

Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks

Muhammad Zulqarnain

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Parit Raja, Batu Pahat, Johor, Malaysia

E-mail: zulqarnainmalik321@gmail.com

Rozaida Ghazali, Muhammad Ghulam Ghouse, Yana Mazwin Mohmad Hassim and Irfan Javid

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Parit Raja, Batu Pahat, Johor, Malaysia

E-mail: rozaida@uthm.edu.my, apik.md@gmail.com, yana@uthm.edu.my, irfanjavid87@gmail.com

Received: 08 March 2020; Accepted: 14 June 2020; Published: 08 December 2020

Abstract: Financial time-series prediction has been long and the most challenging issues in financial market analysis. The deep neural networks is one of the excellent data mining approach has received great attention by researchers in several areas of time-series prediction since last 10 years. Convolutional neural network (CNN) and recurrent neural network (RNN) models have become the mainstream methods for financial predictions. In this paper, we proposed to combine architectures, which exploit the advantages of CNN and RNN simultaneously, for the prediction of trading signals. Our model is essentially presented to financial time series predicting signals through a CNN layer, and directly fed into a gated recurrent unit (GRU) layer to capture long-term signals dependencies. GRU model perform better in sequential learning tasks and solve the vanishing gradients and exploding issue in standard RNNs. We evaluate our model on three datasets for stock indexes of the Hang Seng Indexes (HSI), the Deutscher Aktienindex (DAX) and the S&P 500 Index range 2008 to 2016, and associate the GRU-CNN based approaches with the existing deep learning models. Experimental results present that the proposed GRU-CNN model obtained the best prediction accuracy 56.2% on HIS dataset, 56.1% on DAX dataset and 56.3% on S&P500 dataset respectively.

Index Terms: Deep learning, RNN, GRU, CNN, financial time series prediction.

1. Introduction

Financial time series analysis is the core value in the scope of various research areas, and gained much attention in several engineering problems. The time series data is relates to calculating useful features and pattern features of sequence data. During a period of time, the observations of time series data is collected and taken by sequentially. A time series is a collection of observations of data items taken sequentially during a period of time. Predicting financial time series is greatly complex, due to the substantially the most highly-noise characteristics and the strong form of market effectiveness, accepted through the general. In recent years, Deep Learning models have been applied in many time series financial applications, in particular automatic time series predictions and computer vision [1]. These approaches have also attained and enhanced the performance in financial sequence predictions, when compared to conventional machine learning algorithms. For illustration, some researches on investment market abnormalities have illustrated more than 130 include deviations efficiently overcome the market relying on return predictive signals. These signals are used as features (input) in order to predict particular financial return as the targets (output) by using deep learning approaches. However due to the powerful feature learning ability, auto encoder (AE), convolutional neural network (CNN) and recurrent neural network (RNN) become the famous approaches and have been used in the financial sequence prediction tasks [2].

In order to apply the benefits of these two deep learning techniques concurrently, mixtures of CNN and RNN turn into a research trends. The further method is to stack RNN on CNN. Its deviations have achieved significant results in several supplementary time series predictions tasks [3]. In recently years, machine learning (ML) techniques have presented in which capable to excellently extract the non-linear dimensionality in the complicated financial market data. Especially, Huck [4] organizes recurrent neural networks joint with several criteria of decision-making technique on completely S&P 100 components from 1990 to 2004. They reached directional accuracy of 54.2% on approximately 0.9% returns per week when $k = 6$ (is the stocks buy or sell number in the operation procedure). Motwani [5] Proposed an exhaustive momentum scheme base on deep neural networks (DNNs). Guresen in 2011 had conducted an experiment

to investigate about the efficiency of deep learning approaches in stock market prediction [6]. The experimental applications on the state-of-art tradeoff duration since the U.S. CRSP stock data among 1964 and 2008 generate returns of 46.12% each year. As a related model is executes by [7]. They established the frequencies to the high level with binning returns data each five minutes, and result in better classification accuracy more than 74 percent.

Aloysius [8] Proposed a statistical arbitrage technique based on deep restrictive contingent portfolio categories apply random forests (RFs). Moreover to adjust the return of U.S on per month CRSP stock world between the 1967 to 2011

Furthermore, the average per month return which is risk-adjusted is two percent on U.S. CRSP stock world between 1967 and 2011. All constituent on S&P 500 from 1991 to 2014 are sequence predicting by using deep learning approaches RFs, gradient-learning method is the combination of them. In [9], Gatris-Estrada describes about various applications of deep learning models on time series analysis. According to the obtainable literatures, several models have used are all memory-free, the financial stock marketing prediction is required for the goal of predicting the long short term of stock company. By taking advantage of latest digital computation, the deep learning approaches are very effective for determining non-linear structures in the data of the financial market. Therefore, for this kind of tasks, the recurrent architecture with memory function is operationally more appropriate.

In this research, we follow the combination way and design two networks architecture: Our contributions are summarized as follows: First, we concentrate on revising the standard gated recurrent unit (GRU) model, which greatly addresses vanishing gradient and exploding issue of standard RNNs over the gating mechanism and make the ordinary architecture and retaining the impact of LSTM (is a well-known variant of traditional RNNs). Second, we proposed to combine GRU with CNN architecture to identify financial marketing predictions based on the return predictive signals. Third, we also trained our model with attention mechanism (GRU-CNN) and compare the performance with the traditional deep learning models. Fourth, experiments show that our enhance GRU-CNN model achieves better predictions performance than previous traditional methods. In statistics and economically, the existing GRU-based model obtains good accuracies and more returns. But the proposed GRU-CNN model performs slightly better than GRU-based model.

The rest of this paper is organized as follows: the related work is introduced in section II, while traditional deep learning models are discussed in Section III. The problem identification in standard RNN is given in section IV. The detail of the proposed GRU-CNN architecture and flowchart are described in section V. The experimental setup is provided in section VI. Results and discussion are illustrated in section VII. Finally, this work is summarized conclusion in section 5.

