, GRU	Accuracy, Monthly return	Keras, Tensorflow
[87]	Stocks data from S&P500	2011-2016	OCHLV	1d	1d	DBN	MSE, norm-RMSE, MAE	-
[88]	High-frequency transaction data of the CSI300 futures	2017	Price data	-	1min	DNN, ELM, RBF	RMSE, MAPE, Accuracy	Matlab
[89]	Stocks in the S&P500	1990-2015	Price data	240d	1d	DNN, GBT, RF	Mean return, MDD, Calmar ratio	H2O
[90]	ACI Worldwide, Staples, and Seagate in NASDAQ	2006-2010	Daily closing prices	17d	1d	RNN, ANN	RMSE	-
[91]	Chinese Stocks	2007-2017	OCHLV	30d	1..5d	CNN + LSTM	Annualized Return, Mxm Retracement	Python
[92]	20 stocks in S&P500	2010-2015	Price data	-	-	AE + LSTM	Weekly Returns	-
[93]	S&P500	1985-2006	Monthly and daily log-returns	*	1d	DBN+MLP	Validation, Test Error	Theano, Python, Matlab
[94]	12 stocks from SSE Composite Index	2000-2017	OCHLV	60d	1..7d	DWNN	MSE	Tensorflow
[95]	50 stocks from NYSE	2007-2016	Price data	-	1d, 3d, 5d	SFM	MSE	-

In this survey, we grouped first stock price forecasting articles according to their feature sets, such as studies using only the raw time series data (price data, Open, Close, High, Low, Volume (OCHLV)) for price prediction; studies using various other data, and studies using text mining techniques. Regarding the first group, the corresponding DL models were directly implemented using raw time series for price prediction. Table 1 tabulates the stock price forecasting studies that used only raw time series data in the literature. In Table 1, different methods/models are also listed based on four sub-groups: DNN (networks that are deep but without any given topology details) and LSTM models, multi models, hybrid models, novel methods.

DNN and LSTM models were solely used in 3 papers. In Chong et al. [80], DNN and lagged stock returns were used to predict the stock prices in The Korea Composite Stock Price Index (KOSPI). Chen et al. [81], and Dezsi and Nistor [82] applied raw price data as the input to LSTM models.

Meanwhile, some studies implement multiple DL models for performance comparison using only raw price (OCHLV) data for forecasting. Among the noteworthy studies, Samarawickrama et al. [83] compared RNN, Stacked Recurrent Neural Network (SRNN), LSTM, and GRU. Hiransha et. al. [84] compared LSTM, RNN, CNN, and MLP, whereas in Selvin et al. [85], RNN, LSTM, CNN, and Autoregressive Integrated Moving Average (ARIMA) were preferred. Lee and Yoo [86] compared 3 RNN models (SRNN, LSTM, GRU) for stock price prediction and then constructed a threshold-based portfolio selecting stocks according to predictions. Li et. al. [87] implemented DBN. Finally, the authors of [88] compared 4 different ML models for next price prediction in 1-minute price data: a 1 DL model (AE and RBM), MLP, Radial Basis Function Neural Network (RBF) and Extreme Learning Machine (ELM). They also compared the results for different sized datasets. The authors of [89] used price data and DNN, Gradient Boosted Trees (GBT), and Random Forest (RF) methods for the prediction of stocks in the Standard's & Poor's 500 Index (S&P500). Chandra and Chan [90] used co-operative neuro-evolution, RNN (Elman network), and DFNN for the prediction of stock prices in National Association of Securities Dealers Automated Quotations (NASDAQ) (ACI Worldwide, Staples, and Seagate).

Meanwhile, hybrid models were used in some papers. Liu et al. [91] applied CNN+LSTM. Heaton et al. [92] implemented smart indexing with AE. Batres et al. [93] combined DBN and MLP to construct a stock portfolio by predicting each stock's monthly log-return and choosing only stocks that were expected to perform better than the median stock.

In addition, novel approaches were adapted in some studies. Yuan et al. [94] proposed the novel Deep and Wide Neural Network (DWNN), which is combination of RNN and CNN. Zhang et al. [95] implemented a State Frequency Memory (SFM) recurrent network.

In another group of studies, some researchers again focused on LSTM-based models. However, their input parameters came from various sources including raw price data, technical and/or fundamental analysis, macroeconomic data, financial statements, news, and investor sentiment. Table 2 summarizes these stock price forecasting papers. In Table 2, different methods/models are also listed based on five sub-groups: DNN model; LSTM and RNN models; multiple and hybrid models; CNN model; and novel methods.

DNN models were used in some stock price forecasting papers within this group. In Abe et al. [96], a DNN model and 25 fundamental features were used for prediction of Japan Index constituents. Feng et al. [97] also used fundamental features and a DNN model for prediction. A DNN model and macro economic data, such as GDP, unemployment rate, and inventories, were used by the authors of [98] for the prediction of U.S. low-level disaggregated macroeconomic time series.

LSTM and RNN models were chosen in some studies. Kraus and Feuerriegel [99] implemented LSTM with transfer learning using text mining through financial news and stock market data. Similarly, Minami et al. [100] used LSTM to predict stock's next day price using corporate action events and macro-economic index. Zhang and Tan [101] implemented DeepStockRanker, an LSTM-based model for stock ranking using 11 technical indicators. In Zhuge et al. [102], the authors used the price time series and emotional data from text posts to predict the opening stock price of the next day with an LSTM network. Akita et al. [103] used textual information and stock prices through Paragraph Vector + LSTM for forecasting prices and the comparisons were provided with different classifiers. Ozbayoglu

[104] used technical indicators along with stock data on a Jordan-Elman network for price prediction.

There were also multiple and hybrid models that used mostly technical analysis features as their inputs to the DL model. Several technical indicators were fed into LSTM and MLP networks in Khare et al. [105] for intraday price prediction. Recently, Zhou et al. [106] used a GAN for minimizing Forecast error loss and Direction prediction loss (GAN-FD) model for stock price prediction and compared their model performances against ARIMA, ANN and Support Vector Machine (SVM). Singh et al. [107] used several technical indicator features and time series data with Principal Component Analysis (PCA) for dimensionality reduction cascaded with a DNN (2-layer FFNN) for stock price prediction. Karaoglu et al. [108] used market microstructure-based trade indicators as inputs into an RNN with Graves LSTM detecting the buy-sell pressure of movements in the Istanbul Stock Exchange Index (BIST) to perform price prediction for intelligent stock trading. In Zhou et al. [109], next month's return was predicted, and next-to-be-performed portfolios were constructed. Good monthly returns were achieved with LSTM and LSTM-MLP models.

Meanwhile, in some papers, CNN models were preferred. Abroyan et al. [110] used 250 features, including order details, for the prediction of a private brokerage company's real data of risky transactions. They used CNN and LSTM for stock price forecasting. The authors of [111] used a CNN model and fundamental, technical, and market data for prediction.

Novel methods were also developed in some studies. In Tran et al. [112], with the FI-2010 dataset, bid/ask and volume were used as the feature set for forecasting. In the study, they proposed Weighted Multichannel Time-series Regression (WMTR), and Multilinear Discriminant Analysis (MDA). Feng et al. [113] used 57 characteristic features, including Market equity, Market Beta, Industry momentum, and Asset growth, as inputs to a Fama-French n-factor DL for predicting monthly US equity returns in New York Stock Exchange (NYSE), American Stock Exchange (AMEX), or NASDAQ.

Table 2: Stock Price Forecasting Using Various Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[96]	Japan Index constituents from WorldScope	1990-2016	25 Fundamental Features	10d	1d	DNN	Correlation, Accuracy, MSE	Tensorflow
[97]	Return of S&P500	1926-2016	Fundamental Features:	-	1s	DNN	MSPE	Tensorflow
[98]	U.S. low-level disaggregated macroeconomic time series	1959-2008	GDP, Unemployment rate, Inventories, etc.	-	-	DNN	R^2	-
[99]	CDAX stock market data	2010-2013	Financial news, stock market data	20d	1d	LSTM	MSE, RMSE, MAE, Accuracy, AUC	TensorFlow, Theano, Python, Scikit-Learn
[100]	Stock of Tsugami Corporation	2013	Price data	-	-	LSTM	RMSE	Keras, Tensorflow
[101]	Stocks in China's A-share	2006-2007	11 technical indicators	-	1d	LSTM	AR, IR, IC	-
[102]	SCI prices	2008-2015	OCHL of change rate, price	7d	-	EmotionalAnalysis + LSTM	MSE	-

Table 2: Stock Price Forecasting Using Various Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[103]	10 stocks in Nikkei 225 and news	2001-2008	Textual information and Stock prices	10d	-	Paragraph Vector + LSTM	Profit	-
[104]	TKC stock in NYSE and QQQQ ETF	1999-2006	Technical indicators, Price	50d	1d	RNN (Jordan-Elman)	Profit, MSE	Java
[105]	10 Stocks in NYSE	-	Price data, Technical indicators	20min	1min	LSTM, MLP	RMSE	-
[106]	42 stocks in China's SSE	2016	OCHLV, Technical Indicators	242min	1min	GAN (LSTM, CNN)	RMSRE, DPA, GAN-F, GAN-D	-
[107]	Google's daily stock data	2004-2015	OCHLV, Technical indicators	20d	1d	$(2D)^2$ PCA + DNN	SMAPE, PCD, MAPE, RMSE, HR, TR, R^2	R, Matlab
[108]	GarantiBank in BIST, Turkey	2016	OCHLV, Volatility, etc.	-	-	PLR, Graves LSTM	MSE, RMSE, MAE, RSE, R^2	Spark
[109]	Stocks in NYSE, AMEX, NASDAQ, TAQ intraday trade	1993-2017	Price, 15 firm characteristics	80d	1d	LSTM+MLP	Monthly return, SR	Python,Keras, Tensorflow in AWS
[110]	Private brokerage company's real data of risky transactions	-	250 features: order details, etc.	-	-	CNN, LSTM	F1-Score	Keras, Tensorflow
[111]	Fundamental and Technical Data, Economic Data	-	Fundamental , technical and market information	-	-	CNN	-	-
[112]	The LOB of 5 stocks of Finnish Stock Market	2010	FI-2010 dataset: bid/ask and volume	-	*	WMTR, MDA	Accuracy, Precision, Recall, F1-Score	-
[113]	Returns in NYSE, AMEX, NASDAQ	1975-2017	57 firm characteristics	*	-	Fama-French n-factor model DL	R^2, RMSE	Tensorflow

A number of research papers have also used text mining techniques for feature extraction but used non-LSTM models for stock price prediction. Table 3 summarizes the stock price forecasting papers that used text mining techniques. In Table 3, different methods/models are clustered into three sub-groups: CNN and LSTM models; GRU, LSTM, and RNN models; and novel methods.

CNN and LSTM models were adapted in some of the papers. In Ding et al. [114], events were detected from Reuters and Bloomberg news through text mining, and that information was used for price prediction and stock trading through the CNN model. Vargas et al. [115] used text mining on S&P500 index news from Reuters through an LSTM+CNN hybrid model for price prediction and intraday directional movement estimation together. Lee et al. [116] used financial news data and implemented word embedding with Word2vec along with MA and stochastic oscillator to create inputs for a Recurrent CNN (RCNN) for stock price prediction. Iwasaki et al. [117] also used sentiment analyses through text mining and word embeddings from analyst reports and used sentiment features as inputs to a DFNN model for stock price prediction. Then, different portfolio selections were implemented based on the projected stock returns.

GRU, LSTM, and RNN models were preferred in the next group of papers. Das et al. [118] implemented sentiment analysis on Twitter posts along with stock data for price forecasting using an RNN. Similarly, the authors of [119] used sentiment classification (neutral, positive, and negative) for opening or closing stock price prediction with various LSTM models. They compared their results with SVM and achieved higher overall performance. In Zhongshengz et al. [120], text and price data were used for the prediction of SSE Composite Index (SCI) prices.

Novel approaches were reported in some papers. Nascimento et al. [121] used word embeddings for extracting information from web pages and then combined it with stock price data for stock price prediction. They compared the Autoregressive (AR) model and RF with and without news. The results showed embedding news information improved the performance. Han et al. [122] used financial news and the ACE2005 Chinese corpus. Different event types of Chinese companies were classified based on a novel event-type pattern classification algorithm in Han et al. [122], and also next day stock price change was also predicted using additional inputs.

Table 3: Stock Price Forecasting Using Text Mining Techniques for Feature Extraction

Art.	**Data Set**	**Period**	**Feature Set**	**Lag**	**Horizon**	**Method**	**Performance Criteria**	**Env.**
[114]	S&P500 Index, 15 stocks in S&P500	2006-2013	News from Reuters and Bloomberg	-	-	CNN	Accuracy, MCC	-
[115]	S&P500 index news from Reuters	2006-2013	Financial news titles, Technical indicators	1d	1d	RCNN	Accuracy	-
[116]	TWSE index, 4 stocks in TWSE	2001-2017	Technical indicators, Price data, News	15d	-	CNN + LSTM	RMSE, Profit	Keras, Python, TALIB
[117]	Analyst reports on the TSE and Osaka Exchange	2016-2018	Text	-	-	LSTM, CNN, Bi-LSTM	Accuracy, R-squared	R, Python, MeCab
[118]	Stocks of Google, Microsoft and Apple	2016-2017	Twitter sentiment and stock prices	-	-	RNN	-	Spark, Flume, Twitter API,
[119]	Stocks of CSI300 index, OCHLV of CSI300 index	2009-2014	Sentiment Posts, Price data	1d	1d	Naive Bayes + LSTM	Precision, Recall, F1-score, Accuracy	Python, Keras
[120]	SCI prices	2013-2016	Text data and Price data	7d	1d	LSTM	Accuracy, F1-Measure	Python, Keras
[121]	Stocks from S&P500	2006-2013	Text (news) and Price data	7d	1d	LAR+News, RF+News	MAPE, RMSE	-
[122]	News from Sina.com, ACE2005 Chinese corpus	2012-2016	A set of news text	-	-	Their unique algorithm	Precision, Recall, F1-score	-

4.2. Index Forecasting

Instead of trying to forecast the price of a single stock, several researchers preferred to predict the stock market index. Indexes are generally are less volatile than individual stocks because they are composed of multiple stocks from different sectors and are more indicative of the overall momentum and general state of the economy.

In the literature, different stock market index data have been used for experiments. The most commonly used index data are as follows: S&P500, China Securities Index (CSI)300, National Stock Exchange of India (NIFTY), Tokyo Nikkei Index (NIKKEI)225, Dow Jones Industrial Average (DJIA), Shanghai Stock Exchange (SSE)180, Hong Kong Hang Seng

Index (HSI), Shenzhen Stock Exchange Composite Index (SZSE), London Financial Times Stock Exchange Index (FTSE)100, Taiwan Capitalization Weighted Stock Index (TAIEX), BIST, NASDAQ, Dow Jones Industrial Average 30 (DOW30), KOSPI, S&P500 Volatility Index (VIX), NASDAQ100 Volatility Index (VXN), Brazilian Stock Exchange (Bovespa), Stockholm Stock Exchange (OMX), and NYSE. The authors of the papers [123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 114] used S&P500 as their dataset. The authors of the studies [123, 124, 135, 136, 137] used NIKKEI as their dataset. KOSPI was used in Li et al. [135], Jeong et al. [131], Baek et al. [132]. DJIA was used as the dataset in the papers [123, 136, 137, 138, 139]. The authors of the articles [123, 135, 137, 131] used HSI as the dataset in their stuedies, and SZSE was used in the studies [140, 135, 141, 142].

In addition, in the literature, there are different methods for the prediction of index data. While some studies used only raw time series data, others used various other data such, as technical indicators, index data, social media feeds, news from Reuters, and Bloomberg, and statistical features of data (standard deviation, skewness, kurtosis, omega ratio, fund alpha). In this survey, we first grouped the index forecasting articles according to their feature sets such as studies using only raw time series data (price/index data, OCHLV); then, we clustered the studies using various other data. Table 4 summarizes the index forecasting papers using only raw time series data. Moreover, different methods (models) were used for index forecasting. MLP, RNN, LSTM, and DNN (DFNN or DMLP) methods were the most used methods for index forecasting. In Table 4, these various methods/models are also listed as four sub-groups: ANN, DNN, MLP, and Fuzzy Deep Direct Reinforcement Learning (FDDR) models; RL and DL models; LSTM and RNN models; and novel methods.

