Height-end Multi-layer Perceptron and Machine Learning Methods of Forecasting Bitcoin Price Time Series

Stanley Ziweritin*[1], Emmanuel Nwaeze[2], Joshua Oluwasegun Jatto[3]

***Abstract*—Bitcoin trading is now the center of attraction and people are investing heavily in it. Rapidly fluctuating prices have made few investors' millionaires and others crushing and severe losses. Thousands of investors are now very broke as Bitcoin drains their wallets. It has been nightmares for several investors who have seen their money disappear before their eyes in hopes that the market will recover soon. Existing methods are inefficient for predicting future data in real time. Uncertain price movement can be predicted in advance to avoid future losses for potential investors with greater accuracy. This research focuses on creating high-end multi-layer perceptron with other machine learning techniques to predict Bitcoin price time series. LOESS smoothing technique was applied to clearly show features of underlying data patterns and trends before model fitting using an average rolling mean of 3. The adma's optimization technique was employed with Relu and sigmoid activation functions to update weights of the MLP neural network. It combined the best features of AdaGrad and RMSProp; which improves the learning rate. We adjusted the input weights with output errors to help compute errors in the previous layer. And created KNN and SVM classifiers with pipeline concept for better performance; validated and compared the performance on different fine-tuned hyper-parameter values. The SVM produced 93% accuracy, KNN gave 90%, and MLP recorded an accuracy rate of 96% as the best.**

Index Terms—Forecast, K-nearest neighbor, multi-layer perceptron, support vector machine

I. INTRODUCTION

The Multi-layer perceptron is one of the best-known machine learning techniques used by professionals to solve classification and regression tasks[1],[2],[3],[4]. The deep MLP neural network can also be used to predict future Bitcoin price patterns using time series data [5],[6],[7],[8],[9]. The virtual currency called bitcoin is gradually gaining popularity and everyone including: rich and poor are tweeting and investing a lot due to the many future benefits and it could probably become our future currency[10],[11],[12]. The digital asset can still help investors make a lot of money to build monumental wealth in the crypto-currency market. This is still possible provided you have some prior knowledge of future fluctuating patterns and know when to buy and sell and have cash to afford the risk[12],[13],[14],[15]. Virtual currency is a file stored using an application called a digital wallet on a smart phone or computer system and each transaction made is stored in a list called blockchain[16],[17],[18]. Bitcoin miners receive their rewards through verified transactions added to the blockchain list[19]. It is a digital currency or cash that uses the peer-to-peer Internet to instantly confirm transactions between different users[20]. Digital money can be moved and converted into cash through the use of third-party exchange brokers, including ATMs at a specific market value or peer-to-peer when you sell your Bitcoin[21],[22],[23]. Bitcoin was created by Nakamoto[24] to be worth over $40 billion as it is traded daily. It offers low transaction fees compared to traditional payment systems controlled by decentralized authorities and not issued by individuals, banks or governments. It is one of the most important assets in speculation and investment. Rapidly fluctuating bitcoin prices have made few investors millionaires and inflicted severe and devastating losses on others. Thousands of investors are now very broke as Bitcoin drains their wallets due to fluctuating investment values; in the hope that the Bitcoin market will recover soon. It has been a nightmare for some investors who have seen their money disappear before their eyes. Furthermore, existing methods address the uncertainty and low accuracy applied to real-time data, caused by the problem of over-matching of models and underlying data. Price fluctuation periods can be predicted ahead of time to avoid losses for potential investors using Bitcoon price time series smoothed using moving average. We use our machine learning experience to build a model that can predict short, medium and long term returns using time series in the Bitcoin market, ranging from 2014 to 2022. This research focuses on creating high-end MLP and machine learning methods for predicting Bitcoin price time series. We intend to develop a model that can predict the daily price movement of Bitcoin with greater accuracy using price and payment networks over eight(8) years ranging from 2014 to 2022. The K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are the selected machine learning algorithms used to predict frequent and uncertain movements in cryptocurrency investment prices using seasonal data or time series. Python Spyder IDE is used

- *S. Ziweritin is with Akanu Ibiam Federal Polytechnic, Unwna-Afikpo, Ebonyi State, Nigerian E-mail: stanlo4godsluv@yahoo.com*
- *E. Nwaeze is with Alex Ekwueme Federal University Ndufu-Alike, Ebonyi State, Nigerian. E-mail: nwaezeema@yahoo.com*
- *J. O. Jatto is with Our Kings International School, Port Harcourt, Rivers State, Nigeria. E-mail: jattohjt@gmail.com*

to build, train, test and forecast the fluctuating data patterns of investment values and overcome the identified problem.