2. Related Works

Deep Learning (DL) approaches has been widely used to measure of problems are included computer vision, natural language processing [10], automatic time series forecasting [11], hand-writing recognition [12], and financial time series prediction [13] to name a few, outperformed existing baseline scheme. However, some DL approaches were used financial time series data, and have been applied many classification tasks are included, text-based classification, portfolio optimization, volatility predicting and price base prediction. Rönqvist [14] predicted the financial risk by using DL technique base on the news articles. To compute the level of banking related report which is publicly accessible in the form of textual sequential data and classification approaches was trained on required sentences as distress. For the purpose of accurate prediction two deep learning approaches were used, the aims of first approach is trained by semantic representation to reduce dimensionality reduction, while the other approach is trained to classify the learned representation of each sentence. Mach [15] to predict German stock returns based on news headlines by using text-based classifier, and reported 55% accuracy on a three-classes based prediction of the following trade day, without developing a trading strategy. Ding [16] studied to predicts daily S&P 500 movements by using structured information extracted from headlines.

The authors [17] worked on S&P 500 volatility based on the standard LSTM model by applying the Google stock domestic trends as an indicator of the market volatility. LSTM is a powerful model to addresses the vanishing gradient issues which are common issues in the training of traditional RNN. Dixon [18] proposed deep neural network to trained model using multiple financial instruments for prediction of market trends and also classify the future trend as either positive, flat or negative. X Ding *et al* in [16] conducted a research on financial time series prediction with combination of Natural language processing (NLP). For stock market prediction in [19], authors applied two machine learning approaches such as Least Square Support Vector Machine (LSSVM) and Particle Swarm Optimization (PSO). The dataset consists of combined all symbols with training feature sets, each engineered features and price differences encoded by 9889. The deep learning models usually contain of five fully connected layers and the model is show the prediction of instrument's trend however ignore operation costs.

3. Deep Models

Deep learning techniques were derived from artificial neural networks and nowadays it is a predominant arena of

machine learning and has achieved an outstanding performance in several research areas, like as time series prediction, computer vision and NLP. However, its practicality deep learning becomes more prevalent for several researchers to do research works.

3.1. Convolutional neural network

A convolutional neural network (CNNs) are one of the most widely biologically inventive kind of forward deep neural network (DNN) that has recently achieved popularity due to its achievement in many fields of research such as classification problems and financial time series predictions [20]. CNNs architecture is mainly contained on three layers are included convolution layer, pooling layer and fully connected layer. It demonstrates to be capable to extracts high level features concept from sequential representation of input data, and then combines these features by the subsequent layer in order to achieve higher order features. Each single layer of a CNN uses a differentiable function in order to transfer information from one volume of activations to the next. Convolutional layer is a fundamental layer of a CNN and involved most of the computation process [21]. Training iterations (epochs) uses gradient decent to train the convolutional layer and model parameters are combined by a Rectified Linear Units (ReLU) layer to support non-linear function [8]. The pooling or subsampling layers of CNNs extracts the values of their filter size during the training stage based on the tasks. The schematic presentation of CNN is shown in Fig.1.

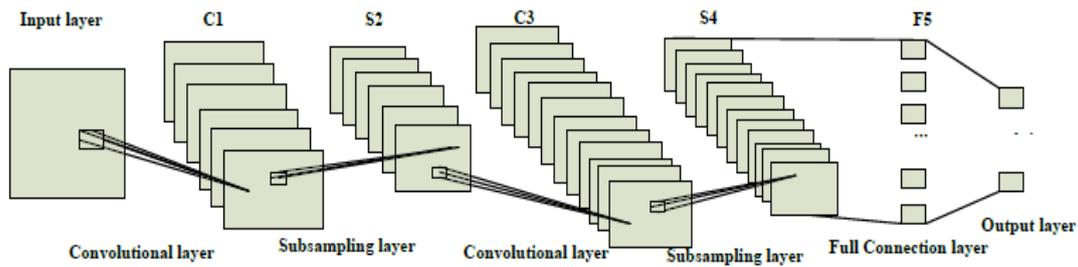


Fig.1. A schematic presentation of CNN

A further kind of layer, the max pooling layer, is commonly used in CNNs. It works on data to compresses and makes smooth data. Max-layer selects the maximum value of the receptive field and produces data invariant to small translational changes [22]. Consequently, generate three CNNs layers to manage various data prediction due to their variances in sizes. When apply subsampling layers and for final output using fully connected layer. This structure of CNN enables the model to acquire filters that is capable to identify particular features in the input data. Recently advance in CNNs for sequential time series prediction included by [23] where the researcher have to proposed an un-decimated convolutional networks for time series modelling based on the un-decimated wavelet transform. Instinctively, the concept of using CNNs to time series prediction would be to determine filters that show the convinced recurrence features in the series and apply to predict the future values. Due to the layered architecture of CNNs, they able to perform better on noisy data, by removal noise in each subsequently layer and capturing only the useful information [24].

3.2. Recurrent neural network

A Recurrent Neural Networks (RNNs) is a kind of supervised neural networks that was initially introduced by Hopfield in 1983. It has connections between the nodes form a bidirectional cycle, and they perform well on the sequential tasks. The RNN approach performs excellently on sequential problem specially extracting temporal information in the loop. RNN is capable to compute a sequential of random dimension through recursively using a transition function to its internal hidden state vector h_t of the sequential inputs. The structure of an RNN is presented in Fig.2. When sequence inputs are required as $X = [x_1, x_2 \dots x_t \dots x_T]$ of dimension T, O_t is a output vector, and U_o are present the hidden state at the time t of the RNN are summarized the following equations:

$$\vec{O}_t = \sigma(\vec{W}_x x_t + \vec{U}_o \vec{h}_{t-1}) \tag{1}$$

$$\vec{O}_t = \sigma(\vec{W}_x x_t + \vec{U}_o \vec{h}_{t+1}) \tag{2}$$

$$O_t = [\vec{O}_t : \vec{O}_t] \tag{3}$$

where $W_x \in R^{O_h \times O_x}$ is refer the weights matrix connection between input layer to hidden layer, and hidden layers weights matrix are presented by $U_o \in R^{O_h \times O_h}$. σ is the sigmoid activation function and W_x, U_o , are parameters of the existing RNN.