Table 4: Index Forecasting Using Only Raw Time Series Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[124]	S&P500, Nikkei225, USD Exchanges	2011-2015	Index data	-	1d, 5d, 7d, 10d	LRNFIS with Firefly-Harmony Search	RMSE, MAPE, MAE	-
[125]	S&P500 Index	1989-2005	Index data, Volume	240d	1d	LSTM	Return, STD, SR, Accuracy	Python, TensorFlow, Keras, R, H2O
[127]	S&P500, VIX	2005-2016	Index data	*	1d	uWN, cWN	MASE, HIT, RMSE	-
[128]	S&P500 Index	2010-2017	Index data	10d	1d, 30d	Stacked LSTM, Bi-LSTM	MAE, RMSE, R-squared	Python, Keras, Tensorflow
[131]	S&P500, KOSPI, HSI, and EuroStoxx50	1987-2017	200-days stock price	200d	1d	Deep Q-Learning and DNN	Total profit, Correlation	-
[132]	S&P500, KOSPI200, 10-stocks	2000-2017	Index data	20d	1d	ModAugNet: LSTM	MSE, MAPE, MAE	Keras
[133]	S&P500, Bovespa50, OMX30	2009-2017	Autoregressive part of the time series	-	1d	LSTM	MSE, Accuracy	Tensorflow, Keras, R
[134]	S&P500	2000-2017	Index data	-	1..4d, 1w, 1..3m	GLM, LSTM+RNN	MAE, RMSE	Python

Table 4: Index Forecasting Using Only Raw Time Series Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[136]	Nikkei225, IXIC, HSI, GSPC, DJIA	1985-2018	OCHLV	5d	1d	LSTM	RMSE	Python, Keras, Theano
[138]	DJIA	-	Index data	-	-	Genetic Deep Neural Network	MSE	Java
[139]	Log returns of the DJIA	1971-2002	Index data	20d	1d	RNN	TR, sign rate, PT/HM test, MSFE, SR, profit	-
[140]	Shanghai A-shares composite index, SZSE	2006-2016	OCHLV	10d	-	Embedded layer + LSTM	Accuracy, MSE	Python, Matlab, Theano
[141]	300 stocks from SZSE, Commodity	2014-2015	Index data	-	-	FDDR, DNN + RL	Profit, return, SR, profit-loss curves	Keras
[142]	Shanghai composite index and SZSE	1990-2016	OCHLV	20d	1d	Ensembles of ANN	Accuracy	-
[143]	TUNINDEX	2013-2017	Log returns of index data	-	5min	DNN with hierarchical input	Accuracy, MSE	Java
[144]	Singapore Stock Market Index	2010-2017	OCHL of last 10 days of index	10d	3d	Feed-forward DNN	RMSE, MAPE, Profit, SR	-
[145]	BIST	1990-2002	Index data	7d	1d	MLP, RNN, MoE	HIT, positive/negative HIT, MSE, MAE	-
[146]	SCI	2012-2017	OCHLV, Index data	-	1..10d	Wavelet + LSTM	MAPE, theil unequal coefficient	-
[147]	S&P500	1950-2016	Index data	15d	1d	LSTM	RMSE	Keras
[148]	ISE100	1987-2008	Index data	-	2d, 4d, 8d, 12d, 18d	TAR-VEC-MLP, TAR-VEC-RBF, TAR-VEC-RHE	RMSE	-
[149]	VIX, VXN, VXD	2002-2014	First five autoregressive lags	5d	1d, 22d	HAR-GASVR	MAE, RMSE	-

ANN, DNN, MLP, and FDDR models were used in some studies. In Lachiheb et al. [143], log returns of the index data were used with a DNN with hierarchical input for the prediction of TUNINDEX data. Yong et al. [144] used a deep FFNN and Open, Close, High, Low (OCHL) of the last 10 days of index data for prediction. In addition, MLP and ANN were used for the prediction of index data. In Yumlu et al. [145], raw index data were used with MLP, RNN, Mixture of Experts (MoE), and Exponential GARCH (EGARCH) for forecasting. In Yang et al. [142], ensembles of ANN with OCHLV of data were used for prediction of the Shanghai composite index.

Furthermore, RL and DL methods were used together for prediction of index data in some studies. In Deng et al.[141], FDDR, DNN, and RL methods were used to predict 300 stocks from SZSE index data and commodity prices. In Jeong et al. [131], Deep Q-Learning and DNN methods and a 200-day stock price dataset were used together for prediction of the S&P500 index.

Most of the preferred methods for prediction of index data using raw time series data have been based on LSTM and RNN. In Bekiros et al. [139], an RNN was used for prediction of log returns of the DJIA index. In Fischer et al. [125], LSTM was used to predict

S&P500 index data. Althelaya et al. [128] used stacked LSTM and Bidirectional LSTM (Bi-LSTM) methods for S&P500 index forecasting. Yan et al. [146] used an LSTM network to predict the next day closing price of Shanghai stock index. In their study, they used wavelet decomposition to reconstruct the financial time series for denoising and better learning. In Pang et al. [140], LSTM was used for prediction of the Shanghai A-shares composite index. Namini et al. [136] used LSTM to predict NIKKEI225, IXIC, HIS, GSPC and DJIA index data. In Takahashi et al. [147] and Baek et al. [132], LSTM was also used for the prediction of the S&P500 and KOSPI200 indexes. Baek et al. [132] developed an LSTM-based stock index forecasting model called ModAugNet. The proposed method was able to beat Buy and Hold (B&H) in the long term with an overfitting prevention mechanism. Elliot et al. [134] compared different ML models (linear models), Generalized Linear Models (GMLs) and several LSTM, and RNN models for stock index price prediction. In Hansson et al. [133], LSTM and the autoregressive part of time series index data were used for prediction of the S&P500, Bovespa50, OMX30 indexes.

Also, some studies adapted novel appraches. In Zhang et al. [138], a genetic DNN was used for DJIA index forecasting. Borovykh et al. [127] proposed a new DNN model called Wavenet convolutional net for time series forecasting. Bildirici et al. [148] proposed a Threshold Autoregressive (TAR)-Vector Error Correction model (VEC)-Recurrent Hybrid Elman (RHE) model for forex and stock index of return prediction and compared several models. Parida et al. [124] proposed a method called Locally Recurrent Neuro-fuzzy Information System (LRNFIS) with Firefly Harmony Search Optimization (FHSO) Evolutionary Algorithm (EA) to predict the S&P500 and NIKKEI225 and USD Exchange price data. Psaradellis et al. [149] proposed Heterogeneous Autoregressive Process (HAR) with GA with a SVR (GASVR) model called HAR-GASVR for prediction of the VIX, VXN, Dow Jones Industrial Average Volatility Index (VXD) indexes.

In the literature, some studies used various input data, such as technical indicators, index data, social media news, news from Reuters, and Bloomberg, and statistical features of data (standard deviation, skewness, kurtosis, omega ratio, fund alpha). Table 5 summarizes the index forecasting papers using these aforementioned various data. DNN, RNN, LSTM, and CNN methods were the most commonly used models in index forecasting. In Table 5, different methods/models are also listed within four sub-groups: DNN model; RNN and LSTM models; CNN model; and novel methods.

A DNN was used as the classification model in some papers. In Chen et al. [150], a DNN and some features of the data (Return, Sharpe-ratio (SR), Standard Deviation (STD), Skewness, Kurtosis, Omega ratio, Fund alpha) were used for prediction. In Widegren et al. [126], DNN, RNN, and technical indicators were used for prediction of the FTSE100, OMX30, S&P500 indexes.

In addition, RNN and LSTM models with various other data were also used for prediction of the indexes. Hsieh et al. [137] used RNN and OCHLV of indexes and technical indicators to predict the DJIA, FTSE, Nikkei, and TAIEX indexes. Mourelatos et al. [151] used GASVR, and LSTM for forecasting. Chen et al. [152] used four LSTM models (technical analysis, attention mechanism and market vector embedding) for prediction of the daily return ratio of the HSI300 index. In Li et al. [135], LSTM with wavelet denoising and index data, volume, and technical indicators were used for prediction of the HSI, SSE, SZSE, TAIEX, NIKKEI, and KOSPI indexes. Si et al. [153] used a MODRL+LSTM method to predict Chinese stock-IF-IH-IC contract indexes. Bao et al. [123] used stacked AEs to generate deep features using OCHL of stock prices, technical indicators, and macroeconomic conditions to feed LSTM to predict future stock prices.

Table 5: Index Forecasting Using Various Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[114]	S&P500 Index, 15 stocks in S&P500	2006-2013	News from Reuters and Bloomberg	-	-	CNN	Accuracy, MCC	-
[116]	TWSE index, 4 stocks in TWSE	2001-2017	Technical indicators, Index data, News	15d	-	CNN + LSTM	RMSE, Profit	Keras, Python, TALIB
[123]	CSI300, NIFTY50, HSI, NIKKEI225, S&P500, DJIA	2010-2016	OCHLV, Technical Indicators	-	1d	WT, Stacked autoencoders, LSTM	MAPE, Correlation coefficient, THEIL-U	-
[126]	FTSE100, OMXS 30, SP500, Commodity, Forex	1993-2017	Technical indicators	60d	1d	DNN, RNN	Accuracy, p-value	-
[129]	S&P500, DOW30, NASDAQ100, Commodity, Forex, Bitcoin	2003-2016	Index data, Technical indicators	-	1w, 1m	CNN	Accuracy	Tensorflow
[130]	BSE, S&P500	2004-2012	Index data, technical indicators	5d	1d..1m	PSO, HMRPSO, DE, RCEFLANN	RMSE, MAPE	-
[135]	HSI, SSE, SZSE, TAIEX, NIKKEI, KOSPI	2010-2016	Index data, volume, technical indicators	2d..512d	1d	LSTM with wavelet denoising	Accuracy, MAPE	-
[137]	DJIA, FTSE, NIKKEI, TAIEX	1997-2008	OCHLV, Technical indicators	26d	1d	RNN	RMSE, MAE, MAPE, THEIL-U	C
[150]	Hedge fund monthly return data	1996-2015	Return, SR, STD, Skewness, Kurtosis, Omega ratio, Fund alpha	12m	3m, 6m, 12m	DNN	Sharpe ratio, Annual return, Cum. return	-
[151]	Stock of National Bank of Greece (ETE).	2009-2014	FTSE100, DJIA, GDAX, NIKKEI225, EUR/USD, Gold	1d, 2d, 5d, 10d	1d	GASVR, LSTM	Return, volatility, SR, Accuracy	Tensorflow
[152]	Daily return ratio of HS300 index	2004-2018	OCHLV, Technical indicators	-	-	Market Vector + Tech. ind. + LSTM + Attention	MSE, MAE	Python, Tensorflow
[153]	Chinese stock-IF-IH-IC contract	2016-2017	Decisions for index change	240min	1min	MODRL+LSTM	Profit and loss, SR	-
[154]	HS300	2015-2017	Social media news, Index data	1d	1d	RNN-Boost with LDA	Accuracy, MAE, MAPE, RMSE	Python, Scikit-learn

Besides, different CNN implementations with various data (technical indicators, news, and index data) have been used in the literature. In Dingli et al. [129], CNN, and index data, and technical indicators were used for the S&P500, DOW30, NASDAQ100 indexes and Commodity, Forex, and Bitcoin prices. In Ding et al. [114], a CNN model with news from Reuters and Bloomberg were used for prediction of the S&P500 index and 15 stocks' prices in S&P500. In Lee et al. [116], CNN + LSTM and technical indicators, index data, and news were used for forecasting of the Taiwan Stock Exchange (TWSE) index and 4 stocks' prices in TWSE.

In addition, some novel methods have been proposed for index forecasting. Rout et al. [130] used RNN models, Recurrent Computationally Efficient Functional Link Neural Network (RCEFLANN), and Functional Link Neural network (FLANN), with their weights optimized using various EAs like Particle Swarm Optimization (PSO), and Modified Version of PSO (HMRPSO), for time series forecasting. Chen et al. [154] used social media news to predict index price and index direction with RNN-Boost with Latent Dirichlet Allocation (LDA) features.

4.3. Commodity Price Forecasting

A number of studies particularly focused on the price prediction of any given commodity, such as gold, silver, oil, and copper. With the increasing number of commodities available for public trading through online stock exchanges, interest in this topic will likely grow in the following years.

In the literature, there have been different methods used for commodity price forecasting. DNN, RNN, FDDR, and CNN are the most used models to predict commodity prices. Table 6 lists the details of the commodity price forecasting studies with DL.

In Dingli et al. [129], the authors used a CNN for predicting the next week's and next month's price directional movement. Meanwhile, RNN and LSTM models were used in some commodity forecasting studies. In Dixon et al. [155], a DNN was used for commodity forecasting. In Widegren et al. [126], forex, and index datasets were used. DNN and RNN were used to predict the prices of time series data. Technical indicators were used as the feature set consisting of Relative Strength Index (RSI), Williams Percent Range (William%R), Commodity Channel Index (CCI), Percentage Price Oscillator (PPOSC), momentum, and Exponential Moving Average (EMA). In Lasheras et al. [156], the authors used an Elman RNN to predict COMEX copper spot price (through New York Mercantile Exchange (NYMEX)) from daily closing prices.

Hybrid and novel models have been adapted in some studies. In Zhao et al. [157], FNN and Stacked Denoising Autoencoders (SDAE) deep models were compared against Support Vector Regressor (SVR), Random Walk (RW), and Markov Regime Switching (MRS) models for WTI oil price forecasting. As performance criteria, accuracy, Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) were used. In Chen et al. [158], the authors aimed to predict WTI crude oil prices using several models, including combinations of DBN, LSTM, Autoregressive Moving Average (ARMA), and RW. MSE was used as the performance criteria. In Deng et al. [141], the authors used FDDR for stock price prediction and trading signal generation. They combined DNN and RL. Profit, return, SR, and profit-loss curves were used as the performance criteria.

Table 6: Commodity Price Forecasting

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[129]	S&P500, DOW30, NASDAQ100, Commodity, Forex, Bitcoin	2003-2016	Price data, Technical indicators	-	1w, 1m	CNN	Accuracy	Tensorflow
[155]	Commodity, FX future, ETF	1991-2014	Price Data	100*5min	5min	DNN	SR, capability ratio, return	C++, Python
[126]	FTSE100, OMX30, S&P500, Commodity, Forex	1993-2017	Technical indicators	60d	1d	DNN, RNN	Accuracy, p-value	-
[156]	Copper prices from NYMEX	2002-2014	Price data	-	-	Elman RNN	RMSE	R
[157]	WTI crude oil price	1986-2016	Price data	1m	1m	SDAE, Bootstrap aggregation	Accuracy, MAPE, RMSE	Matlab
[158]	WTI Crude Oil Prices	2007-2017	Price data	-	-	ARMA + DBN, RW + LSTM	MSE	Python, Keras, Tensorflow
[141]	300 stocks from SZSE, Commodity	2014-2015	Price data	-	-	FDDR, DNN + RL	Profit, return, SR, profit-loss curves	Keras

4.4. Volatility Forecasting

Volatility is directly related to price variations in a given time period and is mostly used for risk assesment and asset pricing. Some researchers implemented models for accurately forecasting the underlying volatility of any given asset.

In the literature, there have been different methods used for volatility forecasting, including LSTM, RNN, CNN, MM, and Generalised Auto-Regressive Conditional Heteroscedasticity (GARCH) models. Table 7 summarizes the studies that focused on volatility forecasting. In Table 7, different methods/models are also represented as three sub-groups: CNN; RNN and LSTM models; and hybrid and novel models.

A CNN model was used in one volatility forecasting study based on HFT data [159]. Meanwhile, RNN and LSTM models were used in some studies. In Tino et al. [160], the authors used financial time series data to predict volatility changes with Markov Models and Elman RNN for profitable straddle options trading. Xiong et al. [161] used price data and different types of Google Domestic trends with LSTM. Zhou et al. [162] used CSI300, and 28 words from the daily search volume based on Baidu as the dataset with LSTM to predict index volatility. Kim et al. [163] developed several LSTM models integrated with GARCH for volatility prediction.

