IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Cryptocurrency Prediction Using Several Machine Learning Techniques

Dr. K.N.S. LAKSHMI [*1], ANISHA NALLAMILLI [*2]

[*1]Professor, Department of Computer Science & Engineering,

*[2]M.Tech Scholar, Department of Computer Science & Engineering, Sanketika Vidhya Parishad Engineering College, P.M. Palem, Visakhapatnam, Andhra Pradesh, 530041

mnslakshmi.vvit@gmail.com[#1], anishan.402@gmail.com[#2]

ABSTRACT

Cryptocurrency is a new sort of asset that has emerged as a result of the advancement of financial technology. Around the world, there are hundreds of cryptocurrencies that are used. This project proposes three types of recurrent neural network (RNN) algorithms used to predict the prices of three types of cryptocurrencies, namely Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH) . The models show excellent predictions depending on the mean absolute percentage error (MAPE). Results obtained from these models show that the gated recurrent unit (GRU) performed better in prediction for all types of cryptocurrency than the long short-term memory (LSTM) and bidirectional LSTM (bi-LSTM) models. Therefore, it can be considered that the best algorithm GRU presents the most accurate prediction for LTC with MAPE percentages of 0.2454%, 0.8267%, and 0.2116% for BTC, ETH, and LTC, respectively. Overall, the prediction models in this paper represent accurate results close to the actual prices of cryptocurrencies.

KEY WORDS:

Cryptocurrency, Litecoin, Bitcoin, Ethereum, Recurrent Neural Network, Gated Recurrent Unit.

1. INTRODUCTION

In 2017, the market capitalization of cryptocurrencies grew exponentially for several months in a row, which greatly increased their appeal. In January 2018, the prices reached their high at almost $800 billion. Although a variety of time series models have been effective in using machine learning to forecast stock market values, their use in predicting cryptocurrency prices has been fairly limited. This makes sense given that a variety of factors, like technical advancement, internal competitiveness, market pressure to provide, economic difficulties, security concerns, political considerations, etc., affect cryptocurrency pricing. If clever invention tactics are used, their extreme volatility offers a significant possibility for large profit. Sadly, cryptocurrencies are less predictable than more established financial forecasts like stock market forecasts since they lack indexes. Here, we&#39 , we will talk about the four-step method for predicting bitcoin values.:

1. Obtaining current cryptocurrency information.
2. Gather data for testing and training.
3. Use an LSTM neural network to predict the price of cryptocurrency.
4. Display the outcomes of the forecast.

It takes a lot of effort for the end user to acquire and maintain all the information in the current system since we attempt to use a manual technique to identify bit coin or cryptocurrency price forecast based on the analytical values gathered from historical data. This inspired me to create an application that would reliably forecast the price of a cryptocurrency or bit coin using a set of pre-defined CNN or ML algorithms, after which I would check the accuracy of each algorithm. The three recurrent neural network (RNN) algorithms proposed in this research are used to forecast the values of three different cryptocurrencies: Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH). Depending on the mean absolute percentage error, the models produce accurate predictions (MAPE). The results of these models demonstrate that the gated recurrent unit (GRU) outperformed the long short-term memory (LSTM) and bidirectional LSTM (bi-LSTM) models in terms of prediction for all types of cryptocurrencies. With MAPE percentages of 0.2454%, 0.8267%, and 0.2116% for BTC, ETH, and LTC, respectively, it can be said that the best algorithm GRU gives the most accurate forecast for LTC..

These models are crucial because they may have a big impact on the economy by assisting traders and investors in determining when to buy and sell cryptocurrencies. These encouraging findings provide credence to the idea that machine learning offers reliable methods for examining the predictability of cryptocurrencies and creating effective trading plans in these markets, even under challenging market situations.