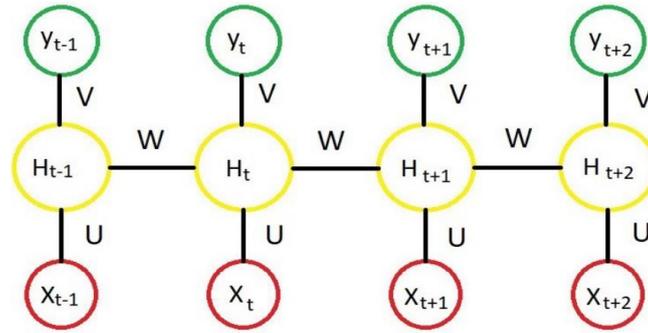


Fig.2. Standard architecture of RNNs

h_t refer to the hidden state at time-step t is computed by O_t and the previous hidden state h_{t-1} . Whereas the RNN is an excellent model for handling sequential problems, it is hard to train with the gradient descent method and suffer from vanishing and exploding gradient (explosion) issue. In contrast, the variants of RNN have been introduced to solve this issue, such as Gated Recurrent Unit (GRU), it avoids overfitting, as well as saves training time. Therefore, we adopted GRU in our technique.

3.3. GRU based model

The gated recurrent unit GRU is comparatively recent development introduced by Cho et al [25], that addresses the common issues of long term dependencies which can leads to poor gradients for larger traditional RNN networks. The GRU is a simplified variation of the LSTM and it has gating mechanism that controls the flow of information inside the unit without having a separate memory cells. GRU is much simpler to calculate because it has two gates called update gate and reset gate are applied to appropriately extract dependencies through various time scales: the reset gate r_t , decide control information from the previous time step is kept in the candidate hidden state; another is the update gate z_t , which decides to handle previous information through away and how much information from the candidate hidden state is added.

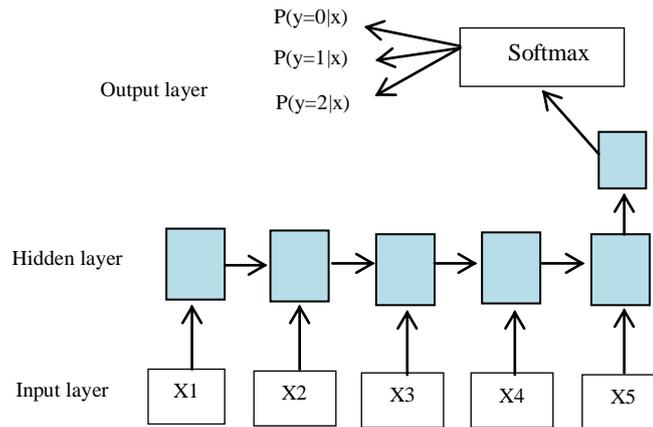


Fig.3. Structure of GRU-based model.

The input layer of the model is composed of multiple neurons is presented in Fig. 3, the number of neurons is decided by the dimension of the features space. The design of GRU combines forget and input gates into a single unit named “update gate”, and has an additional “reset gate” that handle the flow of information inside the unit without having an individual memory cells [26]. GRU also demonstrates the superior ability of modeling to capture long-term dependencies among the sequential elements and have to gain progressive popularity.

To calculated these gates for time step t by the following equations:

Update gate:

$$z_t = \sigma_g (W^z . [h_{t-1}, x_t] + b_z) \tag{4}$$

Reset gate:

$$r_t = \sigma_g (W^r . [h_{t-1}, x_t] + b_r) \tag{5}$$

Candidate state

$$\tilde{h}_t = \tanh(W.[r_t * h_{t-1}, x_t] + b_{\tilde{h}}) \quad (6)$$

Final output

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (7)$$

where W^z , W^r , and W^x are the weights of update gate, reset gate, and candidate state accordingly, and the biases of these gates are presented by b^z , b^r and b^h . Equations (4) and (5) are applied to calculate the update and reset gates respectively. \tilde{h}_t is the candidate hidden state function defined in (6) which is applied to calculate the hidden activation h given by (7)

4. Problem Identification

Define Recurrent neural networks (RNNs) are prevalent architecture that can perform well on sequence dataset. The concept behind RNNs is to store similar portions of the inputs and apply this information to predicting the output in the future. Therefore, the RNNs are most suitable for time series predictions. However, unfortunately the RNNs often suffer from the vanishing gradient and exploding issues, which performance in unsuccessful to capture long-term dependencies. It makes the training of RNN difficult, in two ways: firstly, it cannot process very long sequences if using hyperbolic tanh activation function and secondly, if network is unfolded based on the several time steps which some of the RNN weights initiates to convert too large or too small due to gradient and exploding issues. Based on observation, to avoid this issue, two variations of RNNs have been introduced, which applies a gating mechanism to handle these issues. A latest kind of RNNs called gated recurrent unit (GRU) proposed by Cho *et al* in 2014 [25] that tackle this kind of issue.

Based on the ordinary implementation, to calculate units composed by gates in order to reduce the error in the block, due to replace the hidden layer by in complex block, constructing known as error carousel. Financial sequence predictions contract with the capturing of the basic features to evaluate and the temporary dynamics prediction of financial assets. Due to the inherent uncertainty and non-analytic structure the prior works in this field concentrates on technical analysis, the task showed to be challenging in the prediction of financial markets inherent uncertainty and non-analytic architecture, where classical linear statistical approaches are including the ARIMA model and statistical machine learning (ML) approaches have been extensively used time series tasks [18].