This paper is divided into different sections as follows: Section 1 provides the introduction, Section 2 provides a brief overview of previous approaches to the field of study and research gaps of the proposed model; Section 3 presents the materials and methods used to develop the model; Section 4 focuses on the results and a detailed discussion of the results; Section 5 presents the conclusion of the article.

II. RELATED WORKS

Saad[25] developed a support vector regression model with radial kernel functions to estimate the prices of XRP and Ethereum crypto-currencies. RBF's Gamma hyper-parameter and polynomial kernel values have been set to 0.0001 and 0.00005, respectively. The R-squared(R^2) of RBF provided 78% higher performance than linear and polynomial kernel functions which resulted in 48% and 43% in exceeding the variance of the dataset, respectively. The decision limit was affected for smaller Gamma (γ) hyper-parameter values and the model was heavily penalized for errors in misclassification of the data. Agarwal and Bisht [26]; proposed the use of machine learning and data mining techniques to predict the prices of market capitalization bitcoins with different functions and attributes. The linear regression technique has been adopted, trained and tested to efficiently calculate the intercept and variance in predicting trends and future values. The model produced prediction accuracy measures of 40%; it was bad and below average. Kenney[27] combined several machine learning techniques to create and predict chaotic Bitcoin price behaviors over time in the short, medium, and long term. The projected floating prices in the short term were correct, but ineffective with long term prices. The accuracy of the prediction was below requirements. Jen-Peng and Genesis[28] combined SVM, RF and deep learning techniques to estimate the return of Bitcoin's price trend. The prediction accuracy of the proposed techniques varies from 60% to 70% with polynomial, radial, Epanecknikov, multi-quadratic and point kernel functions. Patil[29] adopted logistic linear regression (LR) in machine learning and artificial neural network(ANN) techniques capable of predicting annual Bitcoin investment prices. The LR achieved 99.87% accuracy in the training phase with an error rate of 0.08 compared to ANN, which recorded 99.97% as the best at predicting Bitcoin time series data. Wu[30] Combined CNN, KNN, and LR to predict Bitcoin's investment values based on inherent relationships. CNN achieved an accuracy rate of 95% with a better multiple correlation coefficient of the R^2 value in predicting Bitcoin price movements over time, with training data being the best among LR and ANN techniques. CNN's R^2 value was also better Sriramya & Reddy[31]; developed a hybrid model consisting of linear regression(LR), K-means, SVM, Naive Bayes, CNN and RNN techniques to predict patterns of future Bitcoin daily prices. The LASSO outperformed the other algorithms followed by LR and NB requires longer training times to predict future models due to the limited number of training and test datasets and required a large dataset to improve accuracy of the forecast. Alahmari[32], developed a series of nonlinear ML techniques consisting of decision tree, KNN regression and LR to predict three different types of future cryptocurrency price models (XRP, Bitcoin and Ethereum) using time series data. The bivariate time series method was used using the daily closing prices of cryptocurrencies as a continuous variable and Morgan Stanley Capital International(MSCI) for all world country indices as the predictive variable. The decision tree regression technique outperformed LR in terms of accuracy and MAE, MSE, RMSE, and r-squared value. The model could not learn to extract non-economic features in the training and testing phases. Jaquart *et al.