2. LITERATURE SURVEY

In this section we will mainly discuss about the background work that is carried out in order to prove the performance of our proposed Method. Literature survey is the most important step in software development process. For any software or application development, this step plays a very crucial role by determining the several factors like time, money, effort, lines of code and company strength. Once all these several factors are satisfied, then we need to determine which operating system and language used for developing the application. Once the programmers start building the application, they will first observe what are the pre-defined inventions that are done on same concept and then they will try to design the task in some innovated manner.

MOTIVATION

[1] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," ***Energy*****, vol. 158, pp. 774–781, 2018.**

For all market players, anticipating short-term power load is one of the most crucial issues. Natural and societal variables have an impact on the short-term power load, which makes load forecasting more challenging. A new hybrid model based on fruit fly optimization algorithm (FOA)-optimized wavelet neural network (WNN), autoregressive integrated moving average (ARIMA), and improved empirical mode decomposition (IEMD) is suggested and compared to previous models in order to increase forecasting accuracy. Simulation findings show that the proposed model outperforms previous comparison models in terms of power load predictions.

[2] N. Ghadimi, A. Akbarimajd, H. Shayeghi, and O. Abedinia, "Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting," ***Energy*****, vol. 161, pp. 130–142, 2018.**

In light of how the electrical market has been reformed, short-term load forecasting is of great relevance. For a power system to operate efficiently, accurate load forecasting is crucial. However, electrical load is non-linear and very volatile. Such complicated signals demand the use of appropriate prediction techniques. In this research, a hybrid prediction approach that uses a cutting-edge feature selection method and a sophisticated forecast engine built on a new intelligent algorithm is proposed. To choose suitable candidates as input for the forecast engine, the power load signal is first filtered using a feature selection approach. Then, a two-stage forecast engine based on ridgelet and Elman neural networks is put into practise. To increase the precision and capabilities of the forecast engine, all of its parameters are selected using a cutting-edge intelligent algorithm. In order to compare the suggested technique with numerous existing algorithms, several electrical markets were taken into consideration as test cases. The forecasting methodology that is being suggested also assesses the absolute forecasting errors in this study (among seven types of measurements i.e., absolute forecasting errors, measures based on percentage errors, symmetric errors, measures based on relative errors, scaled errors, relative measures and other error measures). The outcomes demonstrate that the suggested technique is successful.

[3] A. Goia, C. May, and G. Fusai, "Functional clustering and linear regression for peak load forecasting," ***International Journal of Forecasting*****, vol. 26, no. 4, pp. 700–711, 2010.View at:**

In this study, we examine the challenge of district heating system short-term peak load forecasting utilising historical heating demand data. Our data collection is divided into four distinct periods, each of which has 198 days and 24 hourly observations. Within each period, we may identify both an intra-daily seasonality and a seasonality impact. We make use of the data set's functional character and provide a forecasting strategy based on functional statistics. In specifically, we categorise the daily load curves using a functional clustering technique. We then design a family of functional linear regression models based on the groupings we've created.

We use a functional discriminant analysis to assign new load curves to clusters in order to make predictions. Lastly, we assess performance of the proposed approach in comparison with some classical models

[4] A. H. Nury, K. Hasan, and M. J. B. Alam, "Comparative study of wavelet-ARIMA and wavelet-ANN models for temperature time series data in northeastern Bangladesh," *Journal of King Saud University-Science*, vol. 29, no. 1, pp. 47–61, 2017.