5. GRU-CNN Proposed Architecture

In this section, we further investigate to introduce integrated design of the recurrent convolutional neural network model for the financial prediction of trading signals is presented in Fig. 4. GRU is ability of learning some dependencies which efficiently remove the vanishing gradient issue. GRU is basically similar to an LSTM controls the information inside the unit but it has no output gate. The activation h_t is referring to the final output of GRU at time t among the previous state h_{t-1} and the candidate state \hat{h}_t :

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (8)$$

The update gate z_t help the model to control how much of the previous information updates its activation.

$$z_t = \text{Sigm}(W_{xz}x_t + U_{hz}h_{t-1}) \quad (9)$$

The reset gate r_t determine how much of the previous information to ignored its activation.

$$r_t = \text{Sigm}(W_{xr}x_t + U_{hr}h_{t-1}) \quad (10)$$

The candidate activation \hat{h}_t is computed similarly to the update gate:

$$\tilde{h}_t = \tanh(W_{x\tilde{h}}x_t + U_{h\tilde{h}}(r_t * h_{t-1})) \quad (11)$$

We replace hyperbolic tangent activation function (\tanh) with (Relu) in equation (11) the new equation is:

$$\tilde{h}_t = \text{Relu}(W_{x\tilde{h}}x_t + U_{h\tilde{h}}(r_t * h_{t-1})) \quad (12)$$

where Sigm is a sigmoid activation and $*$ is refer an element-wise multiplication. Reset gate allows the unit of ignore

previous information, when it is very close to zero or off ($r_t \approx 0$). The reset gate is computed through the following equation:

$$r_t = \text{Sigm}(W_{sr}x_t + U_{hr}h_{t-1}) \tag{13}$$

Update gate z_t updates the following information sending by previous activation with time step t and pass to the next step. On the other hand, reset gate r_t control the short-term dependencies and long-term dependencies has controlled by update gates z_t in the GRU model. In this paper, we addresses the limitation of the above models, we proposed a Recurrent Convolutional Neural Network and it apply for the task of financial sequence prediction. We construct the time series prediction model (GRU-CNN financial sequence prediction model) with GRU and CNN deep learning model. It mainly includes five parts of layers namely, input layer, convolutional layer, GRU layer, pooling layer and finally fully connected layer are presented in Fig. 4.

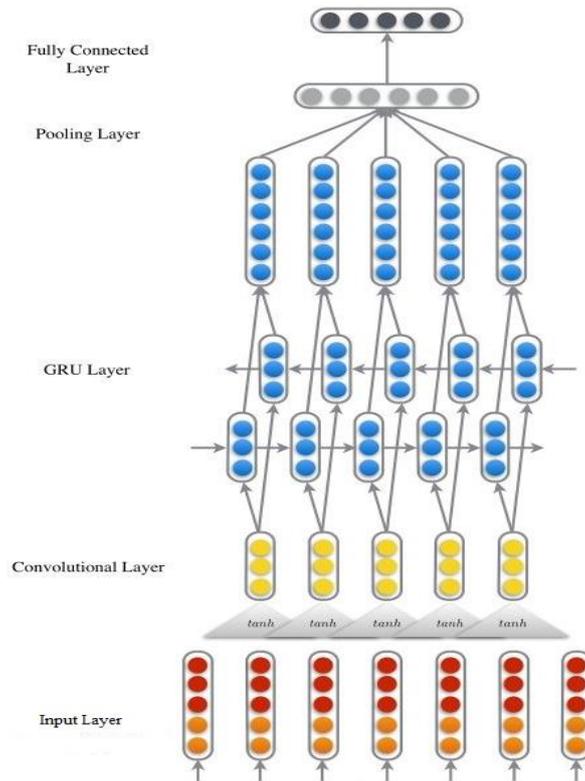


Fig.4. Proposed GRU-CNN architecture

In Fig.4 illustration, the first layer is input layer for preliminary study of data values. In the first layer of CNN, the original values are observed into a high dimensional space, so that the original values prediction can be distinguished better. The 2nd layer of CNN is convolutional layer that perform convolution operation can extract this information by combining high dimension features in a fixed window. Provided the sequential inputs representation $x = (x_1, x_2, \dots, x_n)$ and a context window length k , concatenation of sequential prediction in this window length can be determined as $X_j = [x_j^T, \dots, x_{j+k-1}^T]^T$, and the representation of this sequential prediction sample can be reformatted as $X = (X_1, \dots, X_{n-k+1})$. Provided a weight matrices of the convolutional filters W^{conv} and a linear bias b , the local features representation are calculated:

$$C_j = \tanh(W^{conv} \cdot X_j + b) \tag{14}$$

where $W^{conv} \in \mathbb{R}^{dc \times dx \times k}$, $b \in \mathbb{R}^{dc}$, and \tanh denotes the hyperbolic tangent function. Generally, the convolutional layer output generated from Eq. (14) is further processed as inputs to the GRU layers passing through the reset gate and update gate based on the Eq. (9) and Eq. (10) to perform the most considerable prediction. However, these sequential predictions are independent. GRU has the capability to handle this weakness by applying a gating architecture to extract short-term and long-term dependencies. Therefore, in this research, a GRU layer is built on top of the convolutional layer to continue the financial sequential prediction task. Note that sequence financial prediction using fully connected layer is perform similar as a softmax layer. The difference between fully connected layer and softmax is in only their goals parametrized by all weights matrix W . Softmax layer reduces the cross-entropy or increases the log-likelihood; however

to find out the maximum margin among the data points and various classes is efficiently performed by max-pooling layer.