*,[33] proposed a machine learning model capable of predicting Bitcoin prices in the short term using CNN, RF and gradient boost classifications with time series data. The proposed model was trained to learn and predict the feature set of the target variable, resulting in negative returns on low-precision trades. Mehta & Sasikala[34] used RNN and RF techniques to predict Bitcoin's short-term future investment prices using historical data from 2015 to 2018. The prediction accuracy of RNN yielded 92% more and an error rate of 0.098, which performed better than the RF gave 84% with an error rate of 0.98 after training and testing. Aygün and Kabakçı[35] combined ARIMA, RNN, and CNN ML techniques to predict the prices of Bitcoin's using a univariate time series. The window size of CNN and RNN was set to 32 with the concept of trial and error method. RELU and sigmoid activation functions were used in the hidden layers. The RNN outperformed ARIMA's statistical and CNN techniques in predicting bitcoin prices, but did not perform well with the daily opening prices, average prices, market capitalization and volume features included in the multivariate time series. Pabuccu *et al.*,[36] proposed ANN, Naive Bayes, RF, LR and SVM classifiers to predict future fluctuating Bitcoin price trends. Sigmoid, tangent, and logistic transfer functions were used at the hidden and output layers of the ANN with multiple configurations to determine the best values of hyper-parameters or network settings related to the learning rate, number of iterations or epochs, hidden layer neurons and momentum. The ANN performed best and SVM provided the lowest metric with the discrete dataset, while the RF results produced the highest prediction accuracy and NB the lowest with the continuous dataset. Jane *et al.*,[37] proposed a one-day numerical solution using MLPNN, RF, ANFIS and support vector regression to predict the prices of bitcoin features using a surrogate time series (SGTS). The wavelength transform function was adopted to decompose the original time series into linear and non-linear components with projections representing feature prices. The methods of deferential evolution and meta-heuristic optimization have been adopted to obtain a more accurate and reliable result. MLPs performed better than other techniques, but did not work well with multivariate time series data. Dutta *et al.*,[38] proposed Gated recurrent enclosed unit and LSTM neural network models to predict the investment values of Bitcoin feature prices which outperformed other ML techniques. A single layer GRU with recurring dropout was introduced and trained to predict short and long term price trends based on a collective list of Bitcoin features. The recurring dropout enhanced the performance of GRU compared to other ML techniques with Big Data. Yassin *et al.*,[39] proposed MLP-based nonlinear auto-regressive model with exogenous inputs to predict future bitcoin prices. The PSO-based technique was used to increase the number of input, hidden and output levels of the exogenous technique. The Levenberg-Marquardt technique was adopted in training the MLPs and modified after 4-iterations or epoch that led to an over-fitting of the model. Sin and Wang[40] introduced MLPNN with GA to predict the price trend of bitcoin sales with a set of 200 features. The MLPNN was employed to predict the target class as a binary classification, producing an output of zero(0) or one(1). GA was employed to reduce generalization errors and target output generated using the moving average of the MLPNN ensemble and re-trained to predict a day in advance on Bitcoin's prices. The MLPNN model produced a prediction accuracy of 53%. Kalpanasonika *et al.*[41]; developed MLPNN with 13-neorons and trained it over a period of 256 iterations to predict Bitcoin price movements. This resulted in 99% prediction accuracy with specific features of network settings, but could not perform well with other crypto-currencies like Litecoin, Etherium and Ripple, etc. Ho *et al.*,[42] used the SLTM neural network with linear regression technique to predict the investment values of Bitcoin's fetaures. The Graphic user interface was created to help user read in the 4-input features(open, high, low and close prices) and predict the next target value of bitcoin. The time series data was fitted, trained, and tested on a limited number of items. The SLTM neural network produced better prediction accuracy than the LR technique, but produced low accuracy with other Bitcoin features. Das[43] used wavelet transformers, auto-encoders, and SLTM deep learning models to forecast crypto-currency market values. A seq-to-seq encoder model with a tangent trigger function was created to convert input words into corresponding vectors with layers used to predict fluctuating prices of Bitcoin's.