The study of temperature variance and the forecasting of temperature change both benefit from time-series studies of temperature data. With the exception of the maximum temperature at Sreemangal, Mann-Kendall (M-K) analyses of temperature time-series data in northeastern Bangladesh revealed increasing trends (Sen's slope of maximum and minimum yearly temperatures at Sylhet of 0.03 °C and 0.026 °C, respectively, and a minimum temperature at Sreemangal of 0.024 °C). The linear trends at the Sylhet and Sreemangal stations indicated that the maximum temperature is rising by 2.97 °C and 0.59 °C per hundred years, while the minimum temperature is decreasing by 2.17 °C and 2.73 °C per hundred years, showing that climate change is impacting temperature in this area. The wavelet approach, an autoregressive integrated moving average (ARIMA) model, and an artificial neural network (ANN) used on monthly maximum and minimum temperature data in this research give an alternate method for temperature prediction. A training dataset (1957–2000) is used to build the models, and a testing dataset (2001–2012) is used to determine how well they function. The statistical performance of the models' calibration and validation is measured, and their relative performance based on the forecasting accuracy of out-of-sample forecasts is evaluated. The outcomes show that the wavelet-ARIMA model is superior to the wavelet-ANN model in terms of effectiveness.

3. EXISTING METHODOLOGY

In the existing system there are lot of machine learning approaches implemented for predicting the price of crypto currency and all the primitive algorithms failed to achieve more accuracy and it is not efficient to identify the price prediction of several currencies at once. Hence there is no accurate model which can predict all these things accurately in the existing system.

LIMITATION OF EXISTING SYSTEM

1. The classification produce the less accuracy when compared to Regression methods in scenarios like bit coin or crypto currency price prediction.
2. Comparatively with other algorithms, ML algorithms are very less accurate in finding the price prediction.
3. This is accurate if we use for less dimensions
4. This is not accurate for large dimensional dataset.

4. PROPOSED SYSTEM & ITS ADVANTAGES

This project proposes three types of recurrent neural network (RNN) algorithms used to predict the prices of three types of cryptocurrencies, namely Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH) . The models show excellent predictions depending on the mean absolute percentage error (MAPE). Results obtained from these models show that the gated recurrent unit (GRU) performed better in prediction for all types of cryptocurrency than the long short-term memory (LSTM) and bidirectional LSTM (bi-LSTM) models. Therefore, it can be considered that the best algorithm GRU presents the most accurate prediction for LTC with MAPE percentages of 0.2454%, 0.8267%, and 0.2116% for BTC, ETH, and LTC, respectively

.

ADVANTAGES OF PROPOSED SYSTEM:

The following are the benefits of the proposed system. They are:

1. The results obtained by the GRU is best compared to any other Models.
2. The Accuracy obtained was almost equal to 99 percent which proves using of GRU gives best results.
3. This is accurate for any kind of cryptocurrency to find out the price prediction.

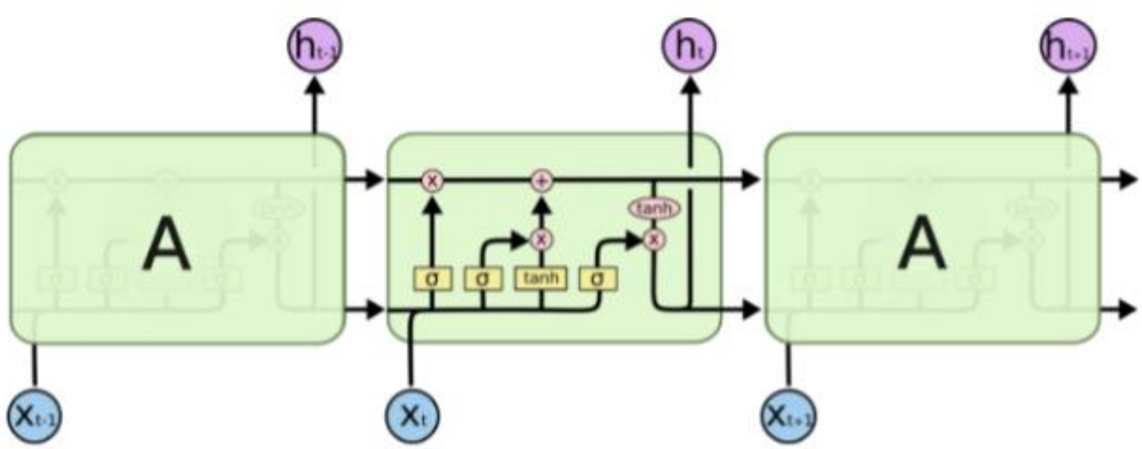
5. PROPOSED ALGORITHMS

The following are the set of models or algorithms which are deployed in this current application. They are as follows:

LSTM

Each LSTM layer may access data from both the layer below it and the one above it using specific gates. The data is sent through the LSTM cells after passing through a number of gates (such as the forget gate, input gate, etc.) and different activation functions (such as the tanh function, relu function). The key benefit of this is that it gives each LSTM cell the ability to retain patterns for a limited period of time. It should be highlighted that the LSTM may recall crucial information while simultaneously forgetting unimportant information. Below are the

LSTM architecture



LSTM architecture

Lets now construct the model. All of the layers are stacked using a sequential model (input, hidden and output). A 20% Dropout layer, a Dense layer with a linear activation function, and an LSTM layer make up the neural network. Adam served as the optimizer in the model I created, while Mean Squared Error served as the loss function.. Mean Absolute Error Without taking into account their direction, it calculates the average size of the mistakes in a series of forecasts. Absolute differences between actual and anticipated observations are averaged over the test sample with each individual difference given the same weight.

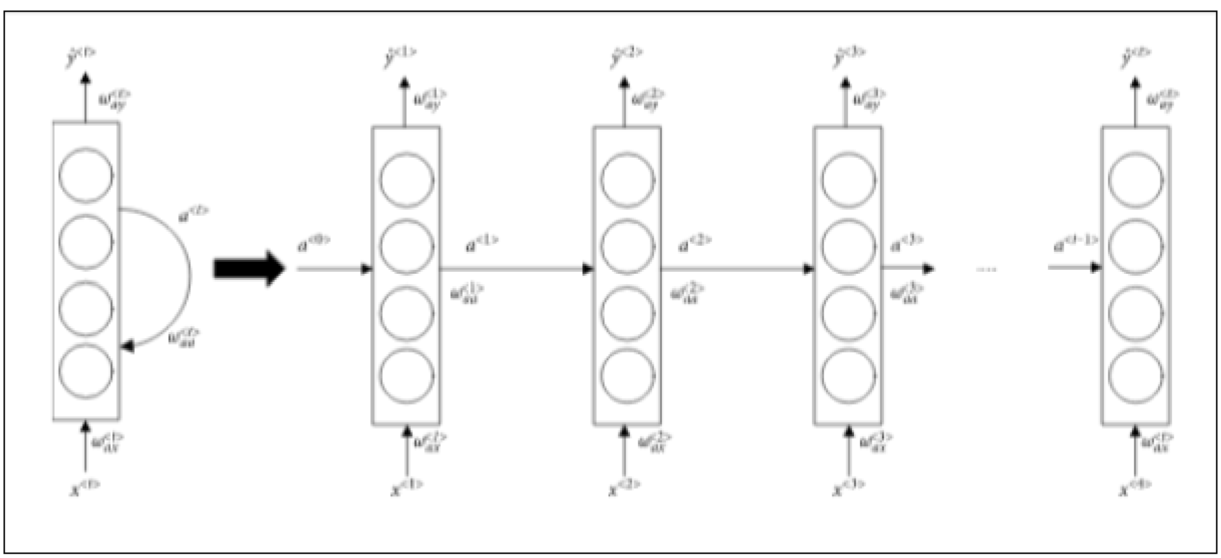
$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

The computed MAE value appears to be good. Let's use the code below to plot the actual and forecasted prices.:

The best method that can be utilised in this situation to forecast Bitcoin cryptocurrency values is machine learning. For the model to generate a forecast that was close to correct, it had to accomplish a number of objectives. This involved choosing a framework that could provide accurate predictions, take other factors into account while making predictions, and be trainable. The author evaluated a number of frameworks while keeping these objectives in mind and ultimately settled on utilising Keras, a neural networks API built on top of Tensorflow. Keras and Support Vector Machine were contrasted. In terms of computation and performance, Keras is less costly than SVM. When we talk about artificial intelligence frameworks, depending on how we would want the learning to be, we can decide over the frameworks that can be used. Following that, it was necessary to choose which layers to include, how many would be needed, and the epoch rates. Since many activation functions are better suited by standardizing and transforming the dataset used for training.