In this research, we concentrate overall performance of the proposed architecture to predict the stock market trading signals. Fig.5 is presenting the flowchart of the proposed algorithm for predict the trading signals.

In order to train the GRU networks efficiently in this research we developed to apply three basic methods. Initially we set the RMSprop optimizer is derived from rprop, and applied mini-batch in the model training. This technique is usually select for RNNs approaches. Secondly, the dropout regularization applied in the hidden layers in order to avoid overfitting. According to a certain probability, the neural network units discarded from the network in the training procedure. Thirdly, we have to use early stopping criteria in this research to achieve the same purpose as a further mechanism. Moreover, the dataset is splitted according to a specific ratio into training set and validation set. The former set is applied for training, while obtained test results by verification set (e.g., every 5-epoch is considered for test).

The best amount of the post-sample data between the samples is verified as 20%, the data divided into training and validation set according to the following ratio of 4:1. We set 10000 epochs in the training process and the maximum early stopping duration to 10. By this means, we provided a particular explanation of the proposed GRU-CNN model as follow:

Input layer with 250 time-steps consisting one feature.

The hidden layer of GRU consists of 27 hidden neurons and 0.5 selected as dropout rate.

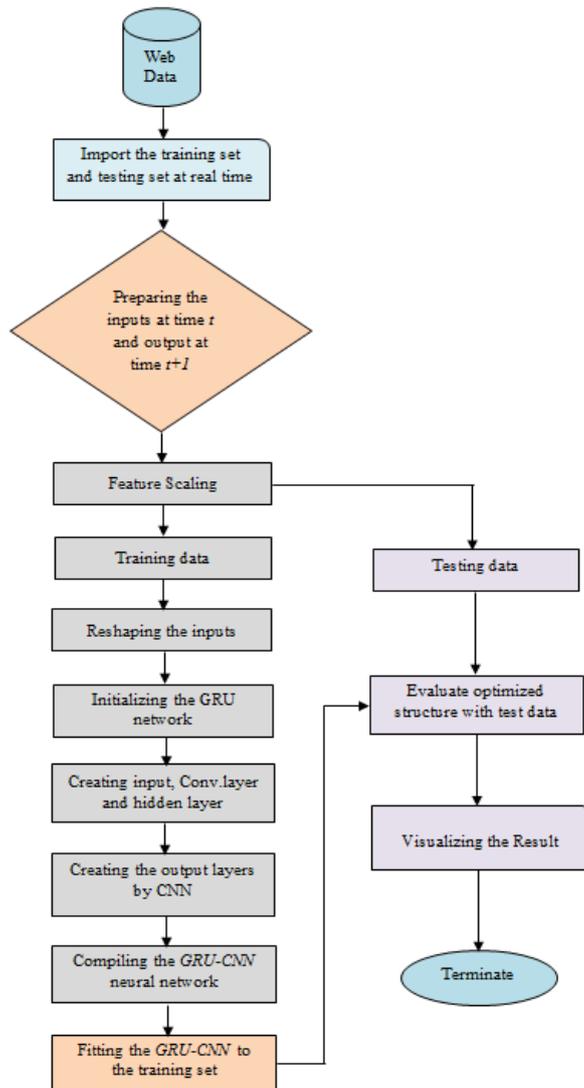


Fig.5. Flowchart of model for predicting the stock market trading signals

5.1. Deep GRU-CNN and Benchmark models

In this section, for the traditional GRU and the proposed GRU-CNN model have mention above, we optimize to combined recurrent neural networks (RNNs) to present the advantage of the existing GRU network, and evaluates the

performance of conventional CNN. Specially, the input features (standardized returns) and the output targets are general in the similar method as signified in the previous subsection.

In this paper, a typically organized a comparative study of deep learning approaches with our proposed GRU-CNN model in order to evaluate the performance and advantage of the model. The input layer consists of 246 neurons according to the input features, and three fully connected nodes are selected in the output layer. For learning high statistical features, we have to select three hidden layers, each layer consist number of several neurons from the first layer to the third layer is 1, 2 and 3 individually. The model training simulated is executed 5 times for each combination of momentum. We applied rectified linear unit (Lelu) as the activation function and learning rate 0.001 as well as 0.5 set as a dropout. Based on the grid search approach these parameters are optimizing to avoid overfitting and under-fitting. For the purpose of validation, we apply k-fold cross validation with k=5 in the data partition. By comparative analysis of the proposed model with different state-of-the-art approaches, we proved that the proposed GRU-based approaches with CNN are capable to achieve more significant results than other state-of-the-art approaches through more complicated and intensive computations.

6. Experimental Setup

In this section, we conducted an experimental study to evaluate the proposed GRU-CNN model for financial sequence predictions on three datasets: HIS dataset, DAX dataset and S&P 500 Index dataset to train & test our model. We compared proposed model with other baseline approaches include, gated recurrent unit (GRU), deep neural network (DNN) and convolutional neural network (CNN).

6.1. Datasets & Software

The first step in this research is Data exploration. In this research the database are contains specific representative in financial marketing predictions by Asia, Europe and the Americas [27] publicly available and downloaded through Yahoo Finance for the period between 2008 and 2016. In this research, we train and test proposed and comparative models on three financial marketing data are namely, HSI, the DAX and the S&P 500 Index as the raw data. Because the main focus of these experiments are completely demonstration financial market and without the instability of separate stocks, for the trading probability technique the high-liquidity are selected as a subsets of the stock market is a most actual test set for computed probability. Statistics summary of experimental datasets is explained in Table 1.

Table 1. Summary of experimental datasets.

Index	Time period	Sample size
HIS	18/05/2008 – 14/05/2016	6640
DAX	18/05/2008 – 11/05/2016	6812
S&P 500	22/05/2008 – 10/05/2016	6719

Data preprocessing and manipulate are carried out in Python 3.6 and anaconda depending on the packages scipy, sklearn, numpy and pandas. For quick implementation of the proposed model and traditional deep learning models are implemented, an effective open source software library for numerically computational using data flow graphs in which permits an ordinary and rapid develop. All the simulation works were carried out on Intel Core i7-3770CPU on a Windows PC with @3.40 GHz, and 4GB RAM machine.