Hybrid and novel approaches have also been adapted in some studies. In Nikolaev et al. [164], an RMDN with GARCH (RMDN-GARCH) model was proposed. In addition, several models, including traditional forecasting models and DL models, were compared for volatility estimation. Psaradellis et al. [149] proposed a novel method called HAR with GASVR (HAR-GASVR) for volatility index forecasting.

Table 7: Volatility Forecasting

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[159]	London Stock Exchange	2007-2008	Limit order book state, trades, buy/sell orders, order deletions	-	-	CNN	Accuracy, kappa	Caffe
[160]	DAX, FTSE100, call/put options	1991-1998	Price data	*	*	MM, RNN	Ewa-measure, iv, daily profits' mean and std	-
[161]	S&P500	2004-2015	Price data, 25 Google Domestic trend dimensions	-	1d	LSTM	MAPE, RMSE	-
[162]	CSI 300, 28 words of the daily search volume based on Baidu	2006-2017	Price data and text	5d	5d	LSTM	MSE, MAPE	Python, Keras
[163]	KOSPI200, Korea Treasury Bond interest rate, AA-grade corporate bond interest rate, gold, crude oil	2001-2011	Price data	22d	1d	LSTM + GARCH	MAE, MSE, HMAE, HMSE	-
[164]	DEM/GBP exchange rate	-	Returns	-	-	RMDN-GARCH	NMSE, NMAE, HR, WHR	-
[149]	VIX, VXN, VXD	2002-2014	First five autoregressive lags	5d	1d, 22d	HAR-GASVR	MAE, RMSE	-

4.5. Bond Price Forecasting

Some financial experts follow the changes in bond prices to analyze the state of the economy, claiming bond prices represent the health of the economy better than the stock market [165]. Historically, long term rates are higher than short term rates under normal economic expansion, whereas immediately before recessions, short term rates pass long term rates, i.e., an inverted yield curve. Hence, accurate bond price prediction is very useful. However, DL implementations for bond price prediction are very scarce. In Bianchi et al. [166], excess bond return was predicted using several ML models, including RF, AE, and PCA networks and a 2-3-4-layer DFNN. 4-layer NN outperformed the other models.

4.6. Forex Price Forecasting

Foreign exchange markets have the highest volumes among all existing financial markets in the world. They are open 24/7, and trillions of dollars worth of foreign exhange transactions happen in a single day. According to the Bank for International Settlements, foreign-exchange trading has a volume of more than 5 trillion USD a day [167]. In addition, there are a large number of online forex trading platforms that provide leveraged transaction opportunities to their subscribers. As a result, there is huge interest in profitable trading strategies by traders. Hence, there are a number of forex forecasting and trading studies based on DL models. Because most of the global financial transactions are based on the US Dollar, almost all forex prediction research papers include USD in their analyses. However, depending on regional differences and intended research focus, various models have been developed accordingly.

In the literature, different methods have been used for forex price forecasting, including RNN, LSTM, CNN, DBN, DNN, AE, and MLP methods. Table 8 provides details about these implementations. In Table 8, different methods/models are listed as four subgroups: Continuous-valued Deep Belief Networks (CDBN), DBN, DBN+RBM, and AE models; DNN, RNN, Psi-Sigma Network (PSN), and LSTM models; CNN models; and hybrid models.

CDBN, DBN, DBN+RBM, and AE models have been used in some studies. In [168], Fuzzy information granulation integrated with CDBN was applied for predicting EUR/USD and GBU/USD exchange rates. They extended a DBN with a Continuous Restricted Boltzman machine (CRBM) to improve performance. In Chao et al. [169], weekly GBP/USD and INR/USD prices were predicted, whereas in Zheng et al. [170], CNY/USD and INR/USD were the main focus. In both cases, DBN was compared with FFNN. Similarly, Shen et al. [171] implemented several different DBN networks to predict weekly GBP/USD, BRL/USD and INR/USD exchange rate returns. Shen et al. [172] combined Stacked AE and SVR for predicting 28 normalized currency pairs using the time series data of USD, GBP, EUR, JPY, AUD, CAD, and CHF.

DNN, RNN, PSN, and LSTM models were preferred in some studies. In Dixon et al. [155], multiple DMLP models were developed for predicting AD and BP futures using 5-minute data over in a 130 day period. Sermpinis et al. [173] used MLP, RNN, GP, and other ML techniques along with traditional regression methods for also predicting EUR/USD time series. They also integrated Kalman filter, LASSO operator, and other models to further improve the results [174]. They further extended their analyses by including PSN and providing comparisons along with traditional forecasters ARIMA, RW, and STAR [175]. To improve performance, they also integrated hybrid time-varying volatility leverage. Sun et al. [176] implemented RMB exchange rate forecasting against JPY, HKB, EUR and USD by comparing RW, RNN, and FFNN performances. Maknickiene et al. [177] predicted various forex time series and created portfolios consisting of these investments. Each network used LSTM (RNN EVOLINO), and different risk appetites for users have been tested. Maknickiene et al. [178] also used EVOLINO RNN + orthogonal input data for predicting USD/JPY and XAU/USD prices over different periods.

Different CNN models were used in some studies. In Persio et al. [179], EUR/USD was once again forecasted using multiple DL models, including MLP, CNN, RNN, and Wavelet+CNN. Korczak et al. [180] implemented forex trading (GBP/PLN) using several different input parameters in a multi-agent-based trading environment. One of the agents used AE+CNN as the prediction model and outperformed all other models.

Hybrid models have also been adapted in some of the researches. Bildirici et al. [148] developed several (TAR-VEC-RHE) models for predicting monthly returns for TRY/USD and compared model performances. Nikolaev et al. [164] compared several models, including traditional forecasting models and DL models, for DEM/GBP prediction. Parida et al. [124] predicted AUD, CHF, MAX, and BRL against USD currency time series data using LRNFIS and compared it with different models. Meanwhile, instead of using LMS-based error minimization during learning, they used FHSO.

Table 8: Forex Price Forecasting

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[168]	EUR/USD, GBP/USD	2009-2012	Price data	*	1d	CDBN-FG	Profit	-
[169]	GBP/USD, INR/USD	1976-2003	Price data	10w	1w	DBN	RMSE, MAE, MAPE, DA, PCC	-
[170]	CNY/USD,INR/USD	1997-2016	Price data	-	1w	DBN	MAPE, R-squared	-
[171]	GBP/USD, BRL/USD, INR/USD	1976-2003	Price data	10w	1w	DBN + RBM	RMSE, MAE, MAPE, accuracy, PCC	-
[172]	Combination of USD, GBP, EUR, JPY, AUD, CAD, CHF	2009-2016	Price data	-	-	Stacked AE + SVR	MAE, MSE, RMSE	Matlab
[155]	Commodity, FX future, ETF	1991-2014	Price Data	100*5min	5min	DNN	SR, capability ratio, return	C++, Python
[126]	FTSE100, OMX30, S&P500, Commodity, Forex	1993-2017	Technical indicators	60d	1d	DNN, RNN	Accuracy, p-value	-
[173]	EUR/USD	2001-2010	Close data	11d	1d	RNN and more	MAE, MAPE, RMSE, THEIL-U	-
[174]	EUR/USD	2002-2010	Price data	13d	1d	RNN, MLP, PSN	MAE, MAPE, RMSE, THEIL-U	-
[175]	EUR/USD, EUR/GBP, EUR/JPY, EUR/CHF	1999-2012	Price data	12d	1d	RNN, MLP, PSN	MAE, MAPE, RMSE, THEIL-U	-
[176]	RMB against USD, EUR, JPY, HKD	2006-2008	Price data	10d	1d	RNN, ANN	RMSE, MAE, MSE	-
[177]	EUR/USD, EUR/JPY, USD/JPY, EUR/CHF, XAU/USD, XAG/USD, QM, QG	2011-2012	Price data	-	-	Evolino RNN	Correlation between predicted, real values	-
[178]	USD/JPY	2009-2010	Price data, Gold	-	5d	EVOLINO RNN + orthogonal input data	RMSE	-
[179]	S&P500, EUR/USD	1950-2016	Price data	30d, 30d*min	1d, 1min	Wavelet+CNN	Accuracy, log-loss	Keras
[180]	USD/GBP, S&P500, FTSE100, oil, gold	2016	Price data	-	5min	AE + CNN	SR, % volatility, avg return/trans, rate of return	H2O
[148]	ISE100, TRY/USD	1987-2008	Price data	-	2d, 4d, 8d, 12d, 18d	TAR-VEC-MLP, TAR-VEC-RBF, TAR-VEC-RHE	RMSE	-
[164]	DEM/GBP exchange rate	-	Returns	-	-	RMDN-GARCH	NMSE, NMAE, HR, WHR	-
[124]	S&P500, NIKKEI225, USD Exchanges	2011-2015	Price data	-	1d, 5d, 7d, 10d	LRNFIS with FHSO	RMSE, MAPE, MAE	-

4.7. Cryptocurrency Price Forecasting

Since cryptocurrencies have become a hot topic in the finance industry in recent years, many studies and implementations have been conducted. Most cryptocurrency studies have focused on price forecasting.

The rise of Bitcoin from 1000 USD in January 2017 to 20,000 USD in January 2018 has attracted much attention, not only from the financial industry, but also from the general public. Recently, papers have been published on price prediction and trading strategy development for Bitcoin and other cryptocurrencies. Given the attention that the underlying technology has attracted, there is a strong chance that new studies will appear in the near future.

In the literature, DNN, LSTM, GRU, RNN, and classical methods (ARMA, ARIMA, Autoregressive Conditional Heteroscedasticity (ARCH), GARCH, etc.) have been used for cryptocurrency price forecasting. Table 9 summarizes the studies that utilized these methods. Lopes [181] combined the opinion market and price prediction for cryptocurrency trading. Text mining combined with 2 models, CNN and LSTM, were used to extract opinion. Bitcoin, Litecoin, and StockTwits were used as the dataset. OCHLV of prices, technical indicators, and sentiment analysis were used as the feature set. McNally et al. [182] compared Bayesian optimized RNN, LSTM, and ARIMA to predict Bitcoin price direction. Sensitivity, specificity, precision, accuracy, and RMSE were used as the performance metrics.

Table 9: Cryptocurrency Price Prediction

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[181]	Bitcoin, Litecoin, StockTwits	2015-2018	OCHLV, technical indicators, sentiment analysis	-	30min, 4h, 1d	CNN, LSTM, State Frequency Model	MSE	Keras, Tensorflow
[182]	Bitcoin	2013-2016	Price data	100d	30d	Bayesian optimized RNN, LSTM	Sensitivity, specificity, precision, accuracy, RMSE	Keras, Python, Hyperas

4.8. Trend Forecasting

Although trend forecasting and price forecasting share the same input characteristics, some researchers prefer to predict the price direction of an asset instead of its actual price. This alters the nature of the problem from regression to classification, and the corresponding performance metrics also change. However, it is worth noting that these two approaches are still fundamentally the same; the difference is in the interpretation of the output.

In the literature, there are different methods for trend forecasting. In this survey, we grouped the articles according to their feature sets, such as studies using only raw time series data (only price data, OCHLV); studies using technical indicators, price data, and fundamental data at the same time; studies using text mining techniques; and studies using various other data. Table 10 summarizes the trend forecasting studies using only raw time series data.

Table 10: Trend Forecasting Using Only Raw Time Series Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[183]	S&P500 stock indexes	1963-2016	Price data	30d	1d	NN	Accuracy, precision, recall, F1-score, AUROC	R, H2o, Python, Tensorflow
[184]	SPY ETF, 10 stocks from S&P500	2014-2016	Price data	60min	30min	FNN	Cumulative gain	MatConvNet, Matlab
[142]	Shanghai composite index and SZSE	1990-2016	OCHLV	20d	1d	Ensembles of ANN	Accuracy	-
[185]	10 stocks from S&P500	-	Price data			TDNN, RNN, PNN	Missed opportunities, false alarms ratio	-
[186]	GOOGL stock daily price data	2012-2016	Time window of 30 days of OCHLV	22d, 50d, 70d	*	LSTM, GRU, RNN	Accuracy, Logloss	Python, Keras
[133]	S&P500, Bovespa50, OMX30	2009-2017	Autoregressive part of the price data	30d	1..15d	LSTM	MSE, Accuracy	Tensorflow, Keras, R
[187]	HSI, DAX, S&P500	1991-2017	Price data	-	1d	GRU, GRU-SVM	Daily return %	Python, Tensorflow
[188]	Taiwan Stock Index Futures	2001-2015	OCHLV	240d	1..2d	CNN with GAF, MAM, Candlestick	Accuracy	Matlab
[189]	ETF and Dow30	1997-2007	Price data			CNN with feature imaging	Annualized return	Keras, Tensorflow
[190]	SSEC, NASDAQ, S&P500	2007-2016	Price data	20min	7min	EMD2FNN	MAE, RMSE, MAPE	-
[191]	23 cap stocks from the OMX30 index in Nasdaq Stockholm	2000-2017	Price data and returns	30d	*	DBN	MAE	Python, Theano

Different methods and models have been used for trend forecasting. In Table 10, these are divided into three sub-groups: ANN, DNN, and FFNN models; LSTM, RNN, and Probabilistic NN models; and novel methods. ANN, DNN, DFNN, and FFNN methods were used in some studies. In Das et al. [183], NN with price data was used for trend prediction of the S&P500 stock indexes. Navon et al. [184] combined deep FNN with a selective trading strategy unit to predict the next price. Yang et al. [142] created an ensemble network of several Backpropagation and ADAM models for trend prediction.

In the literature, LSTM, RNN, and Probabilistic Neural Network (PNN) methods with raw time series data have also been used for trend forecasting. Saad et al. [185] compared Timedelay Neural Network (TDNN), RNN, and PNN for trend detection using 10 stocks from S&P500. Persio et al. [186] compared 3 different RNN models (basic RNN, LSTM, and GRU) to predict the movement of Google stock prices. Hansson et al. [133] used LSTM (and other classical forecasting techniques) to predict the trend of stocks prices. In Shen et al. [187], GRU and GRU-SVM models were used for the trends of the HSI, The Deutscher Aktienindex (DAX), and S&P500 indexes.

There are also novel methods that use only raw time series price/index data in the literature. Chen et al. [188] proposed a method that used a CNN with Gramian Angular Field (GAF), Moving Average Mapping (MAM), and Candlestick with converted image data. In Sezer et al. [189], a novel method of CNN with feature imaging was proposed for prediction of the buy/sell/hold positions of the Exchange-Traded Funds (ETFs)' prices and Dow30 stocks' prices. Zhou et al. [190] proposed a method that uses Empirical Mode Decomposition and Factorization Machine based Neural Network (EMD2FNN) models to forecast the directions of stock closing prices accurately. In Ausmees et al. [191], DBN with price data was used for trend prediction of 23 large cap stocks from the OMX30 index.

Table 11: Trend Forecasting Using Technical Indicators & Price Data & Fundamental Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[192]	KSE100 index	-	Price data, several fundamental data	-	-	ANN, SLP, MLP, RBF, DBN, SVM	Accuracy	-
[193]	Stocks in Dow30	1997-2017	RSI (Technical Indicators)	200d	1d	DMLP with genetic algorithm	Annualized return	Spark MLlib, Java
[194]	SSE Composite Index, FTSE100, PingAnBank	1999-2016	Technical indicators, OCHLV price	24d	1d	RBM	Accuracy	-
[195]	Dow30 stocks	2012-2016	Price data, several technical indicators	40d	-	LSTM	Accuracy	Python, Keras, Tensorflow, TALIB
[196]	Stock price from IBOVESPA index	2008-2015	Technical indicators, OCHLV of price	-	15min	LSTM	Accuracy, Precision, Recall, F1-score, % return, Maximum drawdown	Keras
[197]	20 stocks from NASDAQ and NYSE	2010-2017	Price data, technical indicators	5d	1d	LSTM, GRU, SVM, XGBoost	Accuracy	Keras, Tensorflow, Python
[198]	17 ETF	2000-2016	Price data, technical indicators	28d	1d	CNN	Accuracy, MSE, Profit, AUROC	Keras, Tensorflow
[199]	Stocks in Dow30 and 9 Top Volume ETF	1997-2017	Price data, technical indicators	20d	1d	CNN with feature imaging	Recall, precision, F1-score, annualized return	Python, Keras, Tensorflow, Java
[200]	Borsa Istanbul 100 Stocks	2011-2015	75 technical indicators, OCHLV of price	-	1h	CNN	Accuracy	Keras

Some studies have used technical indicators, price data, and fundamental data at the same time. Table 11 summarizes the trend forecasting papers that used technical indicators, price data, and fundamental data. In addition, these studies are clustered into three subgroups: ANN, MLP, DBN, and RBM models; LSTM and GRU models; and novel methods. ANN, MLP, DBN, and RBM methods were used with technical indicators, price data, and fundamental data in some studies. In Raza et al. [192], several classical and ML models and DBN were compared for trend forecasting. In Sezer et al. [193], technical analysis indicator's (RSI) buy and sell limits were optimized with GA, which was used for buy-sell signals. After optimization, DMLP was also used for function approximation. Liang et al. [194] used technical analysis parameters, OCHLV of prices, and RBM for stock trend prediction.