GRU MODEL

Hybrid neural networks of GRU and CNN. The GRU module and the CNN module make up the framework of the proposed GRU-CNN hybrid neural networks. The power system's time sequence data and spatiotemporal matrices are the inputs, and the outputs are predictions of the load value for the future..



With two gates (update gate and reset gate) as opposed to three (forgetting gate, input gate, and output gate) in LSTM, GRU is an LSTM variant with a gated recurrent neural network structure. Because GRU has fewer training parameters than LSTM, it converges more quickly than LSTM during training [34]. Figure 2 depicts the GRU structure, with as the current unit's input and and tanh as its activation functions. It connects to the input of the following unit and is also the output of the prior unit. are the unit's inputs, are the unit's outputs, are represented by the reset gate and update gate, respectively, and the candidate activation is calculated in a manner akin to that of a conventional recurrent unit. In GRU, there are two gates: the update gate preserves historical information to the present state.

6. IMPLEMENTATION PHASE

The step of implementation is when the theoretical design is translated into a programmatically-based approach. The application will be divided into a number of components at this point and then programmed for deployment. The front end of the application takes Google Collaboratory and as a Back-End Data base we took UCI Heart Patients Records as dataset. Python is being used in this instance to implement the present application. The following five modules make up the bulk of the application. They are listed below:

1. Load Dataset Module
2. Generate Test and Train Data
3. Run Pre-Trained CNN Models
4. Predict the Cryptocurrency Price from Test Dataset
5. Comparative Analysis of Algorithms Accuracy

1) LOAD DATASET MODULE:

We attempt to load the dataset from the CoinDesk website in this module, and then we attempt to provide the data from that excel file as input to the next module.. Dataset URL: BTC_USD_2021-02-12_2022-02-11-CoinDesk.csv

2) GENERATE TEST & TRAIN MODULE

Here, we attempt to split the data into test and training datasets, and we employed a 70:30 ratio to break the large dataset up into smaller chunks. In this case, 70% of the data records are used to train the system, while 30% of the data are utilised to test the model.

3) RUN SEVERAL CNN MODELS MODULE

In this instance, we attempt to run several CNN models on the train dataset and verify the likelihood of each and every attribute contained in that record. After all the records have been analysed, we now try to identify which records the cryptocurrency's price has changed on. Here, we employ LSTM, Bi-LSTM, and GRU Models on test data to attempt to forecast the price of cryptocurrencies.

4) DETECT PRICE PREDICTION MODULE

After all the records have been analysed, we now try to identify which records the cryptocurrency's price has changed on. Here, we employ LSTM, Bi-LSTM, and GRU Models on test data to attempt to forecast the price of cryptocurrencies.