6.2. Generation of training and trading sets

The study analysis on the daily basis of data between 2008 to 2016 are collected the training and trading period is determined training-trading sets. The previous set about 750 days, almost three years for sample training. The latter used for out-of-sample trading is set to 250 days, equivalent to one year. With this structure, we have offered lot of training samples for the proposed architecture in section 3 to be evaluated. Based on this, the sliding-window approach is referred; the training-trading set is move forward by a length of 250 days. Furthermore, the 24 batches are not overlapped with each other looping over the entire dataset among 2008 and 2016 as given in Table 1.

7. Results and Discussion

In this study, we have conducted the experiment setup on three stock indexes datasets (HIS, DAX and S&P 500) and compared the results of proposed GRU-CNN model with three various kinds of deep learning models namely are, GRU, GRU-CNN, DNN and CNN for predict the trading signals on each time-step through the entire time period on these datasets.

For every index, accuracy that has been used as an evaluation metric for trading sets to evaluate the performance of proposed GRU-CNN model and comparative approaches, for the proportion of accurate classification it is an important metric has been achieved by computing in deep learning.

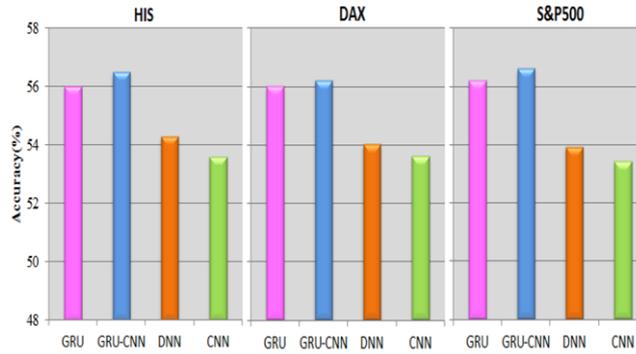


Fig.6. Accuracy comparison of proposed models with existing deep learning models of 3 stock indexes (HIS, DAX and S&P 500) from 1990 until 2016.

In addition, we evaluated the efficiency and effectiveness of the proposed model by daily mean returns and cumulative returns of the trading periods. The formulas calculate the daily mean return and cumulative returns as a follows:

$$Daily\ Returns[\%] = \frac{P_t - P_{t-1}}{P_{t-1}} \times 1 \tag{15}$$

$$Accumulative\ Returns = [R_c] = \frac{P_{current} - P_{initial}}{P_{initial}} \tag{16}$$

where p_t is a current price, p_{t-1} is a previous price and * denotes as a multiplication element.

According to the following trends we reported the accuracies of various deep learning approaches are presented in Fig.6. The performance of the standard GRU model and the proposed GRU-CNN model in the term of accuracy are always greater than 55% on the stock indexes dataset, it is significant benchmark for a dollar neutral scheme, its presents that our proposed approach for analyzing the prediction tasks on financial time series. Regardless this kind of dataset, the proposed GRU-CNN model performs excellent as compared to other state-of-the-art approaches. Experimental results has presented that the proposed model achieved 56.2% results in term of daily returns and Accumulative returns, while GRU is obtain 55.8% , DNN is achieve 53.9%, and 53.5% is achieved by CNN for S&P 500. In this way, a same condition is occur in other two indexes, the proposed GRU-CNN model have achieved excellent performance, with the exclusion of the DAX are presented in Fig.7 and Fig.8.

The exiting GRU based model and proposed GRU-CNN model are show greater return over entire data sets than other baseline DNN and CNN, the proposed and comparative models have slightly achieved higher values on HIS and the S&P 500 as compare to DAX. However, based on the associated characteristics of the datasets the profitability of the similar approach on various datasets is different. Therefore, in this paper both of the proposed approaches demonstration remarkable profitability in the similar circumstances. The experimental results shown that the proposed GRU-CNN model is feasible and efficient in the stock index predicting, in which our model shows the better result, and the overall prediction of existing GRU based model is also superior than DNN and CNN models.

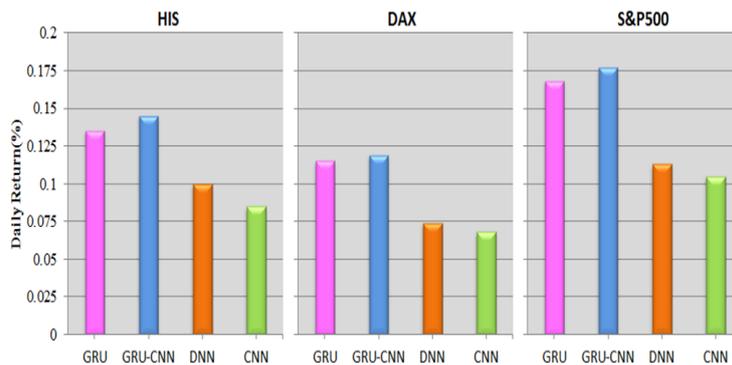


Fig.7. Daily mean returns of 3 stock indexes (HIS, DAX and S&P 500) from 1991 until 2017.