LSTM and GRU methods with technical indicators, price data, and fundamental data were also used in some papers. In Troiano et al. [195], the crossover and Moving Average

Convergence and Divergence (MACD) signals were used to predict the trend of Dow 30 stock prices. Nelson et al. [196] used LSTM for stock price movement estimation. Song et al. [197] used stock prices, technical analysis features, and four different ML models (LSTM, GRU, SVM and eXtreme Gradient Boosting (XGBoost)) to predict the trend of stock prices.

In addition, novel methods using CNN with the price data and technical indicators have been proposed. Gudelek et al. [198] converted the time series of price data to 2-dimensional images using technical analysis and classified them with a deep CNN. Similarly, Sezer et al. [199] also proposed a novel technique that converted financial time series data consisting of technical analysis indicator outputs to 2-dimensional images and classified these images using a CNN to determine the trading signals. Gunduz et al. [200] proposed a method using a CNN with correlated features combined to predict the trend of stock prices.

Besides, there have also been studies using text mining techniques. Table 12 summarizes the trend forecasting papers using text mining techniques. Different methods/models are represented by four sub-groups: DNN, DMLP, and CNN with text mining models; GRU model; LSTM, CNN, and LSTM+CNN models; and novel methods. In the first group of studies, DNN, DMLP, and CNN with text mining were used for trend forecasting. In Huang et al. [201], the authors used different models, including Hidden Markov Model (HMM), DMLP, and CNN using Twitter moods, to predict the next day's movement. Peng et al. [202] used the combination of text mining and word embeddings to extract information from financial news and a DNN model for prediction of stock trends.

Moreover, GRU methods with text mining techniques have also been used for trend forecasting. Huynh et al. [203] used financial news from Reuters and Bloomberg, stock price data, and a Bidirectional Gated Recurrent Unit (Bi-GRU) model to predict future stock movements. Dang et al. [204] used Stock2Vec and Two-stream GRU (TGRU) models to generate input data from financial news and stock prices. Then, they used the sign difference between the previous close and next open for the classification of stock prices. The results were better than those of state-of-the-art models.

LSTM, CNN, and LSTM+CNN models were also used for trend forecasting. Verma et al. [205] combined news data with financial data to classify stock price movement and assessed them with certain factors. They used an LSTM model as the NN architecture. Pinheiro et al. [206] proposed a novel method that used a character-based neural language model using financial news and LSTM for trend prediction. In Prosky et al. [207], sentiment/mood prediction and price prediction based on sentiment, price prediction with text mining, and DL models (LSTM, NN, CNN) were used for trend forecasting. Liu et al. [208] proposed a method that used two separate LSTM networks to construct an ensemble network. One of the LSTM models was used for word embeddings with word2Vec to create a matrix information input to the CNN. The other was used for price prediction using technical analysis features and stock prices.

In the literature, there are also novel methods to predict the trend of time series data. Yoshihara et al. [209] proposed a novel method that uses a combination of RBM, DBN, and word embedding to create word vectors for an RNN-RBM-DBN network to predict the trend of stock prices. Shi et al. [210] proposed a novel method called DeepClue that visually interpretted text-based DL models in predicting stock price movements. In their proposed method, financial news, charts, and social media tweets were used together to predict stock price movement. Zhang et al. [211] proposed a method that performed information fusion from several news and social media sources to predict the trend of stocks. Hu et al. [212] proposed a novel method that used text mining techniques and Hybrid Attention Networks based on financial news for trend forecasting of stocks. Wang et al. [213] combined technical analysis and sentiment analysis of social media (related financial topics) and created a Deep Random Subspace Ensembles (DRSE) method for classification. Matsubara et al. [214] proposed a method that used a Deep Neural Generative Model (DGM) with news articles using a Paragraph Vector algorithm to create the input vector for prediction of stock trends. Li et al. [215] implemented intraday stock price direction classification using financial news and stock prices.

Table 12: Trend Forecasting Using Text Mining Techniques

Art.	**Data Set**	**Period**	**Feature Set**	**Lag**	**Horizon**	**Method**	**Performance Criteria**	**Env.**
[201]	S&P500, NYSE Composite, DJIA, NASDAQ Composite	2009-2011	Twitter moods, index data	7d	1d	DNN, CNN	Error rate	Keras, Theano
[202]	News from Reuters and Bloomberg, Historical stock security data	2006-2013	News, price data	5d	1d	DNN	Accuracy	-
[203]	News from Reuters, Bloomberg	2006-2013	Financial news, price data	-	1d, 2d, 5d, 7d	Bi-GRU	Accuracy	Python, Keras
[204]	News about Apple, Airbus, Amazon from Reuters, Bloomberg, S&P500 stock prices	2006-2013	Price data, news, technical indicators	-	-	Two-stream GRU, stock2vec	Accuracy, precision, AUROC	Keras, Python
[205]	NIFTY50 Index, NIFTY Bank/Auto/IT/Energy Index, News	2013-2017	Index data, news	1d, 2d, 5d	1d	LSTM	MCC, Accuracy	-
[206]	News from Reuters, Bloomberg, stock price/index data from S&P500	2006-2013	News and sentences	-	1h, 1d	LSTM	Accuracy	-
[207]	30 DJIA stocks, S&P500, DJI, news from Reuters	2002-2016	Price data and features from news articles	1m	1d	LSTM, NN, CNN and word2vec	Accuracy	VADER
[208]	APPL from S&P500 and news from Reuters	2011-2017	News, OCHLV, Technical indicators	-	1d	CNN + LSTM, CNN+SVM	Accuracy, F1-score	Tensorflow
[209]	News, Nikkei Stock Average and 10-Nikkei companies	1999-2008	News, MACD	-	1d	RNN, RBM+DBN	Accuracy, P-value	-
[210]	News from Reuters and Bloomberg for S&P500 stocks	2006-2015	Financial news, price data	1d	1d	DeepClue	Accuracy	Dynet software
[211]	Price data, index data, news, social media data	2015	Price data, news from articles and social media	1d	1d	Coupled matrix and tensor	Accuracy, MCC	Jieba
[212]	News and Chinese stock data	2014-2017	Selected words in a news	10d	1d	HAN	Accuracy, Annual return	-
[213]	Sina Weibo, Stock market records	2012-2015	Technical indicators, sentences	-	-	DRSE	F1-score, precision, recall, accuracy, AUROC	Python

Table 12: Trend Forecasting Using Text Mining Techniques

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[214]	Nikkei225, S&P500, news from Reuters and Bloomberg	2001-2013	Price data and news	1d	1d	DGM	Accuracy, MCC, %profit	-
[215]	News, stock prices from Hong Kong Stock Exchange	2001	Price data and TF-IDF from news	60min	(1..6)*5min	ELM, DLR, PCA, BELM, KELM, NN	Accuracy	Matlab

Moreover, studies have also used different data variations. Table 13 summarizes the trend forecasting papers using these various data clustered into two sub-groups: LSTM, RNN, and GRU models and CNN models.

LSTM, RNN, and GRU methods with various data representations have been used in some trend forecasting papers. Tsantekidis et al. [216] used limit order book time series data and an LSTM method for trend prediction. Sirignano et al. [217] proposed a novel method that used limit order book flow and history information to determine stock movements using LSTM. The results of the proposed method were remarkably stationary. Chen et al. [154] used social media news, LDA features, and an RNN model to predict the trend of index prices. Buczkowski et al. [218] proposed a novel method that used expert recommendations (Buy, Hold, or Sell), emsemble of GRU, and LSTM to predict the trend of stock prices.

CNN models with different data representations were also used for trend prediction. Tsantekidis et al. [219] used the last 100 entries from the limit order book to create images for stock price prediction using a CNN. Using the limit order book data to create a 2D matrix-like format with a CNN for predicting directional movement was innovative. In Doering et al. [159], HFT microstructure forecasting was implemented with a CNN.

Table 13: Trend Forecasting Using Various Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[216]	Nasdaq Nordic (Kesko Oyj, Outokumpu Oyj, Sampo, Rautaruukki, Wartsila Oyj)	2010	Price and volume data in LOB	100s	10s, 20s, 50s	LSTM	Precision, Recall, F1-score, Cohen's k	-
[217]	High-frequency record of all orders	2014-2017	Price data, record of all orders, transactions	2h	-	LSTM	Accuracy	-
[154]	Chinese, The Shanghai-Shenzhen 300 Stock Index (HS300	2015-2017	Social media news (Sina Weibo), price data	1d	1d	RNN-Boost with LDA	Accuracy, MAE, MAPE, RMSE	Python, Scikit learn
[218]	ISMIS 2017 Data Mining Competition dataset	-	Expert identifier, class predicted by expert	-	-	LSTM + GRU + FCNN	Accuracy	-
[219]	Nasdaq Nordic (Kesko Oyj, Outokumpu Oyj, Sampo, Rautaruukki, Wartsila Oyj)	2010	Price, Volume data, 10 orders of the LOB	-	-	CNN	Precision, Recall, F1-score, Cohen's k	Theano, Scikit learn, Python

Table 13: Trend Forecasting Using Various Data

Art.	Data Set	Period	Feature Set	Lag	Horizon	Method	Performance Criteria	Env.
[159]	London Stock Exchange	2007-2008	Limit order book state, trades, buy/sell orders, order deletions	-	-	CNN	Accuracy, kappa	Caffe

5. Current Snaphot of The Field

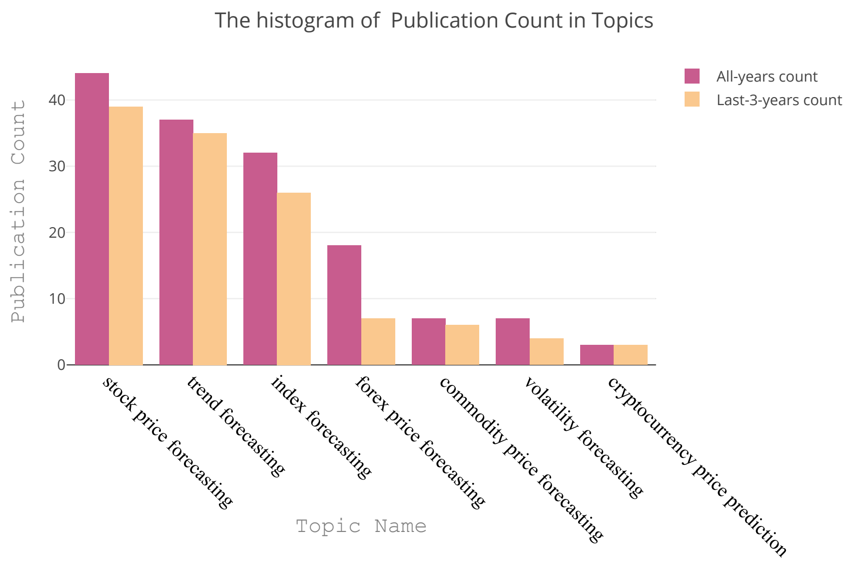

Figure 5: The histogram of Publication Count in Topics

After reviewing all research papers specifically targeted at financial time series forecasting implementations using DL models, we are now ready to provide some overall statistics about the current state of the field. The number of papers included in our survey was 140. We categorized the papers according to their forecasted asset type. We also analyzed the studies based on their DL model choices, frameworks for the development environment, data sets, comparable benchmarks, and some other differentiating criteria such as feature sets and numbers of citations, which could not be included in this paper due to space constraints. We will now summarize our notable observations to provide interested research with important highlights within the field.

Figure 5 presents the various asset types that researchers developed their corresponding forecasting models for. As expected, stock market-related prediction studies dominate the field. Stock price forecasting, trend forecasting, and index forecasting were the top three picks for financial time series forecasting research. So far, 46 papers have been published for stock price forecasting, 38 for trend forecasting and 33 for index forecasting. These studies

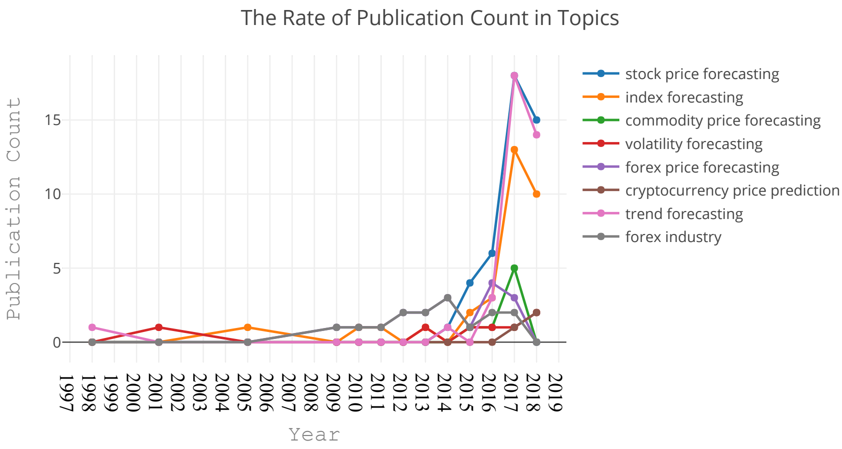

Figure 6: The rate of Publication Count in Topics

constitute more than 70% of all studies, indicating high interest. Besides the above, there were 19 papers on forex prediction and 7 on volatility forecasting. Meanwhile cryptocurrency forecasting has started attracting researchers; however, only 3 papers on this topic have been published yet, but this number is expected to increase in the coming years [220]. Figure 6 highlights the rate of publication counts for various implementation areas throughout the years. Meanwhile, Figure 7 provides more details about the choice of DL models over various implementation areas.

Figure 8 illustrates the increasing appetite of researchers to develop DL models for financial time series implementations. Meanwhile, as Figure 9 indicates, most studies were published in journals (57 of them) and conferences (49 papers), but a considerable number of arXiv papers (11) and graduate theses (6) also exist.

One of the most important issues for a researcher is where they can publish their findings. During our review, we also carefully investigated where each paper was published. We tabulated our results for the top journals for financial time series forecasting in Fig 10. According to these results, the journals with the most published papers include Expert Systems with Applications, Neurocomputing, Applied Soft Computing, The Journal of Supercomputing, Decision Support Systems, Knowledge-based Systems, European Journal of Operational Research, and IEEE Access. The interested researchers should also consider the trends over the last 3 years, as tendencies can vary depending on the particular implementation areas.