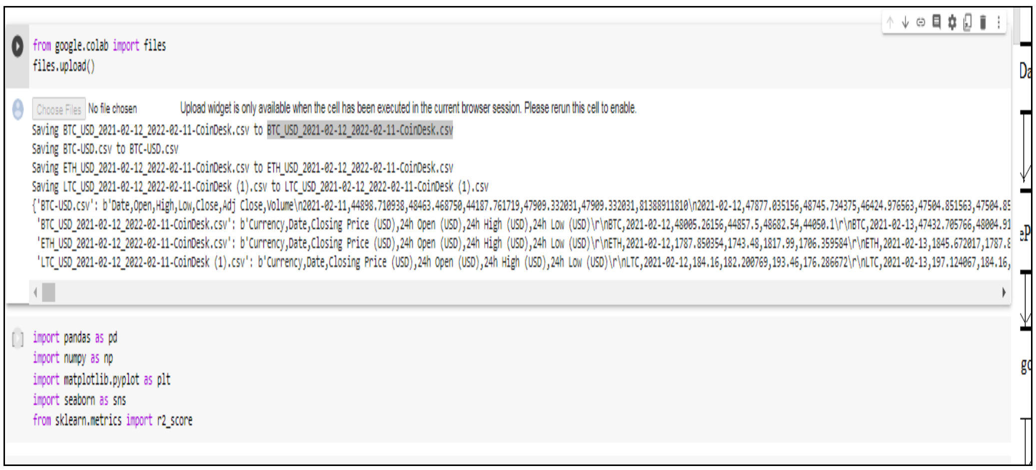
5) COMPARATIVE ANALYSIS OF ALGORITHMS ACCURACY

We evaluated the dataset using LSTM, Bi-LSTM, and GRU Models on test data in this application to attempt and forecast the price of cryptocurrencies. The final study states unequivocally that GRU provides the highest accuracy when compared to all other models.

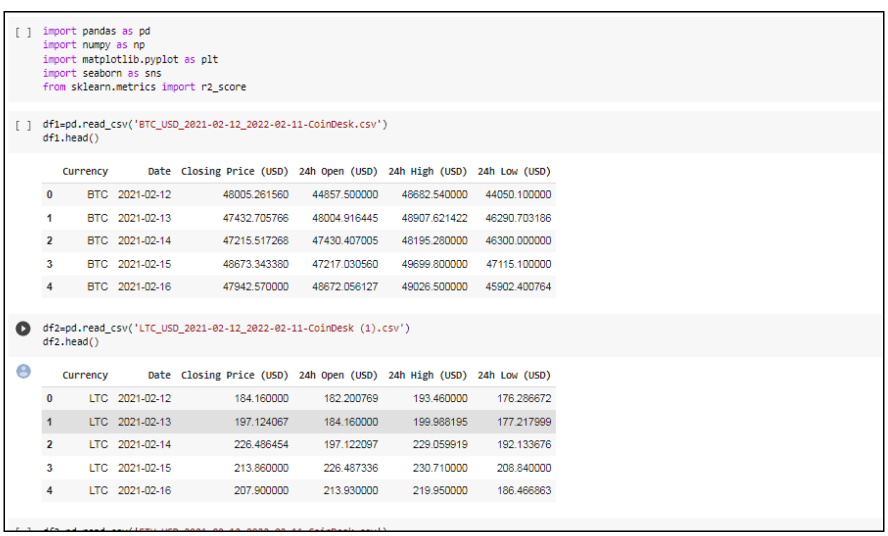
7. EXPERIMETAL REPORTS

In this proposed application, we try to use google collab as working platform and try to show the performance of our proposed application.