III. MATERIALS AND METHODS

Dataset: Experimental data was obtained from a public library made available at https://www.investing.com/crypto/bitcoin/historical-dat which has attributes: open price, high, low and closes as shown in Table 1. The datset was divided into 80%(2206) training and 20%(552) testing set with attributes.

Table1: Bitcoin Time dataset

	Dates	Open	- - -	Volume	Weight_Price
0	9/17/2014	465.864	- - -	21056800	460.95
1	9/18/2014	456.86	- - -	34483200	437.82
2	9/19/2014	424.103	- - -	37919700	407.82
3	9/20/2014	394.673	- - -	36863600	404.19
4	9/21/2014	408.085	- - -	26580100	403.13
- -	- - -	- - -	- - -	- - -	- - -
2754	4/2/2022	46285.5		45868.95	46241.31
2755	4/3/2022	45859.13	- - -	46453.57	46315.07
2756	4/4/2022	46445.27	- - -	46622.68	46273.71
2757	4/5/2022	46624.51	- - -	45555.99	46207.86
2758	4/6/2022	45491.38	- - -	45413.17	45288.27

The study strategy consists of pre-processing steps, SVM, KNN, MLP model building, and model training, prediction and evaluation.

3.1 Preprocessing: involves dividing or grouping proposed system data into specified attributes, labels (input classes and targets), training, and validation in a way that improves model performance. We use the moving average smoothing technique to smooth out point segments with little change over time. It filters noisy data from randomly fluctuating points from the original set. A model trained with smoothed estimates of sample data yield results with better accuracy than unsmoothed data, which can be done using equation 1.

$$\text{Moving Average(MA)} = \frac{A_1 + A_2 + \ldots + A_n}{n} \qquad 1$$

where A1 Bitcoin prices and "n" is the total period. It uses historical data with current observations to show the trend of the underlying data and help provide the best fitting model. We set the sliding window to small width because larger windows can result in overly smoothed time series which can effectively negate the effect of the Bitcoin seasonal data trend. LOCalized regression (LOESS) and LOCally weighted regression(LOWESS) are used in the neighborhood points called from the Python package statistics models and have been used to control the degree of smoothing with the argument called fraction which controls the accuracy of neighborhood data points in the fitting indicates the model.

3.2 Algorithms employed: High-end non-linear SVM, KNN and MLP algorithms are used to predict the outcome of the target variable using Bitcoin's time series data. SVM is a statistical learning technique that uses machine learning methods in its predictions to solve regression or classification tasks[44]. By default, the SVM produces high resistance to model over-fitting, but this depends on the smoothing parameter C. The SVM handles a regression task using a quadratic optimizer. It accepts input data into a high-dimensional feature transformation using a Gaussian kernel and performs linear regression on the transformed space. Turn the above process into a nonlinear regression activity in low-dimensional space, as shown in equations 1 and 2.

$$\text{The dataset } G = \{(x_i, y_i)\}_{i=1}^{N} \qquad 2$$

Given an unknown function called g(x), we can determine a function called f that approximates g(x) given knowledge of the data set G, as shown in equation 2 below.

$$f(x) = \textstyle\sum_i \omega_i \emptyset_i(x) + b \qquad 3$$

where $\emptyset_i(x)$ is the feature data while ω_i and b are the estimated variables obtained from the dataset. We proposed to use the loss function insensitive to Ɛ, where the error below is not penalized. The SVM nonlinear linear regression class was adopted to map data in a high-dimensional space using the concept of pipelined soft voting to reduce white noise. Data points are allowed to fit within the inner margin with a penalty such that white noise does not occur in the training dataset and is controlled using the gamma (γ) parameter.

Algorithm 1: Support vector machine(SVM)

Step	**Processes involved**
1	Start
2	Find candidate_SV with closest pair from classification (SV=>support vector)
3	If there are violating points:
4	Find violating_points

5	Compute candidate_SV= candidate_SV + voilating_points)
6	If there is any α_p< 0 due to the addition of c to S that gives negative:
7	Candidate_SV = candidate_SV
8	Repeat module to prune all data points
9	end_if
10	end_if

The K-nearest neighbor: KNN algorithm assumes that similar points exist in close proximity or near to each other. KNN captures the idea of similarity sometimes called distance, proximity or closeness. We proposed a new weight function called the Gaussian kernel to solve the model over-fitting problem and calculate the weight value of the Gaussian kernel weight function KNN as shown below in equation 4.

$$w^{i} = \exp\left(-\frac{d\left(x^{(i)},\text{query}\right)^{2}}{\sigma^{2}}\right) \qquad 4$$

Where w is the weight variable, σ is a parameter and query is the estimation point. The dataset was split into k-folds with training and test sets in the cross-validation test. The decision boundary of KNN is relaxed around K training data samples of $\varkappa_r$, r=1,..,K as the closest point to the distance x *, and rank by majority vote among the neighboring k. We create a KNN model based on the training dataset. Given tabular set of data(D) = (X,Y). The implementation processes of KNN contains the following steps

Algorithm 2: KNN

Steps	**Processes involved**
1	Start
2	Load Dataset
3	initialize K to your chosen number of neighbors
4	FUNCTION KNN(f, X_train, K)
5	L<-{(X_t, Y_t), For i in range(n)} as training set of observations X with class Y_i
6	Calculate d(X_t, X) for i in range(n) Select set of KNN training points to the query points$(Y_t, Y_{t-1}, Y_{t-2}, \ldots, Y_{t-n})$
7	Sort ordered collection of distances and indices
8	Pick first K entries from the sorted collection
9	Get labels of the selected K entries
10	Forecast points with distance weight-weight voting
11	Return class category

Muti-layer perceptron neural network: MLP is a perceptron with multiple layers that has input, output and hidden layers[45],[46],[47]. The input, hidden and output layers of MLPs are completely interconnected. The data is sent to the model via channels weighed by the nodes[48]. The weights between inputs and hidden layers can be computed using the activation function(sigmoid, Relu and tanh). The forward pass computes outputs corresponding to the input and backward pass propagates the error backwards through the network starting at the output units. The MLP neural network structure has 6 inputs (date, high, low, closed, custom volume), 18 hidden levels and output level. The MLP structure includes:

Step 1: Model definition: We defined a sequential model to create a layer-to-layer stacking, as required by a multilayer perceptive neural network, and added layers one by one with the appropriate number of input features[49]. The Relu activation function was used on the first hidden layer set to 10 nodes and the sigmoid trigger functions on the second hidden layer set to 8 nodes.