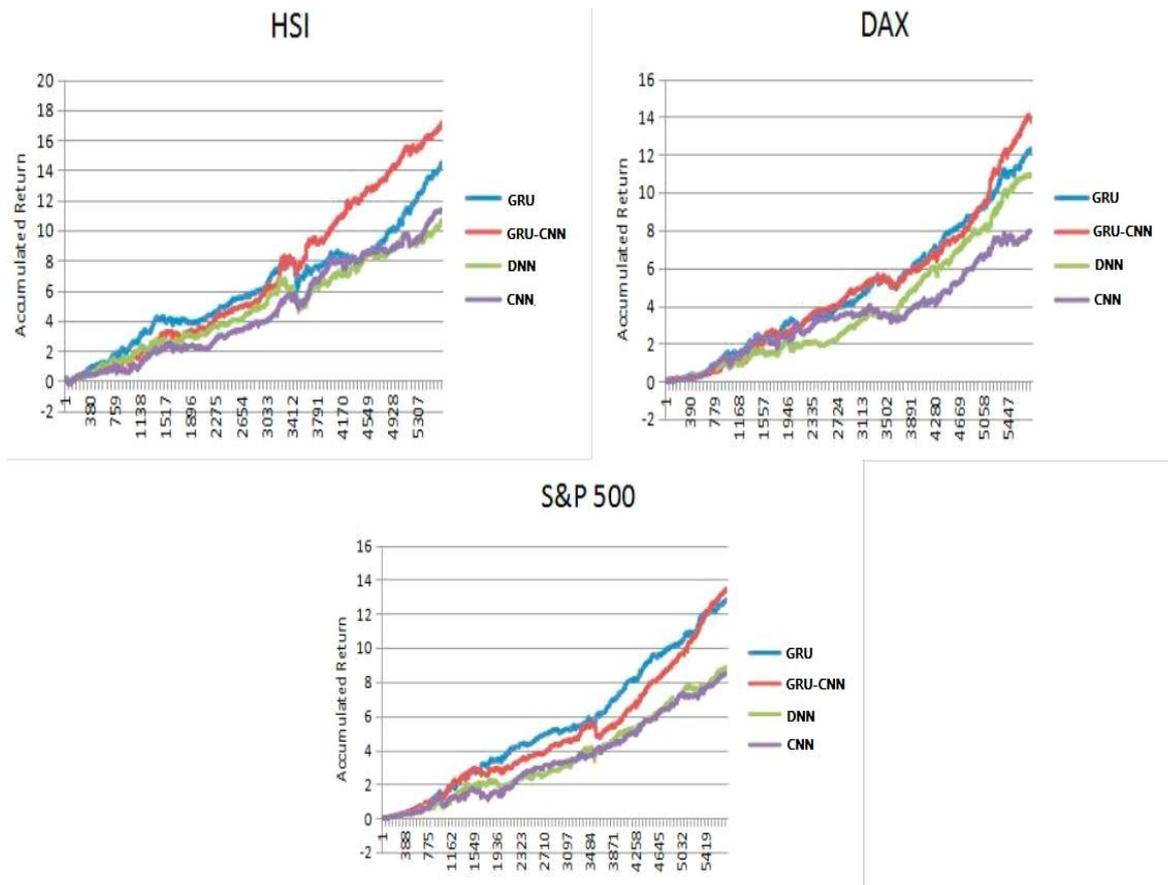


Fig.8. Accumulated normalized return in the trading periods of the HIS, the DAX and the S&P 500.

8. Conclusion

In this paper, we present combination based architecture of CNN with RNN for financial sequence predictions. We applied GRU network and replace its traditional output layer with CNN to predict the operation signal on the three stock indexes HIS, the DAX and the S&P 500, among May 1990 and August 2016. The proposed model is showed effectiveness and higher value through the experimental results detailed. Based on the proposed mechanism, the GRU-based architecture is used to capture meaningful information from a vast array of financial time-series data has been presented to be effectively, and for the final output that CNN outperform to usage fully connected layer. Our proposed model achieved superior performance in the term of accuracy, daily mean returns, and accumulative returns as compared to others traditional deep learning models namely GRU, DNN, and CNN. Furthermore, it will be remarkable to observe future work on implementing proposed model for further time series applications such as weather forecasting, earthquake prediction and signal processing.

Acknowledgments

The authors would like to thank the Ministry of Higher Education Malaysia (MOHE) and Universiti Tun Hussein Onn Malaysia for funding this research activity under the Fundamental Research Grant Scheme (FRGS/1/2017/ICT02/UTHM/02/5), vote no. 1641.

References

- [1] S. Chambon *et al.*, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, 2018.
- [2] P. Zhou *et al.*, "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap.)*, pp. 207–212, 2016.
- [3] A. M. El-Masry, M. F. Ghaly, M. A. Khalafallah, and Y. A. El-Fayed, "Deep Learning for Event-Driven Stock Prediction," *J. Sci. Ind. Res. (India)*, vol. 61, no. 9, pp. 719–725, 2017.
- [4] N. Huck, "Pairs selection and outranking: An application to the S&P 100 index," *Eur. J. Oper. Res.*, vol. 196, no. 2, pp. 819–