1) LOAD DATASET FROM WEBSITE



2) IMPORT LIBRARIES AND VIEW FILE CONTENT



3) TEST AND TRAIN THE DATA

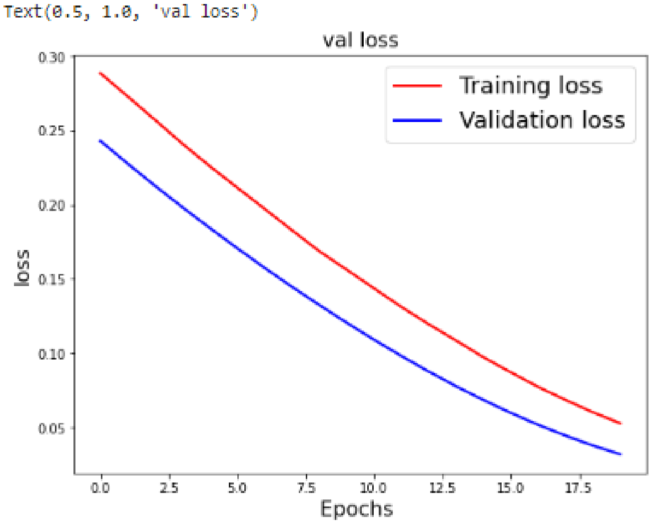
```
[ ] X_train,y_train=split_data(train)
    X_test,y_test=split_data(test)
    print(f'length of X_train is {len(X_train)}, y_train is {len(y_train)}')
    print(f'length of X_test is {len(X_test)}, y_test is {len(y_test)}')

    length of X_train is 90, y_train is 90
    length of X_test is 30, y_test is 30

    X_train=np.array(X_train)
    X_test=np.array(X_test)
    y_train=np.array(y_train)
    y_test=np.array(y_test)
    y_train=y_train.reshape(-1,1)
    y_test=y_test.reshape(-1,1)
    print(f'length of X_train is {(X_train.shape)}, y_train is {(y_train.shape)}')
    print(f'length of X_test is {(X_test.shape)}, y_test is {(y_test.shape)}')

    length of X_train is (90, 2, 1), y_train is (90, 1)
    length of X_test is (30, 2, 1), y_test is (30, 1)

[ ] from tensorflow.keras.layers import Dense,Input,LSTM,TimeDistributed,Bidirectional
    from tensorflow.keras.models import Sequential


    bi_lstm=Sequential()
    bi_lstm.add(Input(shape=(2,1)))
    bi_lstm.add(Bidirectional(LSTM(units=20,activation='relu')))
    # lstm.add(LSTM(units=50,return_sequences=True))
    bi_lstm.add((Dense(1)))

    bi_lstm.compile(loss='mean_squared_error',optimizer='adam')
```

4) APPLY CNN MODELS

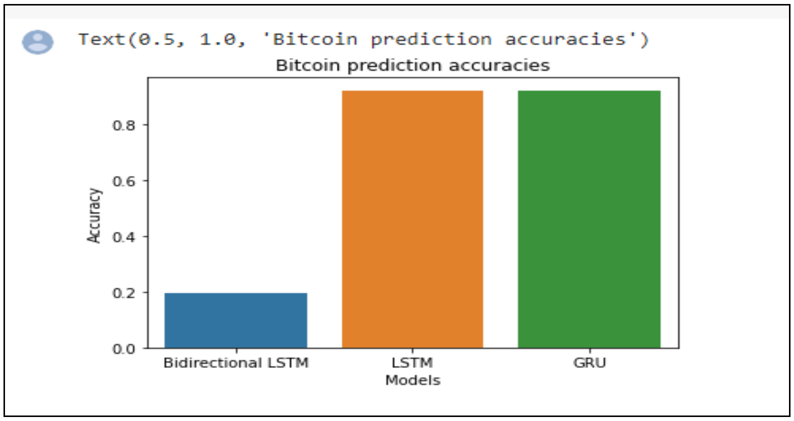
```
from tensorflow.keras.layers import Dense,Input,LSTM,TimeDistributed,Bidirectional
from tensorflow.keras.models import Sequential


bi_lstm=Sequential()
bi_lstm.add(Input(shape=(2,1)))
bi_lstm.add(Bidirectional(LSTM(units=20,activation='relu')))
# lstm.add(LSTM(units=50,return_sequences=True))
bi_lstm.add((Dense(1)))

bi_lstm.compile(loss='mean_squared_error',optimizer='adam')

bi_lstm.summary()

WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel as fallback when running on GPU.
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 40)               3520
 l)

 dense (Dense)               (None, 1)                 41

=================================================================
Total params: 3,561
Trainable params: 3,561
Non-trainable params: 0
_________________________________________________________________
```