Step 2: Model compilation: we used a stochastic(Adam's) gradient descending algorithm optimizer with the loss function to evaluate and search for different weights to improve the training model. Adam's is the best training algorithm selected for MLPs where data is entered one by one, neurons are activated; data is processed and emitted through a process called forward pass. The activation functions map the weighted input to the output and adjust the threshold values of activated neurons to amplify the output signal. The error value is calculated to compare the target with the expected output and redistribute it across the network. Network weights have been updated for calculation error and processed for all training examples. Cross-entropy was used as the basis and loss for the binary classification set on Keras with metrics adjusted for accuracy.

Step 3: model fitting: The MLP is trained to perform forward and back passes and iteration divided into batches. Epoch provides row passes of training data and the batch_size sent to take the length of training data. The model learns to map rows from input to output classifications to understand the correlation between independent and target variables[50].

Step 4: Evaluuate: We evaluated the MLPs using testing set with a function to pass the same input and output at training state. The function returns loss, validation loss, validation accuracy and accuracy computed using: accuracy=model.evalute(X_test, y_test) in Python where "X_test" is input and "y_test" as target.

Algorithm 3: MLPs

Step	**Processes involved**
1	Start
2	Begin Epoch
3	For each training sample, d, do:
4	Propagate input and forward through the network:
5	compute network output for d's input values
6	Propagate errors backward through the network:
7	for each network output unit
8	$\delta_j = O_j \cdot (1-O_j)*(t_1 - O)$
9	for each hidden unit j
10	$\delta_j = O_j \cdot (1-O_j)* \sum \delta_k \cdot W_{jk}$
11	Update weights w_j by back-propagation error using learning rule
12	$W_{ij}(New)=\Delta W_{ij} + W_{ij}(old)$ Where $\Delta W_{ij} = \pi.\ \delta_j.\ O_i$
13	End Epoch

In the back propagation we used output error to adjust the weights of inputs at the output layer and compute the error at the previous layer. We repeated this process of back-propagating errors through a number of layers using the 'RELU' and sigmoid activation function.

3.3 Forecasting: is a technique that accepts historical data as input to make an estimate of the direction and behavior of future data trends[51],[52]. It can extract short-term and long-term recurring data patterns[53]. We can rewrite the time series of future virtual currency prices based on the past as:

$$y=f(x) \qquad 5$$

Where 'y' is the observation over time as future observations and 'x' the input of past and present observations which can virtually be transformed into the following equation:

$$Y_{t+1} = f(Y_t, Y_{t-1}, Y_{t-2}, \dots, Y_{t-n}) \quad 6$$

But we don't know Y_{t+1}, the data of tomorrow reframed to have the full dataset starting from yesterday for the testing situation

$$Y_t = f(Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots, Y_{t-n-1}) \quad 7$$

Where y_t is the target as output of future prices while $y_{t-1}, y_{t-2}, y_{t-3}, \dots$ are the past prices for the training situation using a regression function.

3.4 Evaluation metrics used

This is the final step used to measure the model's performance in terms of accuracy, standard deviation, mean score, RMSE and receivers operating characteristic(ROC) curve and area under the curve (AUC) to determine which is best.

IV. RESULTS AND DISCUSSIONS

The proposed system results are presented and discussed in detail using suitable ML tools. The design and implementation was done with some improvements to provide better and more accurate Bitcoin prices.

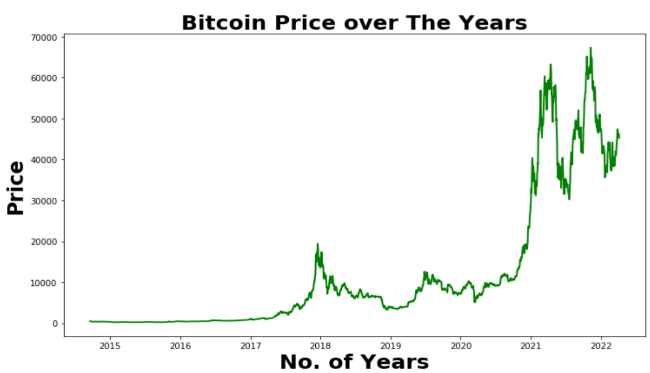

Figure 1: Bitcoin Price trend over the Years

Figure 1 is the graph used to provide an overview of opening, lowest and closing price behaviour(trend) of Bitcoins over time using historical data that ranges from 2015 to the years 2022.