Carefully analyzing Figure 11 clearly validates the dominance of RNN-based models (65 papers) among all others for DL model choices, followed by DMLP (23 papers) and CNN (20 papers). The inner-circle represents all years considered, while the outer circle provides only the studies within the last 3 years. We should note that the RNN is a general model

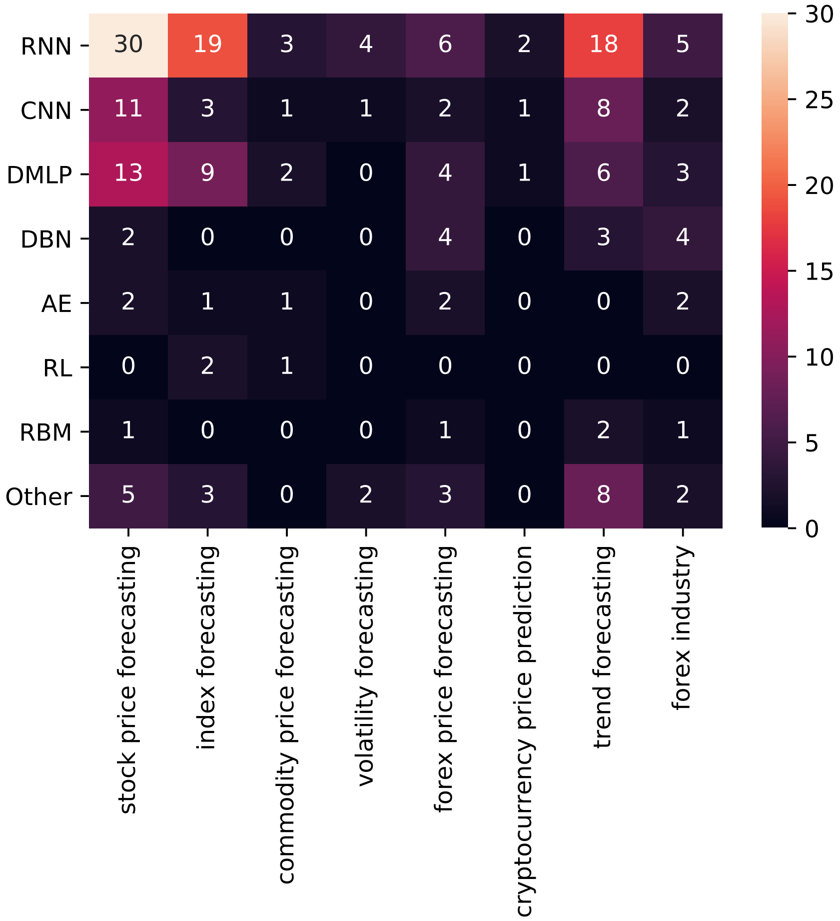

Figure 7: Topic-Model Heatmap

with several versions, including LSTM and GRU. For RNN, researchers mostly prefer LSTM due to its relatively simple model development phase; however, other types of RNN are also common. Figure 12 provides a snapshot of the RNN model distribution. As mentioned above, LSTM had the highest interest among all with 58 papers, while Vanilla RNN and GRU had 27 and 10 papers, respectively. Hence, it is clear that LSTM is the most popular DL model for financial time series forecasting and regression studies.

Meanwhile, DMLP and CNN were generally preferred for classification problems. Because time series data generally consist of temporal components, some data preprocessing might be required before actual classification can occur. Hence, many of these implementations utilize feature extraction, selection techniques, and possible dimensionality reduction methods. Many researchers mainly use DMLP due to the fact that its shallow MLP version has been used extensively before and has a proven successful track record for many different financial applications, including financial time series forecasting. Consistent with our observations, DMLP was also mostly preferred in the stock, index, and particular trend forecasting because it is by definition, a classification problem with two (uptrend or downtrend)

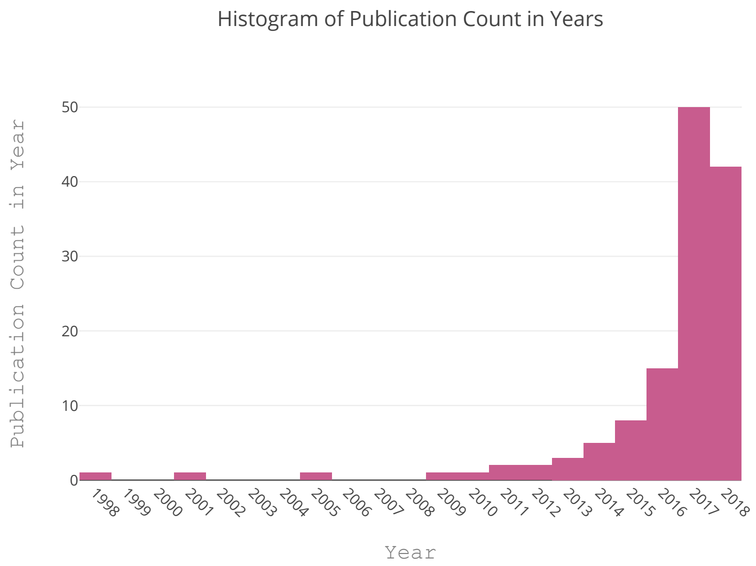

Figure 8: The histogram of Publication Count in Years

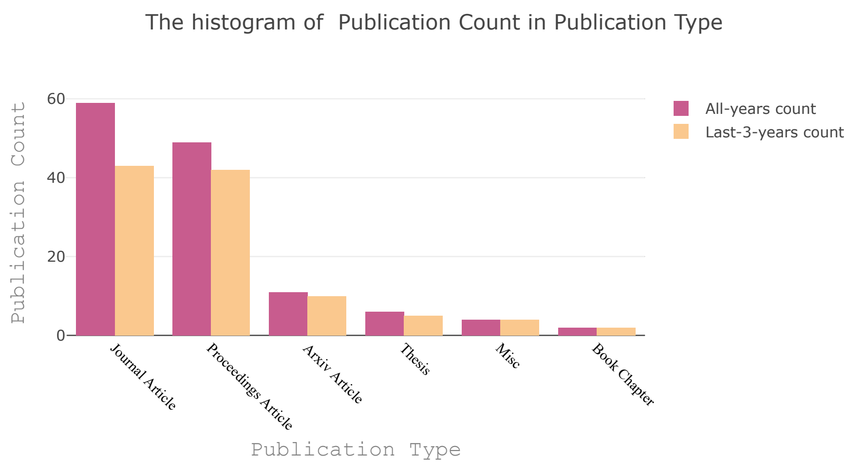

Figure 9: The histogram of Publication Count in Publication Types

and three (uptrend, stationary, or downtrend) class instances.

In addition to DMLP, CNN is also a popular choice for classification-type financial time series forecasting implementations. Most of these studies appeared within the last 3 years. As mentioned before, to convert time-varying sequential data into a more stationary classifiable form, some preprocessing might be necessary. Even though some 1-D representations

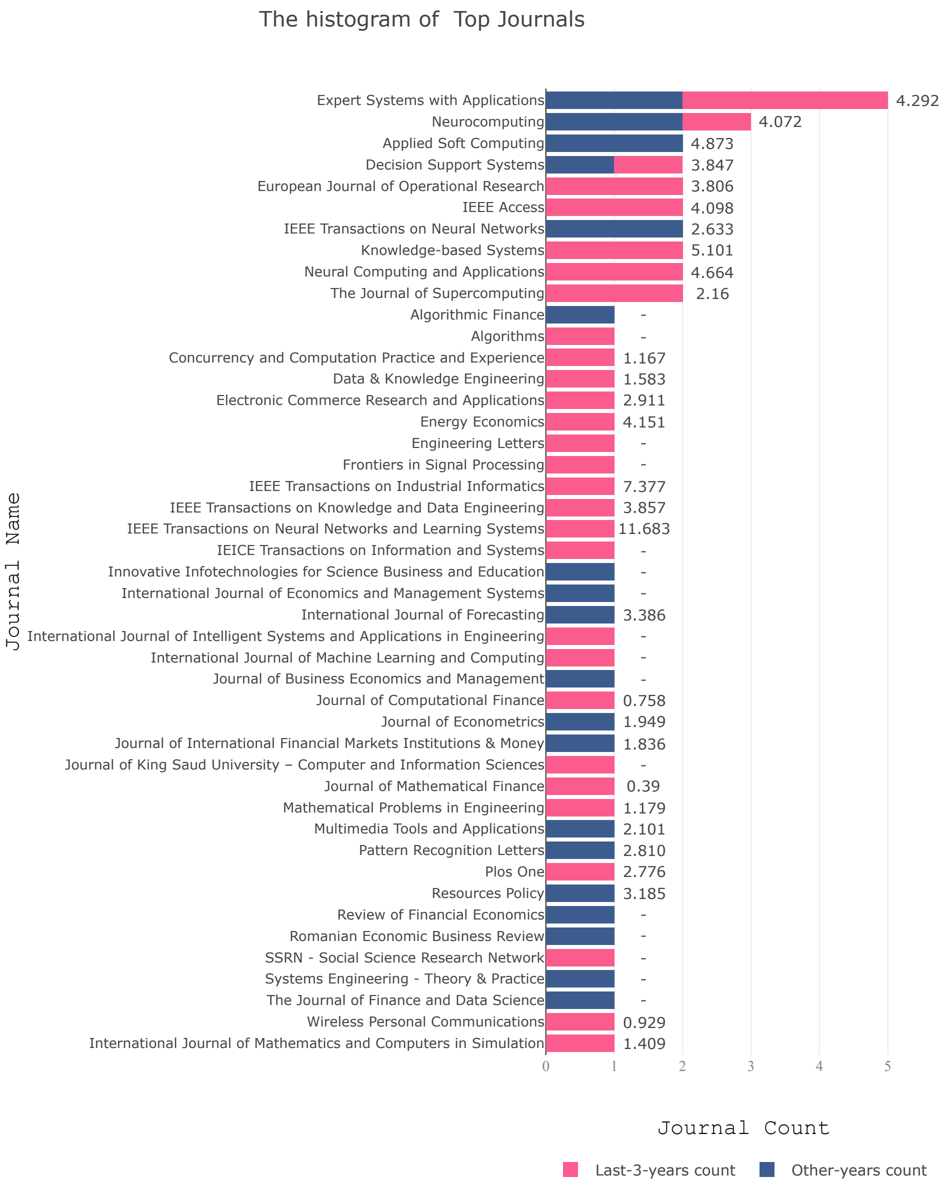

Figure 10: Top Journals - corresponding numbers next to the bar graph are representing the impact factor of the journals

exist, the 2-D implementation for CNN is more common, mostly inherited through image recognition applications of CNN from computer vision implementations. In some studies [188, 189, 193, 199, 219], innovative transformations of financial time series data into an image-like representation have been adapted, and impressive performances have been achieved. As a result, CNN might increase its share of interest for financial time series

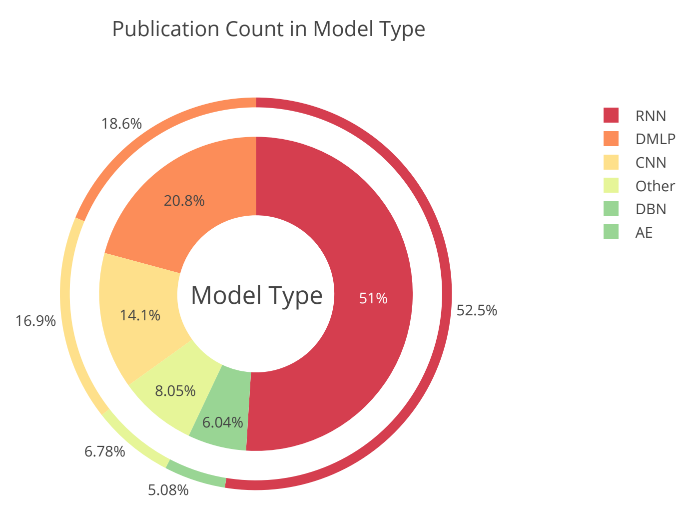

Figure 11: The Piechart of Publication Count in Model Types

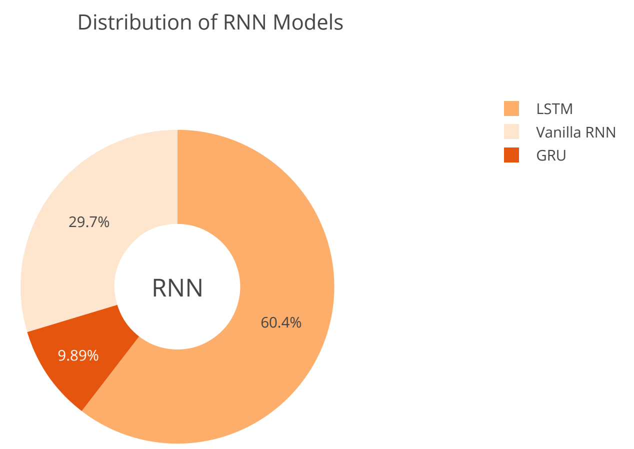

Figure 12: Distribution of RNN Models

forecasting in the next few years.

As one final note, Figure 13 shows which frameworks and platforms the researchers and developers used while implementing their work. We tried our best to extract this information from the papers. However, we must keep in mind that not every publication provided their development environment. Also, most papers did not give details, preventing us from a

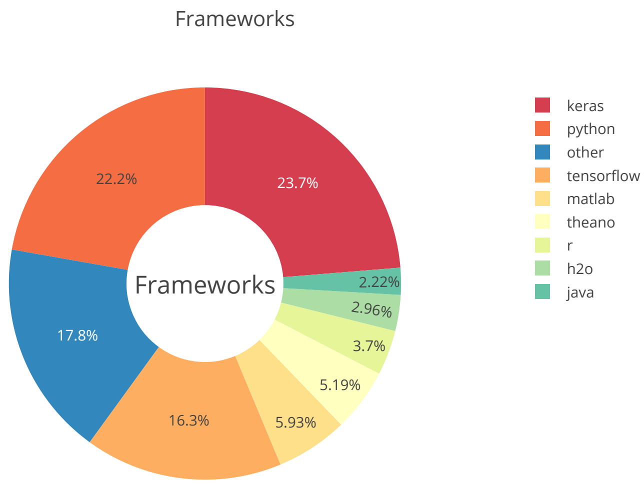

Figure 13: The Preferred Development Environments

more thorough comparison chart, i.e, some researchers claimed they used Python, but no further information was given, while some others mentioned the use of Keras or TensorFlow, providing more details. Also, within the “Other" section, the usage of Pytorch has increased in the last year or so, even though it is not visible from the chart. Regardless, Python-related tools were the most influential technologies behind the implementations covered in this survey.

6. Discussion and Open Issues

From an application perspective, even though financial time series forecasting has a relatively narrow focus, i.e., the implementations were mainly based on price or trend prediction, depending on the underlying DL model, very different and versatile models exist in the literature. We must remember that even though financial time series forecasting is a subset of time-series studies, due to the embedded profit-making expectations from successful prediction models, some differences exist, such that higher prediction accuracy sometimes might not reflect a profitable model. Hence, the risk and reward structure must also be taken into consideration. At this point, we will try to elaborate on our observations about these differences in various model designs and implementations.

6.1. DL Models for Financial Time Series Forecasting

According to the publication statistics, LSTM was the preferred choice of most researchers for financial time series forecasting. LSTM and its variations utilized time-varying data with feedback embedded representations, resulting in higher performances for time series prediction implementations. Because most financial data, one way or another, included time-dependent components, LSTM was the natural choice in financial time series forecasting problems. Meanwhile, LSTM is a special DL model derived from a more general classifier family, namely RNN.

Careful analysis of Figure 11 illustrates the dominance of RNNs (which mainly consist of LSTM). As a matter of fact, more than half of the published papers on time series forecasting fall into the RNN model category. Regardless of its problem type, price, or trend prediction, the ordinal nature of the data representation forced researchers to consider RNN, GRU, and LSTM as viable preferences for their model choices. Hence, RNN models were chosen, at least for benchmarking, in many studies for performance comparison with other developed models.

Meanwhile, other models were also used for time series forecasting problems. Among those, DMLP had the most interest due to the market dominance of its shallow cousin (MLP) and its wide acceptance and long history within ML society. However, there is a fundamental difference in how DMLP- and RNN-based models were used for financial time series prediction problems.

DMLP fits well for both regression and classification problems. However, in general, data order independence must be preserved to better utilize the internal working dynamics of such networks, even though some adjustments can be made through the learning algorithm configuration. In most cases, either trend components of the data need to be removed from the underlying time series or some data transformations might be needed so that the resulting data becomes stationary. Regardless, some careful preprocessing might be necessary for a DMLP model to be successful. In contrast, RNN-based models can work directly with time-varying data, making it easier for researchers to develop DL models.