- 825, 2009.
- [5] M. Motwani and A. Tiwari, "A Novel Semi Supervised Algorithm for Text Classification Using BPNN by Active Search," *IJCSI Int. J. Comput. Sci. Issues*, vol. 11, no. 3, pp. 154–160, 2014.
 - [6] E. Guresen, G. Kayakutlu, and T. U. Daim, "Expert Systems with Applications Using artificial neural network models in stock market index prediction," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10389–10397, 2011.
 - [7] M. F. Dixon, D. Klabjan, and J. Bang, "Implementing Deep Neural Networks for Financial Market Prediction on the Intel Xeon Phi," *Ssrn*, 2015.
 - [8] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *2017 Int. Conf. Commun. Signal Process.*, pp. 0588–0592, 2017.
 - [9] G. Batres-estrada, "Deep Learning for Multivariate Financial Time Series," *DEGREE Proj. Math. Stat. , Second Lev. Stock. SWEDEN2015*, 2015.
 - [10] R. Collobert and J. Weston, "A unified architecture for natural language processing," *Proc. 25th Int. Conf. Mach. Learn. - ICML '08*, pp. 160–167, 2008.
 - [11] U. Akram, R. Ghazali, L. H. Ismail, and M. Zulqarnain, "An improved Pi-Sigma Neural Network with Error Feedback for Physical Time Series Prediction," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, pp. 1–7, 2019.
 - [12] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv : 1506 . 00019v4 [cs . LG] 17 Oct 2015," pp. 1–38, 2015.
 - [13] R. Ghazali, A. J. Hussain, P. Liatsis, and H. Tawfik, "The application of ridge polynomial neural network to multi-step ahead financial time series prediction," *Neural Comput. Appl.*, vol. 17, no. 3, pp. 311–323, 2008.
 - [14] S. Rönqvist and P. Sarlin, "Bank distress in the news : Describing events through deep learning," *Artif. Intell. Rev.*, vol. 1603, pp. 1–30, 2016.
 - [15] "Improving Decision Analytics With Deep Learning : The Case Of Financial Disclosures," *J. Mach. Learn. Res.*, vol. 74, pp. 1–10, 2016.
 - [16] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep Learning for Event-Driven Stock Prediction," *springer*, no. Ijcai, pp. 2327–2333, 2015.
 - [17] Z. Lanbouri and S. Achchab, "A new approach for Trading based on Long-Short Term memory Ensemble technique," *IJCSI Int. J. Comput. Sci. Issues*, vol. 16, no. 3, pp. 27–31, 2019.
 - [18] M. Dixon, D. Klabjan, and J. Hoon, "Classification-based financial markets prediction using deep neural networks," *J. Mach. Learn. Res.*, vol. 6, pp. 67–77, 2017.
 - [19] O. Hegazy, O. S. Soliman, and M. A. Salam, "A Machine Learning Model for Stock Market," *arXiv Prepr. arXiv1402.7351*, vol. 4, no. 12, pp. 17–23, 2014.
 - [20] J. Gamboa, "Deep Learning for Time-Series Analysis," *axiv*, vol. 69, pp. 1–13, 2017.
 - [21] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep Learning for Extreme Multi-label Text Classification," *Proc. 40th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '17*, pp. 115–124, 2017.
 - [22] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, "A comparative review on deep learning models for text classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 325–335, 2020.
 - [23] R. Mittelman, "Time-series modeling with undecimated fully convolutional neural networks," *axiv*, no. August, pp. 1–9, 2015.
 - [24] S. Lahmiri, "Wavelet low- and high-frequency components as features for predicting stock prices with backpropagation neural networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 26, no. 2, pp. 218–227, 2014.
 - [25] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv*, no. September, pp. 1–15, 2014.
 - [26] M. Zulqarnain, S. A. Ishak, R. Ghazali, and N. M. Nawati, "An Improved Deep Learning Approach based on Variant Two-State Gated Recurrent Unit and Word Embeddings for Sentiment Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 594–603, 2020.
 - [27] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.

Authors' Profiles



Muhammad Zulqarnain received his Bachelor and Master degree in Computer Science & Information Technology from The Islamia University of Bahawalpur (IUB), Pakistan. He received his M.Phil degree (Master of Philosophy) from National College of Business Administration & Economics, Lahore, Pakistan. He is currently pursuing Ph.D from University Tun Hussein Onn Malaysia. His research interest is Machine Learning and Deep learning for natural language processing and its application.



Rozaida Ghazali is currently a Professor at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM). She graduated with a Ph.D. degree in Higher Order Neural Networks from the School of Computing and Mathematical Sciences at Liverpool John Moores University, United Kingdom in 2007. Earlier, in 2003 she completed her M.Sc. degree in Computer Science from Universiti Teknologi Malaysia (UTM). She received her B.Sc. (Hons) degree in Computer Science from Universiti Sains Malaysia (USM) in 1997. In 2001, Rozaida joined the academic staff in UTHM. Her research area includes neural networks, swarm intelligence, optimization, data mining, and time series prediction. She has successfully supervised a number of PhD and master students and published more than 100 articles in various international journals and

conference proceedings. She acts as a reviewer for various journals and conferences, and as an editor in a few Springer conference proceedings. She has also served as a conference chair, and as a technical committee for numerous international conferences.



Muhammad Ghulam Ghouse passed his Bachelor Degree in Computer Science from Dada Bhoj Institute of Higher Education, Karachi, Pakistan specialized in Programming, Networking and Developing. He has received his Master Degree in Information Technology from Hamdard University, Karachi, Pakistan specialized in Networking, Cryptography and Cryptanalysis. He has currently been pursuing Ph. D from University Tun Hussein Onn Malaysia (UTHM). His research of Interest of Hybrid Cryptography by using DNA Steganography and its applications and simulations.



Yana Mazwin Mohmad Hassim is a senior lecturer at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM). She graduated with a PhD degree from Universiti Tun Hussein Onn Malaysia (UTHM) in 2016. Earlier, in 2006 she completed her Master's degree in Computer Science from Universiti of Malaya (UM). She received her Bachelor of Information Technology (Hons) degree majoring in Industrial Computing from Universiti Kebangsaan Malaysia (UKM) in 2001. In 2003, Yana Mazwin joined the academic staff in UTHM. Her research area includes neural networks, swarm intelligence, optimization and classification.



Irfan Javid is lecturer at Department of Computer Science and information Technology, University of Poonch, Rawalakot AJK, Pakistan. He obtained his Master degree in Software Engineering from Bahria University, Islamabad, Pakistan. Currently, he is pursuing PhD from University Tun Hussein Onn Malaysia. His research interests are in the field of data science, Deep learning and computer programming.

How to cite this paper: Muhammad Zulqarnain, Rozaida Ghazali, Muhammad Ghulam Ghouse, Yana Mazwin Mohmad Hassim, Irfan Javid, "Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks ", International Journal of Intelligent Systems and Applications(IJISA), Vol.12, No.6, pp.21-32, 2020. DOI: 10.5815/ijisa.2020.06.02