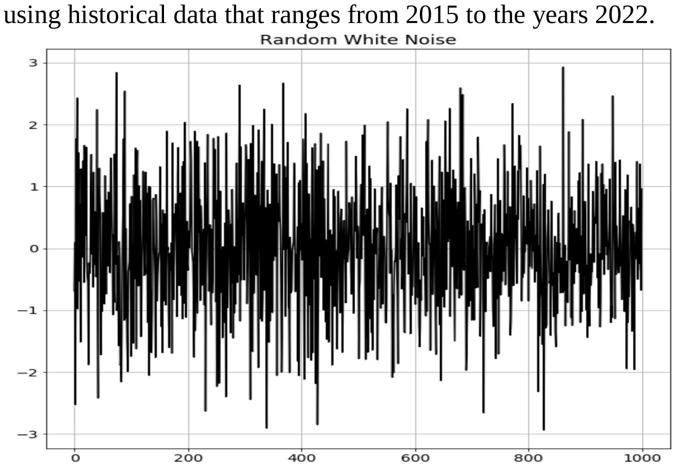

Figure 2: The Random White Noise

Figure 2 shows the presence of white noise in the time series having zero mean, equal variance, zero correlation and variables which are independent. This cannot be used to make reasonable prediction about random Bitcoon price movement which required time series smoothing to build a better model.

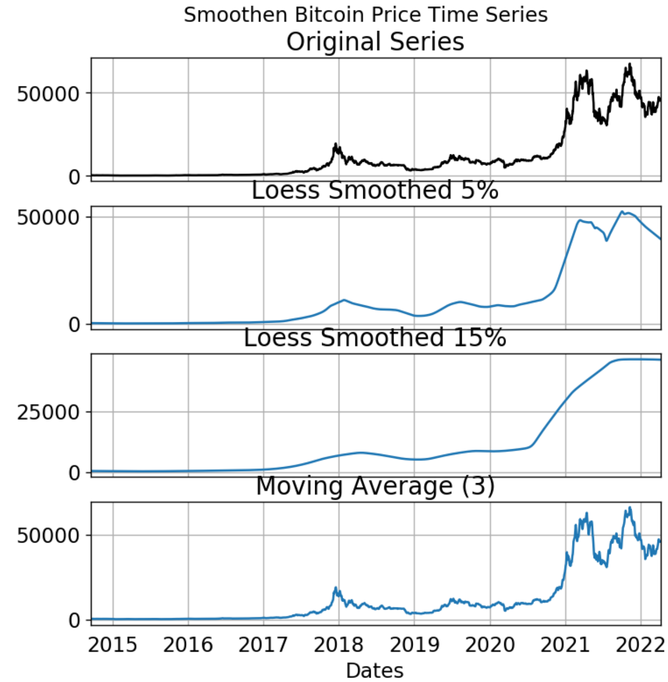

Figure 3: Smoothen Bitcoin Price Time series

Figure 3 is the graph of smoothing Bitcoin time series random variability providing lots of short and small term fluctuations to show underlying features. It's been applied to ignore or smoothen out small and short term fluctuations in the times using the average rolling mean of 3 with 5% and 15% fraction value across range of data. The 15% smoothed curve produced a better times series than 5% with the best underlying data features.

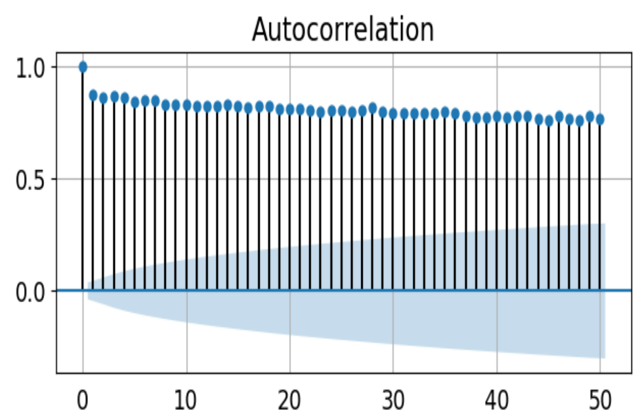

Figure 4: Autocorrelation

Figure 4 is the graph showing the spike corresponding to each lag ranging from 0, 10, 20, 30, 40 to 50 and the height depicts values of the autocorrelation function. The lag zero autocorrelation is equal to 1 for each term and itself. To every spike recorded above the 0.5 line is considered to be statistically significant measured to be different from 0 and highly correlated to each other. Meaning that; when the price of virtual currency increases, it will tend to continue increasing and when the price decreases, it will tend to continue decreasing.