As a result, most DMLP implementations had embedded data preprocessing before the learning stage. However, this inconvenience did not prevent researchers from using DMLP and its variations during their model development process. Instead, many versatile data representations were attempted to achieve higher overall prediction performances. A combination of fundamental and/or technical analysis parameters along with other features, such as financial sentiment through text mining, were embedded in such models. In most DMLP studies, the corresponding problem was treated as classification, especially in trend prediction models, whereas RNN-based models directly predicted the next value of the time series. Both approaches had some success in outperforming the underlying benchmark; hence it is not possible to claim superiority of one model type over another. However, as a general rule of thumb, researchers prefer RNN-based models for time series regression and DMLP for trend classification (or buy-sell point identification).

Another model that has increased in popularity recently is CNN. CNN also works better for classification problems, and unlike RNN-based models, it is more suitable for either non-time varying or static data representations. The comments made about DMLP are also mostly valid for CNN. Furthermore, unlike DMLP, CNN mostly requires locality within the data representation for better-performing classification results. One particular implementation area of CNN is image-based object recognition problems. In recent years, CNN-based models have dominated this field, handily outperforming all other models. Meanwhile, most financial data are time-varying, and it might not be easy to implement CNN directly for financial applications. However, in some recent studies, various independent research groups followed an innovative transformation of 1-D time-varying financial data into 2-D mostly stationary image-like data so that they could utilize the power of CNN through adaptive filtering and implicit dimensionality reduction. Hence, with that approach, they were able to develop successful models.

There is also a rising trend of using deep RL-based financial algorithmic trading implementations; these are mostly associated with various agent-based models, where different agents interact and learn from their interactions. This field has even more opportunities to offer with advancements in financial sentiment analysis through text mining to capture investor psychology; as a result, behavioral finance can benefit from these particular studies associated with RL-based learning models coupled with agent-based studies.

Other models including DBN, AE, and RBM, were also used by several researchers, and superior performances were reported in some of their work. However, interested readers should check these studies case by case to see how they were modelled from data representation and learning perspectives.

6.2. Discussions on Selected Features

Regardless of the underlying forecasting problem, the raw time series data was almost always somehow embedded directly or indirectly within the feature vector, which is particularly valid for RNN-based models. However, in most other model types, other features were also included. Fundamental analysis and technical analysis features were among the most favorable choices for stock/index forecasting studies.

Meanwhile, in recent years, financial text mining has gained particular attention, mostly for extracting investor/trader sentiment. The streaming flow of financial news, tweets, statements, and blogs allowed researchers to build better and more versatile prediction and evaluation models, integrating numerical and textual data. The general methodology involves extracting financial sentiment analysis through text mining and combining that information with fundamental/technical analysis data to achieve better overall performance. It is logical to assume that this trend will continue with the integration of more advanced text and NLP techniques.

6.3. Discussions on Forecasted Asset Types

Although forex price forecasting is always popular among researchers and practitioners, stock/index forecasting has always had the most interest among all asset groups. Regardless, price/trend prediction and algo-trading models were mostly embedded with these prediction studies.

These days, financial time series forecasting research on cryptocurrencies is a hot topic. Cryptocurrency price prediction has growing demand from the financial community. Because this topic is relatively new, we might see more studies and implementations in the near future due to high expectations and promising rewards.

There were also a number of publications in commodity price forecasting research, particularly for predicting the price of oil. Oil price prediction is crucial due to its tremendous effect on world economic activities and planning. Meanwhile, gold is considered a safe investment and almost every investor, at one time, considers allocating some portion of their portfolio for gold-related investments. In times of political uncertaintly, many people turn to gold to protect their savings. Although we did not encounter a noteworthy study for gold price forecasting, due to its historical importance, there might be opportunities in this area in years to come.

6.4. Open Issues and Future Work

Despite the general motivation for financial time series forecasting remaining fairly unchanged, the means of achieving financial goals vary depending on the choices and trade-off between traditional techniques and newly developed models. Because our fundamental focus is on the application of DL for financial time series studies, we will try to assess the current state of the research and extrapolate that into the future.

6.4.1. Model Choices for the Future

The dominance of RNN-based models for price/trend prediction will probably not disappear anytime soon, mainly due to their easy adaptation to most asset forecasting problems. Meanwhile, some enhanced versions of the original LSTM or RNN models, generally integrated with hybrid learning systems, are now becoming more common. Readers should check individual studies and assess their performances to see which one fits the best for their particular needs and domain requirements.

We have observed increasing interest in 2-D CNN implementations of financial forecasting problems by converting the time series into an image-like data type. This innovative methodology seems to work quite satisfactorily and provides promising opportunities. More studies of this kind will probably continue in the near future.

Nowadays, new models are generated through older models via modifying or enhancing existing models so that better performances can be achieved. Such topologies include Generative Adversarial Network (GAN), and Capsule networks. They have been used in various non-financial studies; however, financial time series forecasting has not been investigated for those models yet. As such, there can be exciting opportunities both from research and practical points of view.

Another DL model that has been investigated thoroughly is Graph CNN. Graphs can be used to represent portfolios, social networks of financial communities, fundamental analysis data, etc. Even though graph algorithms can directly be applied to such configurations, different graph representations can also be implemented for time series forecasting problems. Not much work has been done on this particular topic; however, through graph representations of time series data and implementing graph analysis algorithms or implementing CNN through these graphs are among the possibilities that researchers can choose.

As a final note for future models, we believe that deep RL- and agent-based models offer great opportunities for researchers. HFT algorithms and robo-advisory systems highly depend on automated algorithmic trading systems that can decide what to buy and when to buy without any human intervention. These aforementioned models can fit very well in such challenging environments. The rise of the machines will also lead to a technological (and algorithmic) arms race between Fintech companies and quant funds to be the best in their neverending search for “achieving alpha". New research in these areas can be exactly what is required.

6.4.2. Future Projections for Financial Time Series Forecasting

Most probably, for the foreseeable future, financial time series forecasting will have close research cooperation with other financial application areas, such as algorithmic trading and portfolio management, as was the case before. However, changes in the available data characteristics and introduction of new asset classes might not only alter the forecasting strategies of developers but also force developers to seek new or alternative techniques to better adapt to these new challenging working conditions. In addition, metrics such as Continuous Ranked Probability Score (CRPS) [221] for evaluating probability distributions might be included for more thorough analysis.

One rising trend, not only for financial time series forecasting, but for all intelligent decision support systems, is human-computer interaction and NLP research. Within that field, text mining and financial sentiment analysis are of particular importance to financial time series forecasting. Behavioral finance may benefit from new advancements in these fields.

To utilize the power of text mining, researchers have started developing new data representations, such as Stock2Vec [204], which can be useful for combining textual and numerical data for better prediction models. Furthermore, NLP-based ensemble models that integrate data semantics with time-series data might increase the accuracy of existing models.

One area that can significantly benefit from interconnected financial markets is automated statistical arbitrage trading model development. It has been used in forex and commodity markets before. In addition, many practitioners currently seek arbitrage opportunities in cryptocurrency markets [220] due to the huge number of coins available on various marketplaces. Price disruptions, high volatility, and bid-ask spread variations cause arbitrage opportunities across different platforms. Some opportunists develop software models that can track these price anomalies for instant materialization of profits. Also, it is possible to construct pairs of trading portfolios across different asset classes using appropriate models. It is possible that DL models can learn (or predict) these opportunities faster and more efficiently than classical rule-based systems. This will also benefit HFT studies, which are constantly looking for faster and more efficient trading algorithms and embedded systems with minimum latency. To achieve that, Graphics Processing Unit (GPU)- or Field Programmable Gate Array (FPGA)-based hardware solutions embedded with DL models can be utilized. There is a lack of research accomplished in this hardware aspect of financial time series forecasting and algorithmic trading. Given that there is sufficient computing power available, it is worth investigating the possibility of better algorithms, because the rewards are high.

6.5. Responses to our Initial Research Questions

We are now ready to address to our initially stated research questions. Based on our observations, our answers to these questions are as follows:

- Which DL models are used for financial time series forecasting ?

  Response: RNN-based models (particularly LSTM) are the most commonly used models. Meanwhile, CNN and DMLP have been used extensively in classification-type implementations (such as trend classification) as long as appropriate data processing is applied to the raw data.

- How does the performance of DL models compare with that of their traditional machine learning counterparts ?

  Response: In the majority of studies, DL models were better than ML ones. However, there were also many cases where their performances were comparable. There were even two particular studies ([82, 175]) where ML models performed better than DL models. Meanwhile, the preference of DL implementations over ML models is growing. Advances in computing power, availability of big data, superior performance, implicit feature learning capabilities, and user friendly model development environment for DL models are among the main reasons for this migration.

  One important issue that might be worth mentioning is the possibility of the publication bias of DL over ML models. Since DL is more recent than ML, a published successful DL implementation might attract more audience than a comparable successful ML model. Hence, the researchers implicitly might have an additional motivation to develop DL models. However, this is probably a valid concern for every academic publication regardless of the study area [222]. Meanwhile, in this survey, our aim was to extract the published DL studies for financial forecasting without any prior assumptions, so the reader can decide which model works best for them through their own experiences.

- What is the future direction of DL research for financial time series forecasting ?

  Response: NLP, semantics, and text mining-based hybrid models ensembled with time-series data might be more common in the near future.

7. Conclusions

Financial time series forecasting has been very popular among ML researchers for more than 40 years. The financial community had a new boost lately with the introduction of DL implementations for financial prediction research, and many new publications have appeared accordingly. In our survey, we wanted to review existing studies to provide a snapshot of the current research status of DL implementations for financial time series forecasting. We grouped the studies according to their intended asset classes along with the preferred DL model associated with the problem. Our findings indicate that although financial forecasting has a long research history, overall interest within the DL community is on the rise through utilization of new DL models; hence, many opportunities exist for researchers.

8. Acknowledgement

This work is supported by Scientific and Technological Research Council of Turkey (TUBITAK) grant no 215E248.

Glossary

AdaGrad Adaptive Gradient Algorithm. 6, 7

ADAM Adaptive Moment Estimation. 6–8, 13, 14, 33

AE Autoencoder. 5, 9, 13, 14, 18, 19, 26, 29–31, 46

AI Artificial Intelligence. 3

AIS Annealed Importance Sampling. 12

AMEX American Stock Exchange. 20, 21

ANN Artificial Neural Network. 3–5, 8, 11–13, 18, 20, 23, 24, 31, 33, 34

AR Active Return. 20

AR Autoregressive. 22

ARCH Autoregressive Conditional Heteroscedasticity. 32

ARIMA Autoregressive Integrated Moving Average. 19, 20, 30, 32

ARMA Autoregressive Moving Average. 27, 28, 32

AUC Area Under the Curve. 20

AUROC Area Under the Receiver Operating Characteristics. 33, 34, 36

B&H Buy and Hold. 25

BELM Basic Extreme Learning Machine. 37

Bi-GRU Bidirectional Gated Recurrent Unit. 35, 36

Bi-LSTM Bidirectional LSTM. 22, 23, 25

BIST Istanbul Stock Exchange Index. 20, 21, 23, 24

Bovespa Brazilian Stock Exchange. 23, 25

BPTT Back Propagation Through Time. 7, 8

BSE Bombay Stock Exchange. 26

CCI Commodity Channel Index. 27

CD Contrastive Divergence. 12, 13

CDAX German Stock Market Index Calculated by Deutsche Börse. 20

CDBN Continuous-valued Deep Belief Networks. 30

CDBN-FG Fuzzy Granulation with Continuous-valued Deep Belief Networks. 31

CNN Convolutional Neural Network. 1, 2, 5, 6, 10, 17–22, 25–42, 45–47, 49

CRBM Continuous Restricted Boltzman machine. 30

CRPS Continuous Ranked Probability Score. 48

CSE Colombo Stock Exchange. 18

CSI China Securities Index. 18, 22, 26, 28, 29

DA Direction Accuracy. 31

DAX The Deutscher Aktienindex. 29, 33

DBN Deep Belief Network. 1, 5, 12, 13, 18, 19, 27–31, 33–36, 46

DE Differential Evolution. 26

DFNN Deep Feedforward Neural Network. 17, 19, 21, 23, 29, 33

DGM Deep Neural Generative Model. 36, 37

DJI Dow Jones Index. 36

DJIA Dow Jones Industrial Average. 22–26, 36

DL Deep Learning. 1–5, 7, 9, 10, 13, 14, 16–21, 23, 24, 27–30, 35, 38–40, 44, 45, 47–49

DLR Deep Learning Representation. 37

DMLP Deep Multilayer Perceptron. 5–8, 10, 23, 30, 34, 35, 39–41, 45, 49

DNN Deep Neural Network. 6, 10, 18–21, 23–33, 35, 36

DOW30 Dow Jones Industrial Average 30. 23, 26, 28

DP Dynamic Programming. 15, 16

DPA Direction Prediction Accuracy. 21

DRL Deep Reinforcement Learning. 2, 5, 16

DRSE Deep Random Subspace Ensembles. 36

DWNN Deep and Wide Neural Network. 18, 19

EA Evolutionary Algorithm. 25, 27

EC Evolutionary Computation. 3, 4

EGARCH Exponential GARCH. 24

ELM Extreme Learning Machine. 18, 19, 37

EMA Exponential Moving Average. 27

EMD2FNN Empirical Mode Decomposition and Factorization Machine based Neural Network. 33, 34

ETF Exchange-Traded Fund. 28, 31, 33, 34

FCNN Fully Connected Neural Network. 37

FDDR Fuzzy Deep Direct Reinforcement Learning. 23, 24, 27, 28

FFNN Feedforward Neural Network. 13, 14, 20, 24, 30, 33

FHSO Firefly Harmony Search Optimization. 25, 30, 31

FLANN Functional Link Neural network. 27
FNN Fully Connected Neural Network. 7, 27, 33
FTSE London Financial Times Stock Exchange Index. 23, 25, 26, 28, 29, 31, 34
GA Genetic Algorithm. 4, 25, 34, 51
GAF Gramian Angular Field. 33
GAN Generative Adversarial Network. 21, 47
GAN-FD GAN for minimizing Forecast error loss and Direction prediction loss. 20
GARCH Generalised Auto-Regressive Conditional Heteroscedasticity. 28, 29, 31, 32, 52
GASVR GA with a SVR. 25, 26, 28, 51
GBT Gradient Boosted Trees. 18, 19
GDAX Global Digital Asset Exchange. 26
GLM Generalized Linear Model. 23
GML Generalized Linear Model. 25
GP Genetic Programming. 3, 4, 30
GPA The Gaussian Process Approach. 7, 8
GRU Gated-Recurrent Unit. 8, 18, 19, 21, 32–37, 40, 45
GS Grid Search. 7, 8, 10, 12–14, 16
GSPC S&P500 Commodity Price Index. 24
HAN Hybrid Attention Network. 36
HAR Heterogeneous Autoregressive Process. 25, 28, 51
HAR-GASVR HAR with GASVR. 24, 28, 29
HFT High Frequency Trading. 17, 28, 37, 47, 48
HIT Hit Rate. 23, 24
HMAE Heteroscedasticity Adjusted MAE. 29
HMM Hidden Markov Model. 35
HMRPSO Modified Version of PSO. 26, 27
HMSE Heteroscedasticity Adjusted MSE. 29
HR Hit Rate. 21, 29, 31
HS China Shanghai Shenzhen Stock Index. 26, 37
HSI Hong Kong Hang Seng Index. 22–26, 33
IBOVESPA Indice Bolsa de Valores de Sao Paulo. 34
IC Information Coeffiencient. 20
IR Information Ratio. 20
ISE Istanbul Stock Exchange Index. 24, 31
IXIC NASDAQ Composite Index. 24
KELM Kernel Extreme Learning Machine. 37
KL-Divergence Kullback Leibler Divergence. 12
KOSPI The Korea Composite Stock Price Index. 18, 23, 25, 26, 29
KSE Korea Stock Exchange. 34
LAR Linear Auto-regression Predictor. 22
LDA Latent Dirichlet Allocation. 26, 27, 37
LOB Limit Order Book Data. 37
LRNFIS Locally Recurrent Neuro-fuzzy Information System. 23, 25, 31
LSTM Long-Short Term Memory. 1, 2, 5, 8, 9, 17–30, 32–37, 40, 44, 45, 47, 49
MACD Moving Average Convergence and Divergence. 34, 36
MAD Mean Absolute Deviation. 18
MAE Mean Absolute Error. 18, 20, 21, 23, 24, 26, 29, 31, 33, 37
MAM Moving Average Mapping. 33
MAPE Mean Absolute Percentage Error. 18, 21–24, 26–29, 31, 33, 37
MASE Mean Standard Deviation. 23
MC Monte Carlo. 15, 16
MCC Matthew Correlation Coefficient. 22, 26, 36, 37
MDA Multilinear Discriminant Analysis. 20, 21
MDD Maximum Drawdown. 18
MDP Markov Decision Process. 14, 15
MI Mutual Information. 18
ML Machine Learning. 1–5, 7, 19, 25, 29, 30, 34, 35, 45, 49
MLP Multilayer Perceptron. 10, 18–21, 23, 24, 29–31, 34, 40, 45
MM Markov Model. 29
MODRL Multi-objective Deep Reinforcement Learning. 26
MoE Mixture of Experts. 24
MOEA Multiobjective Evolutionary Algorithm. 4
MRS Markov Regime Switching. 27
MS Manual Search. 7, 8, 10, 12–14, 16
MSE Mean Squared Error. 13, 14, 18, 20, 21, 23, 24, 26–29, 31–34
MSFE Mean Squared Forecast Error. 24
MSPE Mean Squared Prediction Error. 20
NASDAQ National Association of Securities Dealers Automated Quotations. 18–21, 23, 26, 28, 33, 34, 36
NIFTY National Stock Exchange of India. 22, 26, 36
NIKKEI Tokyo Nikkei Index. 22, 23, 25, 26, 31
NLP Natural Language Processing. 9, 46, 48, 49
NMAE Normalized Mean Absolute Error. 29, 31
NMSE Normalized Mean Square Error. 18, 29, 31
NN Neural Network. 5, 6, 10, 16, 29, 33, 35–37
norm-RMSE Normalized RMSE. 18
NSE National Stock Exchange of India. 18
NYMEX New York Mercantile Exchange. 27, 28
NYSE New York Stock Exchange. 18, 20, 21, 23, 34, 36
OCHL Open, Close, High, Low. 20, 24, 26
OCHLV Open, Close, High, Low, Volume. 18, 19,

References

[1] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications : A survey. *arXiv preprint arXiv:2002.05786*, 2020.