5) VALIDATION GRAPH

```
[ ] Text(0.5, 1.0, 'val loss')
```



6) PERFORMANCE OF ALL THE 3 ALGORITHMS



8. CONCLUSION

The GRU-CNN hybrid neural network, which is based on the GRU model and the CNN model, as introduced in this research with the goal of increasing the accuracy of STLF. The CNN module of the GRU-CNN model excels in processing spatiotemporal matrices, whereas the GRU module of the model is dedicated to processing time sequence data. By identifying characteristics from changeable data that affects the power system, the suggested model can anticipate electrical demand rapidly and precisely. . The suggested GRU-CNN model was contrasted with the BPNN, GRU, and CNN models in the real-world tests of the Wuwei region electrical load forecasting. The outcomes demonstrate that the proposed GRU-CNN model can efficiently extract the hidden features of the datasets from both time sequence data and spatiotemporal matrix data. The suggested GRU-CNN model obtains the greatest performance across the four models, as shown by the GRU-CNN model's lowest MAPE and RMSE values

9. REFERENCES

1) "A review and study of regression and machine learning models for commercial building electrical load forecasting," Renewable and Sustainable Energy Reviews, vol. 73, pp. 1104–1122, 2017. B. Yildiz, J. I. Bilbao, and A. B. Sproul.
2) "Research and use of a novel hybrid model based on a deep neural network for power load forecasting: a case study in Australia," Energies, vol. 12, no. 13, p. 2467, 2019. 3. K. Ni, J. Wang, G. Tang, and D. Wei.
3) "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841-851, 2019. W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang.
4) viewL. Yang and H. Yang, “Analysis of different neural networks and a new architecture for short-term load forecasting,” *Energies*, vol. 12, no. 8, 2019.
5) C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” *Energies*, vol. 11, no. 12, 2018.
6) J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, “Short term electricity load forecasting using a hybrid model,” *Energy*, vol. 158, pp. 774–781, 2018.
7) N. Ghadimi, A. Akbarimajd, H. Shayeghi, and O. Abedinia, “Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting,” *Energy*, vol. 161, pp. 130–142, 2018.
8) A. Goia, C. May, and G. Fusai, “Functional clustering and linear regression for peak load forecasting,” *International Journal of Forecasting*, vol. 26, no. 4, pp. 700–711, 2010.

9) A. H. Nury, K. Hasan, and M. J. B. Alam, “Comparative study of wavelet-ARIMA and wavelet-ANN models for temperature time series data in northeastern Bangladesh,” *Journal of King Saud University-Science*, vol. 29, no. 1, pp. 47–61, 2017.
10) C.-M. Lee and C.-N. Ko, “Short-term load forecasting using lifting scheme and ARIMA models,” *Expert Systems With Applications*, vol. 38, no. 5, pp. 5902–5911, 2011.
11) G.-YInformation Sciences, vol. 438, pp. 46-57, 2018. Chen, M. Gan, and G.-L. Chen, "Generalized exponential autoregressive models for nonlinear time series: stationarity, estimation, and applications."
12) 15. "Artificial neural network for short-term demand forecasting in distribution systems," Energies, vol. 7, no. 3, pp. 1576-1598, 2014. L. Hernández, C. Baladrón, J. Aguiar, et al.
13) 16. J.-p. Liu and C.-l. Li, "The short-term power load forecasting based on wavelet least square support vector machine with DWT-IR for feature selection," Sustainability, vol. 9, no. 7, 2017.

10. About the Authors

Dr.K.N.S LAKSHMI is currently working as a Professor in Department of Computer Science and Engineering at Sanketika Vidhya Parishad Engineering College, P.M. Palem, Visakhapatnam, Andhra Pradesh. She has more than 16 years of teaching experience. Her research interest includes Machine Learning, Adhoc Networks, Network Security, Python.

ANISHA NALLAMILLI is currently pursuing her 2 years M.Tech course in Department of Computer Science and Engineering at Sanketika Vidhya Parishad Engineering College, P.M. Palem, Visakhapatnam, Andhra Pradesh. Her area of interest includes Python, Java and Machine Learning.