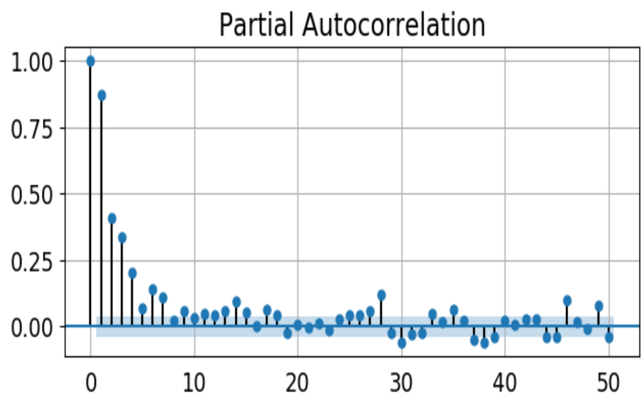

Figure 5: Partial Autocorrelation

Figure 5 is the partial autocorrelation graph of minimum daily Bitcoin price returns that resulted in white noise and recorded near zero for all registered lags. This reveals that lags are required in building the regression analysis to have a good model. The historical and present time series(lags) are helpful in predicting the future trend.

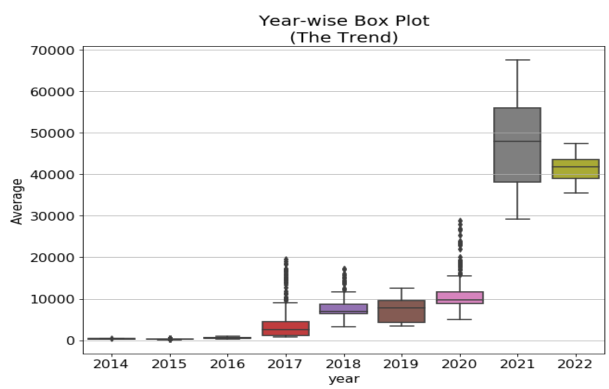

Figure 6: The Year-Wise trend of Bitcoin

Figure 6 depicts the Bitcoin yearly Box plot and there is change variability in the time series data from 2014 to 2022. The year 2021 produced the highest, followed by 2022, 2020 and decreases from 2019, 2018 to 2014.

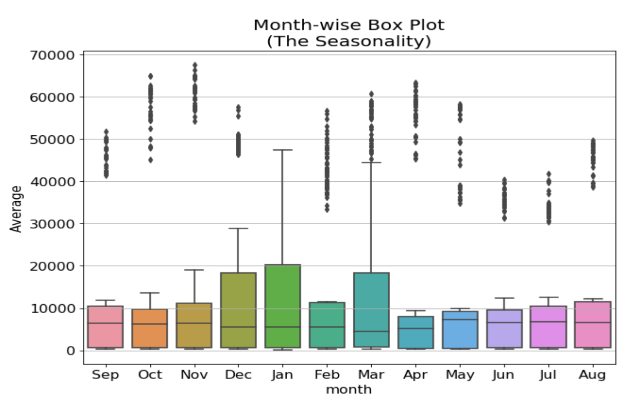

Figure 7: The Month-Wise trend of Bitcoin

Figure 7 is the Box plot of Bitcoin month-wise trend recorded a cyclic pattern obtained ranging from September to December and from January to August.