[2] Rafik A. Aliev, Bijan Fazlollahi, and Rashad R. Aliev. Soft computing and its applications in business and economics. In *Studies in Fuzziness and Soft Computing*, 2004.

[3] Ludmila Dymowa. *Soft Computing in Economics and Finance*. Springer Berlin Heidelberg, 2011.

[4] Boris Kovalerchuk and Evgenii Vityaev. *Data Mining in Finance: Advances in Relational and Hybrid Methods*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[5] Anthony Brabazon and Michael O'Neill, editors. *Natural Computing in Computational Finance*. Springer Berlin Heidelberg, 2008.

[6] Arash Bahrammirzaee. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, 19(8):1165–1195, June 2010.

[7] D. Zhang and L. Zhou. Discovering golden nuggets: Data mining in financial application. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 34(4):513–522, November 2004.

[8] Asunción Mochón, David Quintana, Yago Sáez, and Pedro Isasi Viñuela. Soft computing techniques applied to finance. *Applied Intelligence*, 29:111–115, 2007.

[9] Sendhil Mullainathan and Jann Spiess. Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, May 2017.

[10] Shu-Heng Chen, editor. *Genetic Algorithms and Genetic Programming in Computational Finance*. Springer US, 2002.

[11] Ma. Guadalupe Castillo Tapia and Carlos A. Coello Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *2007 IEEE Congress on Evolutionary Computation*. IEEE, September 2007.

[12] Antonin Ponsich, Antonio Lopez Jaimes, and Carlos A. Coello Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, June 2013.

[13] Ruben Aguilar-Rivera, Manuel Valenzuela-Rendon, and J.J. Rodriguez-Ortiz. Genetic algorithms and darwinian approaches in financial applications: A survey. *Expert Systems with Applications*, 42(21): 7684–7697, November 2015.

[14] Roy Rada. Expert systems and evolutionary computing for financial investing: A review. *Expert Systems with Applications*, 34(4):2232–2240, 2008.

[15] Yuhong Li and Weihua Ma. Applications of artificial neural networks in financial economics: A survey. In *2010 International Symposium on Computational Intelligence and Design*. IEEE, October 2010.

[16] Michal Tkáč and Robert Verner. Artificial neural networks in business: Two decades of research. *Applied Soft Computing*, 38:788 – 804, 2016.

[17] B. Elmsili and B. Outtaj. Artificial neural networks applications in economics and management research: An exploratory literature review. In *2018 4th International Conference on Optimization and Applications (ICOA)*, pages 1–6, April 2018.

[18] Marc-André Mittermayer and Gerhard F Knolmayer. Text mining systems for market response to news: A survey. September 2006.

[19] Leela Mitra and Gautam Mitra. Applications of news analytics in finance: A review. In *The Handbook of News Analytics in Finance*, pages 1–39. John Wiley & Sons, Ltd., May 2012.

[20] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, November 2014.

[21] Colm Kearney and Sha Liu. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, 33:171–185, May 2014.

[22] B. Shravan Kumar and Vadlamani Ravi. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114:128–147, December 2016.

[23] Frank Z. Xing, Erik Cambria, and Roy E. Welsch. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73, October 2017.

[24] Bruce J Vanstone and Clarence Tan. A survey of the application of soft computing to investment and financial trading. In Brian C Lovell, Duncan A Campbell, Clinton B Fookes, and Anthony J Maeder, editors, *Proceedings of the Eighth Australian and New Zealand Intelligent Information Systems Conference (ANZIIS 2003)*, pages 211–216. The Australian Pattern Recognition Society, 2003. Copyright The Australian Pattern Recognition Society 2003. All rights reserved. Permission granted.

[25] Ehsan Hajizadeh, H. Davari Ardakani, and Jamal Shahrabi. Application of data mining techniques in stock markets : A survey. 2010.

[26] Binoy B. Nair and V.P. Mohandas. Artificial intelligence applications in financial forecasting – a survey and some empirical results. *Intelligent Decision Technologies*, 9(2):99–140, December 2014.

[27] Rodolfo C. Cavalcante, Rodrigo C. Brasileiro, Victor L.F. Souza, Jarley P. Nobrega, and Adriano L.I. Oliveira. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55:194–211, August 2016.

[28] Bjoern Krollner, Bruce J. Vanstone, and Gavin R. Finnie. Financial time series forecasting with machine learning techniques: a survey. In *ESANN*, 2010.

[29] P. D. Yoo, M. H. Kim, and T. Jan. Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 835–841, November 2005.

[30] G Preethi and B Santhi. Stock market forecasting techniques: A survey. *Journal of Theoretical and Applied Information Technology*, 46:24–30, December 2012.

[31] George S. Atsalakis and Kimon P. Valavanis. Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36(3):5932–5941, April 2009.

[32] Amitava Chatterjee, O.Felix Ayadi, and Bryan E. Boone. Artificial neural network and the financial markets: A survey. *Managerial Finance*, 26(12):32–45, December 2000.

[33] R. Katarya and A. Mahajan. A survey of neural network techniques in market trend analysis. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 873–877, December 2017.

[34] Yong Hu, Kang Liu, Xiangzhou Zhang, Lijun Su, E.W.T. Ngai, and Mei Liu. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing*, 36:534–551, November 2015.

[35] Wei Huang, K. K. Lai, Y. Nakamori, and Shouyang Wang. Forecasting foreign exchange rates with artificial neural networks: A review. *International Journal of Information Technology & Decision Making*, 03(01):145–165, 2004.

[36] Dadabada Pradeepkumar and Vadlamani Ravi. Soft computing hybrids for forex rate prediction: A comprehensive review. *Computers & Operations Research*, 99:262 – 284, 2018.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[38] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[39] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[40] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[41] Barry L Kalman and Stan C Kwasny. Why tanh: choosing a sigmoidal function. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581. IEEE, 1992.

[42] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[43] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[44] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[45] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[46] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[47] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[48] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[49] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[50] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[54] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[55] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[56] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[57] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[58] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[59] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[60] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[61] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

[62] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[63] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

[64] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[65] Xueheng Qiu, Le Zhang, Ye Ren, P. Suganthan, and Gehan Amaratunga. Ensemble deep learning for regression and time series forecasting. In *2014 IEEE Symposium on Computational Intelligence in*

Ensemble Learning (CIEL), pages 1–6, 2014.

[66] Rafael Hrasko, André GC Pacheco, and Renato A Krohling. Time series prediction using restricted boltzmann machines and backpropagation. *Procedia Computer Science*, 55:990–999, 2015.

[67] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[68] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36, 2012.

[69] Abdel-rahman Mohamed, George Dahl, and Geoffrey Hinton. Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, volume 1, page 39. Vancouver, Canada, 2009.

[70] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.

[71] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, pages 384–391, 2009.

[72] Chengwei Yao and Gencai Chen. Hyperparameters adaptation for restricted boltzmann machines based on free energy. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 243–248. IEEE, 2016.

[73] Miguel A Carreira-Perpinan and Geoffrey E Hinton. On contrastive divergence learning. In *Aistats*, volume 10, pages 33–40. Citeseer, 2005.

[74] Prasanna Tamilselvan and Pingfeng Wang. Failure diagnosis using deep belief learning based health state classification. *Reliability Engineering & System Safety*, 115:124–135, 2013.

[75] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[76] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J Kennedy. Relational autoencoder for feature extraction. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 364–371. IEEE, 2017.

[77] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[78] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[79] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

[80] Eunsuk Chong, Chulwoo Han, and Frank C. Park. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83:187–205, October 2017.

[81] Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, October 2015.

[82] Eva Dezsi and Ioan Alin Nistor. Can deep machine learning outsmart the market? a comparison between econometric modelling and long- short term memory. *Romanian Economic Business Review*, 11(4.1):54–73, December 2016.

[83] A.J.P. Samarawickrama and T.G.I. Fernando. A recurrent neural network approach in predicting daily stock prices an application to the sri lankan stock market. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*. IEEE, December 2017.

[84] M Hiransha, E.A. Gopalakrishnan, Vijay Krishna Menon, and K.P. Soman. Nse stock market prediction using deep-learning models. *Procedia Computer Science*, 132:1351–1362, 2018.

[85] Sreelekshmy Selvin, R Vinayakumar, E. A Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference*

on Advances in Computing, Communications and Informatics (ICACCI). IEEE, September 2017.

[86] Sang Il Lee and Seong Joon Yoo. Threshold-based portfolio: the role of the threshold and its applications. *The Journal of Supercomputing*, September 2018.

[87] Xiumin Li, Lin Yang, Fangzheng Xue, and Hongjun Zhou. Time series prediction of stock price using deep belief networks with intrinsic plasticity. In *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, May 2017.

[88] Lin Chen, Zhilin Qiao, Minggang Wang, Chao Wang, Ruijin Du, and Harry Eugene Stanley. Which artificial intelligence algorithm better predicts the chinese stock market? *IEEE Access*, 6:48625–48633, 2018.

[89] Christopher Krauss, Xuan Anh Do, and Nicolas Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259 (2):689–702, June 2017.

[90] Rohitash Chandra and Shelvin Chand. Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance. *Applied Soft Computing*, 49:462–473, December 2016.

[91] Shuanglong Liu, Chao Zhang, and Jinwen Ma. Cnn-lstm neural network model for quantitative strategy analysis in stock markets. In *Neural Information Processing*, pages 198–206. Springer International Publishing, 2017.

[92] J. B. Heaton, N. G. Polson, and J. H. Witte. Deep learning in finance, 2016.

[93] Bilberto Batres-Estrada. Deep learning for multivariate financial time series. Master's thesis, KTH, Mathematical Statistics, 2015.

[94] Zhaozheng Yuan, Ruixun Zhang, and Xiuli Shao. Deep and wide neural networks on multiple sets of temporal data with correlation. In *Proceedings of the 2018 International Conference on Computing and Data Engineering - ICCDE 2018*. ACM Press, 2018.

[95] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD17*. ACM Press, 2017.

[96] Masaya Abe and Hideki Nakayama. Deep learning for forecasting stock returns in the cross-section. In *Advances in Knowledge Discovery and Data Mining*, pages 273–284. Springer International Publishing, 2018.

[97] Guanhao Feng, Jingyu He, and Nicholas G. Polson. Deep learning for predicting asset returns, 2018.

[98] Jianqing Fan, Lingzhou Xue, and Jiawei Yao. Sufficient forecasting using factor models. *SSRN Electronic Journal*, 2014.

[99] Mathias Kraus and Stefan Feuerriegel. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, 104:38–48, December 2017.

[100] Shotaro Minami. Predicting equity price with corporate action events using lstm-rnn. *Journal of Mathematical Finance*, 08(01):58–63, 2018.

[101] Xiaolin Zhang and Ying Tan. Deep stock ranker: A lstm neural network model for stock selection. In *Data Mining and Big Data*, pages 614–623. Springer International Publishing, 2018.

[102] Qun Zhuge, Lingyu Xu, and Gaowei Zhang. Lstm neural network with emotional analysis for prediction of stock price. 2017.

[103] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, June 2016.

[104] A. Ozbayoglu. Neural based technical analysis in stock market forecasting. In *Intelligent Engineering Systems Through Artificial Neural Networks, Volume 17*, pages 261–266. ASME, 2007.

[105] Kaustubh Khare, Omkar Darekar, Prafull Gupta, and V. Z. Attar. Short term stock price prediction using deep learning. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, May 2017.

[106] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, 2018:1–11,

2018.
[107] Ritika Singh and Shashi Srivastava. Stock prediction using deep learning. *Multimedia Tools and Applications*, 76(18):18569–18584, December 2016.
[108] Sercan Karaoglu and Ugur Arpaci. A deep learning approach for optimization of systematic signal detection in financial trading systems with big data. *International Journal of Intelligent Systems and Applications in Engineering*, SpecialIssue(SpecialIssue):31–36, July 2017.
[109] Bo Zhou. Deep learning and the cross-section of stock returns: Neural networks combining price and fundamental information. *SSRN Electronic Journal*, 2018.
[110] Narek Abroyan and. Neural networks for financial market risk classification. *Frontiers in Signal Processing*, 1(2), August 2017.
[111] Google. System and method for computer managed funds to outperform benchmarks.
[112] Dat Thanh Tran, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Tensor representation in high-frequency financial data for price change prediction. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, November 2017.
[113] Guanhao Feng, Nicholas G. Polson, and Jianeng Xu. Deep factor alpha, 2018.
[114] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2327–2333. AAAI Press, 2015.
[115] Manuel R. Vargas, Beatriz S. L. P. de Lima, and Alexandre G. Evsukoff. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, June 2017.
[116] Che-Yu Lee and Von-Wun Soo. Predict stock price with financial news based on recurrent convolutional neural networks. In *2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, December 2017.
[117] Hitoshi Iwasaki and Ying Chen. Topic sentiment asset pricing with dnn supervised learning. *SSRN Electronic Journal*, 2018.
[118] Sushree Das, Ranjan Kumar Behera, Mukesh Kumar, and Santanu Kumar Rath. Real-time sentiment analysis of twitter streaming data for stock prediction. *Procedia Computer Science*, 132:956–964, 2018.
[119] Jiahong Li, Hui Bu, and Junjie Wu. Sentiment-aware stock market prediction: A deep learning method. In *2017 International Conference on Service Systems and Service Management*. IEEE, June 2017.
[120] Zhongshengz. Measuring financial crisis index for risk warning through analysis of social network. Master's thesis, 2018.
[121] Janderson B. Nascimento and Marco Cristo. The impact of structured event embeddings on scalable stock forecasting models. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web - WebMedia15*. ACM Press, 2015.
[122] Songqiao Han, Xiaoling Hao, and Hailiang Huang. An event-extraction approach for business analysis from online chinese news. *Electronic Commerce Research and Applications*, 28:244–260, March 2018.
[123] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, 12(7):e0180944, July 2017.
[124] A.K. Parida, R. Bisoi, and P.K. Dash. Chebyshev polynomial functions based locally recurrent neuro-fuzzy information system for prediction of financial and energy market data. *The Journal of Finance and Data Science*, 2(3):202–223, September 2016.
[125] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, October 2018.
[126] Philip Widegren. Deep learning-based forecasting of financial assets. Master's thesis, KTH, Mathematical Statistics, 2017.
[127] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Dilated convolutional neural networks for time series forecasting. *Journal of Computational Finance*, October 2018.

[128] Khaled A. Althelaya, El-Sayed M. El-Alfy, and Salahadin Mohammed. Evaluation of bidirectional lstm for short-and long-term stock market prediction. In *2018 9th International Conference on Information and Communication Systems (ICICS)*. IEEE, April 2018.

[129] Alexiei Dingli and Karl Sant Fournier. Financial time series forecasting–a deep learning approach. *Int. J. Mach. Learn. Comput*, 7(5):118–122, 2017.

[130] Ajit Kumar Rout, P.K. Dash, Rajashree Dash, and Ranjeeta Bisoi. Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach. *Journal of King Saud University - Computer and Information Sciences*, 29(4):536–552, October 2017.

[131] Gyeeun Jeong and Ha Young Kim. Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications*, 117:125–138, March 2019.

[132] Yujin Baek and Ha Young Kim. 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*, 113:457–480, December 2018.

[133] Magnus Hansson. On stock return prediction with lstm networks. 2017.

[134] Aaron Elliot and Cheng Hua Hsu. Time series prediction : Predicting stock price, 2017.

[135] Zhixi Li and Vincent Tam. Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, November 2017.

[136] Sima Siami-Namini and Akbar Siami Namin. Forecasting economics and financial time series: Arima vs. lstm, 2018.

[137] Tsung-Jung Hsieh, Hsiao-Fen Hsiao, and Wei-Chang Yeh. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Applied Soft Computing*, 11(2):2510–2525, March 2011.

[138] Luna M. Zhang. Genetic deep neural networks using different activation functions for financial data mining. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, October 2015.

[139] Stelios D. Bekiros. Irrational fads, short-term memory emulation, and asset predictability. *Review of Financial Economics*, 22(4):213–219, November 2013.

[140] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, and Victor Chang. An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, January 2018.

[141] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664, March 2017.

[142] Bing Yang, Zi-Jia Gong, and Wenqi Yang. Stock market index prediction using deep neural network ensemble. In *2017 36th Chinese Control Conference (CCC)*. IEEE, July 2017.

[143] Oussama Lachiheb and Mohamed Salah Gouider. A hierarchical deep neural network design for stock returns prediction. *Procedia Computer Science*, 126:264–272, 2018.

[144] Bang Xiang Yong, Mohd Rozaini Abdul Rahim, and Ahmad Shahidan Abdullah. A stock market trading system using deep neural network. In *Communications in Computer and Information Science*, pages 356–364. Springer Singapore, 2017.

[145] Serdar Yümlü, Fikret S. Gürgen, and Nesrin Okay. A comparison of global, recurrent and smoothed-piecewise neural models for istanbul stock exchange (ise) prediction. *Pattern Recognition Letters*, 26 (13):2093–2103, October 2005.

[146] Hongju Yan and Hongbing Ouyang. Financial time series prediction based on deep learning. *Wireless Personal Communications*, 102(2):683–700, December 2017.

[147] Takahashi. Long memory and predictability in financial markets. *Annual Conference of the Japanese Society for Artificial Intelligence*, 2017.

[148] Melike Bildirici, Elçin A. Alp, and Özgür Ö. Ersin. Tar-cointegration neural network model: An empirical analysis of exchange rates and stock returns. *Expert Systems with Applications*, 37(1):2–11, January 2010.

[149] Ioannis Psaradellis and Georgios Sermpinis. Modelling and trading the u.s. implied volatility indices.

evidence from the vix, vxn and vxd indices. *International Journal of Forecasting*, 32(4):1268–1283, October 2016.

[150] Jiaqi Chen, Wenbo Wu, and Michael Tindall. Hedge fund return prediction and fund selection: A machine-learning approach. Occasional Papers 16-4, Federal Reserve Bank of Dallas, November 2016.

[151] Marios Mourelatos, Christos Alexakos, Thomas Amorgianiotis, and Spiridon Likothanassis. Financial indices modelling and trading utilizing deep learning techniques: The athens se ftse/ase large cap use case. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*. IEEE, July 2018.

[152] Yuzhou Chen, Junji Wu, and Hui Bu. Stock market embedding and prediction: A deep learning method. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*. IEEE, July 2018.

[153] Weiyu Si, Jinke Li, Peng Ding, and Ruonan Rao. A multi-objective deep reinforcement learning approach for stock index future's intraday trading. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. IEEE, December 2017.

[154] Weiling Chen, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee. Leveraging social media news to predict stock index movement using rnn-boost. *Data & Knowledge Engineering*, August 2018.

[155] Matthew Francis Dixon, Diego Klabjan, and Jin Hoon Bang. Classification-based financial markets prediction using deep neural networks. *SSRN Electronic Journal*, 2016.

[156] Fernando Sánchez Lasheras, Francisco Javier de Cos Juez, Ana Suárez Sánchez, Alicja Krzemień, and Pedro Riesgo Fernández. Forecasting the comex copper spot price by means of neural networks and arima models. *Resources Policy*, 45:37–43, September 2015.

[157] Yang Zhao, Jianping Li, and Lean Yu. A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, 66:9–16, August 2017.

[158] Yanhui Chen, Kaijian He, and Geoffrey K.F. Tso. Forecasting crude oil prices: a deep learning based model. *Procedia Computer Science*, 122:300–307, 2017.

[159] Jonathan Doering, Michael Fairbank, and Sheri Markose. Convolutional neural networks applied to high-frequency market microstructure forecasting. In *2017 9th Computer Science and Electronic Engineering (CEEC)*. IEEE, September 2017.

[160] P. Tino, C. Schittenkopf, and G. Dorffner. Financial volatility trading using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12(4):865–874, July 2001.

[161] Ruoxuan Xiong, Eric P. Nichols, and Yuan Shen. Deep learning stock volatility with google domestic trends, 2015.

[162] Yu-Long Zhou, Ren-Jie Han, Qian Xu, and Wei-Ke Zhang. Long short-term memory networks for csi300 volatility prediction with baidu search volume. 2018.

[163] Ha Young Kim and Chang Hyun Won. Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications*, 103:25–37, August 2018.

[164] Nikolay Nikolaev, Peter Tino, and Evgueni Smirnov. Time-dependent series variance learning with recurrent mixture density networks. *Neurocomputing*, 122:501–512, December 2013.

[165] Campbell R. Harvey. Forecasts of economic growth from the bond and stock markets. *Financial Analysts Journal*, 45(5):38–45, 1989.

[166] Daniele Bianchi, Matthias Büchner, and Andrea Tamoni. Bond risk premia with machine learning. *SSRN Electronic Journal*, September 2018.

[167] Venketas Warren. Forex market size: A traders advantage, 2019.

[168] Ren Zhang, Furao Shen, and Jinxi Zhao. A model with fuzzy granulation and deep belief networks for exchange rate forecasting. In *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2014.

[169] Jing Chao, Furao Shen, and Jinxi Zhao. Forecasting exchange rate with deep belief networks. In *The 2011 International Joint Conference on Neural Networks*. IEEE, July 2011.

[170] Jing Zheng, Xiao Fu, and Guijun Zhang. Research on exchange rate forecasting based on deep belief network. *Neural Computing and Applications*, May 2017.

[171] Furao Shen, Jing Chao, and Jinxi Zhao. Forecasting exchange rate using deep belief networks and

conjugate gradient method. *Neurocomputing*, 167:243–253, November 2015.

[172] Hua Shen and Xun Liang. A time series forecasting model based on deep learning integrated algorithm with stacked autoencoders and svr for fx prediction. In *ICANN*, 2016.

[173] Georgios Sermpinis, Jason Laws, Andreas Karathanasopoulos, and Christian L. Dunis. Forecasting and trading the eur/usd exchange rate with gene expression and psi sigma neural networks. *Expert Systems with Applications*, 39(10):8865–8877, August 2012.

[174] Georgios Sermpinis, Christian Dunis, Jason Laws, and Charalampos Stasinakis. Forecasting and trading the eur/usd exchange rate with stochastic neural network combination and time-varying leverage. *Decision Support Systems*, 54(1):316–329, December 2012.

[175] Georgios Sermpinis, Charalampos Stasinakis, and Christian Dunis. Stochastic and genetic neural network combinations in trading and hybrid time-varying leverage effects. *Journal of International Financial Markets, Institutions and Money*, 30:21–54, May 2014.

[176] Bo SUN and Chi XIE. Rmb exchange rate forecasting in the context of the financial crisis. *Systems Engineering - Theory & Practice*, 29(12):53–64, December 2009.

[177] Nijolė Maknickienė and Algirdas Maknickas. Financial market prediction system with evolino neural network and deplhi method. *Journal of Business Economics and Management*, 14(2):403–413, May 2013.

[178] Nijole Maknickiene, Aleksandras Vytautas Rutkauskas, and Algirdas Maknickas. Investigation of financial market prediction by recurrent neural network. 2014.

[179] Luca Di Persio and Oleksandr Honchar. Artificial neural networks approach to the forecast of stock market price movements. *International Journal of Economics and Management Systems*, (1):158–162, 2016.

[180] Jerzy Korczak and Marcin Hernes. Deep learning for financial time series forecasting in a-trader system. In *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*. IEEE, September 2017.

[181] Gonçalo Duarte Lima Freire Lopes. Deep learning for market forecasts. 2018.

[182] Sean McNally, Jason Roche, and Simon Caton. Predicting the price of bitcoin using machine learning. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, March 2018.

[183] Sanjiv Das, Karthik Mokashi, and Robbie Culkin. Are markets truly efficient? experiments using deep learning algorithms for market movement prediction. *Algorithms*, 11(9):138, September 2018.

[184] Ariel Navon and Yosi Keller. Financial time series prediction using deep learning, 2017.

[185] E.W. Saad, D.V. Prokhorov, and D.C. Wunsch. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9 (6):1456–1470, 1998.

[186] Luca Di Persio and Oleksandr Honchar. Recurrent neural networks approach to the financial forecast of google assets. *International Journal of Mathematics and Computers in Simulation*, 11:713, 2017.

[187] Guizhu Shen, Qingping Tan, Haoyu Zhang, Ping Zeng, and Jianjun Xu. Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, 131:895–903, 2018.

[188] Jou-Fan Chen, Wei-Lun Chen, Chun-Ping Huang, Szu-Hao Huang, and An-Pin Chen. Financial time-series data analysis using deep convolutional neural networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, November 2016.

[189] Omer Berat Sezer and Ahmet Murat Ozbayoglu. Financial trading model with stock bar chart image time series with deep convolutional neural networks. *arXiv preprint arXiv:1903.04610*, 2019.

[190] Feng Zhou, Hao min Zhou, Zhihua Yang, and Lihua Yang. Emd2fnn: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, 115:136–151, 2019.

[191] Kristiina Ausmees, Slobodan Milovanovic, Fredrik Wrede, and Afshin Zafari. Taming deep belief networks. 2017.

[192] Kamran Raza. Prediction of stock market performance by using machine learning techniques. In *2017*

International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT). IEEE, April 2017.

[193] Omer Berat Sezer, Murat Ozbayoglu, and Erdogan Dogdu. A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters. *Procedia Computer Science*, 114:473–480, 2017.

[194] Qiubin Liang, Wenge Rong, Jiayi Zhang, Jingshuang Liu, and Zhang Xiong. Restricted boltzmann machine based stock market trend prediction. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017.

[195] Luigi Troiano, Elena Mejuto Villa, and Vincenzo Loia. Replicating a trading strategy by means of lstm for financial industry applications. *IEEE Transactions on Industrial Informatics*, 14(7):3226–3234, July 2018.

[196] David M. Q. Nelson, Adriano C. M. Pereira, and Renato A. de Oliveira. Stock markets price movement prediction with lstm neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017.

[197] Yuan Song and Yingnian Wu. Stock trend prediction: Based on machine learning methods. Master's thesis, 2018.

[198] M. Ugur Gudelek, S. Arda Boluk, and A. Murat Ozbayoglu. A deep learning based stock trading model with 2-d cnn trend detection. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, November 2017.

[199] Omer Berat Sezer and Ahmet Murat Ozbayoglu. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538, September 2018.

[200] Hakan Gunduz, Yusuf Yaslan, and Zehra Cataltepe. Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems*, 137:138–148, December 2017.

[201] Yifu Huang, Kai Huang, Yang Wang, Hao Zhang, Jihong Guan, and Shuigeng Zhou. Exploiting twitter moods to boost financial trend prediction based on deep network models. In *Intelligent Computing Methodologies*, pages 449–460. Springer International Publishing, 2016.

[202] Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016.

[203] Huy D. Huynh, L. Minh Dang, and Duc Duong. A new model for stock price movements prediction using deep neural network. In *Proceedings of the Eighth International Symposium on Information and Communication Technology - SoICT 2017*. ACM Press, 2017.

[204] L. Minh Dang, Abolghasem Sadeghi-Niaraki, Huy D. Huynh, Kyungbok Min, and Hyeonjoon Moon. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access*, pages 1–1, 2018.

[205] Ishan Verma, Lipika Dey, and Hardik Meisheri. Detecting, quantifying and accessing impact of news events on indian stock indices. In *Proceedings of the International Conference on Web Intelligence - WI17*. ACM Press, 2017.

[206] Leonardo dos Santos Pinheiro and Mark Dras. Stock market prediction with deep learning: A character-based neural language model for event-based trading. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 6–15, 2017.

[207] Jordan Prosky, Xingyou Song, Andrew Tan, and Michael Zhao. Sentiment predictability for stocks. *CoRR*, abs/1712.05785, 2017.

[208] Yang Liu, Qingguo Zeng, Huanrui Yang, and Adrian Carrio. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In *Knowledge Management and Acquisition for Intelligent Systems*, pages 102–113. Springer International Publishing, 2018.

[209] Akira Yoshihara, Kazuki Fujikawa, Kazuhiro Seki, and Kuniaki Uehara. Predicting stock market trends by recurrent deep neural networks. In *Lecture Notes in Computer Science*, pages 759–769.

Springer International Publishing, 2014.
[210] Lei Shi, Zhiyang Teng, Le Wang, Yue Zhang, and Alexander Binder. Deepclue: Visual interpretation of text-based deep stock prediction. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2018.
[211] Xi Zhang, Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fang, and Philip S. Yu. Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems*, 143:236–247, March 2018.
[212] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 261–269, New York, NY, USA, 2018. ACM.
[213] Qili Wang, Wei Xu, and Han Zheng. Combining the wisdom of crowds and technical analysis for financial market prediction using deep random subspace ensembles. *Neurocomputing*, 299:51–61, July 2018.
[214] Takashi Matsubara, Ryo Akita, and Kuniaki Uehara. Stock price prediction by deep neural generative model of news articles. *IEICE Transactions on Information and Systems*, E101.D(4):901–908, 2018.
[215] Xiaodong Li, Jingjing Cao, and Zhaoqing Pan. Market impact analysis via deep learned architectures. *Neural Computing and Applications*, March 2018.
[216] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning to detect price change indications in financial markets. In *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, August 2017.
[217] Justin Sirignano and Rama Cont. Universal features of price formation in financial markets: Perspectives from deep learning. *SSRN Electronic Journal*, 2018.
[218] Przemyslaw Buczkowski. Predicting stock trends based on expert recommendations using gru/lstm neural networks. In *Lecture Notes in Computer Science*, pages 708–717. Springer International Publishing, 2017.
[219] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*. IEEE, July 2017.
[220] Thomas G Thomas Günter Fischer, Christopher Krauss, and Alexander Deinert. Statistical arbitrage in cryptocurrency markets. *Journal of Risk and Financial Management*, 12, 2019.
[221] T Gneiting and A E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
[222] Hannah R Rothstein, Alex J Sutton, and Michael Borenstein. Publication bias in meta-analysis – prevention, assessment and adjustment. *John Wiley & Sons, Ltd*, 2005.