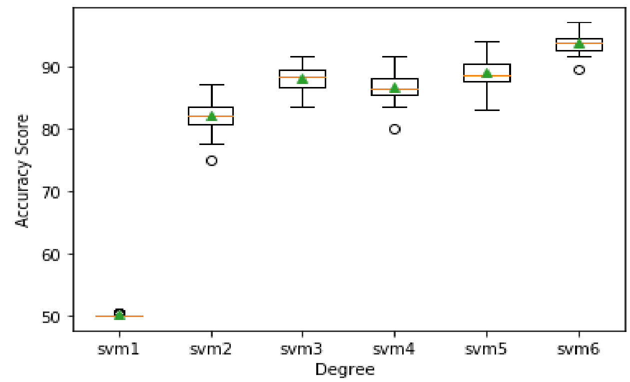

Figure 8: The accuracy plot of support vectors

Figure 8 is the plot of SVM accuracy rate against different fine-tuned support vectors. The prediction accuracy increased across different growing number of support vectors that ranges from 1, 2, 3, 4, 5 to 6. The value mapped to SVM6 produced the highest prediction accuracy, followed by svm5, svm4, and svm1 gave the least. The prediction accuracy increases along different support vectors as we increase the degree of hyper-parameter value.

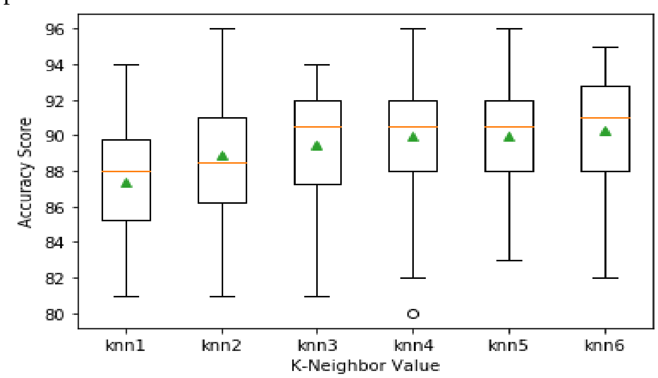

Figure 9: The accuracy plot of KNN

Figure 9 depicts the KNN accuracy rate across neighborhood values from 1, 2, 3, to 6 as obtained from the dataset. There is a significant increase in prediction accuracy against the k-neighbor values as we increase n_Neighbor values of the KNN technique. The k=6 produced 92% as the highest, followed by k=5, k=4 to k=1 having the least prediction accuracy.

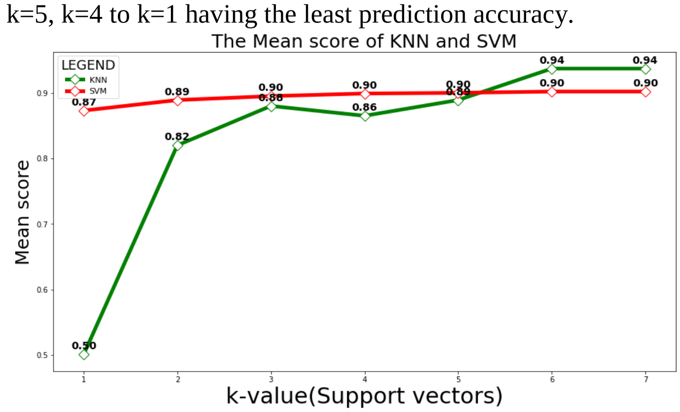

Figure 10: The graph of SVM and KNN Mean score

Figure 10 shows the KNN and SVM variation in mean score value across different k-neighbors. The means score value of SVM at k=1(0.87), 2(0.89), 3(0.90) are measured to be closer to 1 and higher than KNN. Also the KNN mean score value at k=6(0.94), 7(0.94) outperformed the SVM model in predicting the target variable.

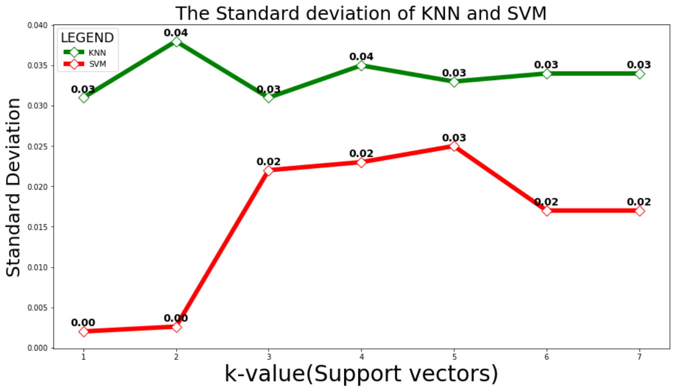

Figure 11: The standard deviation of KNN and SVM

Figure 11 shows the graph of KNN and SVM variation in standard deviation across different support vectors. The standard deviation of SVM at k=1(0.00), 2(0.00), 3(0.02), 4(0.03), 5(0.02) and 6(0.02) are all measured to be closer to 0 and smaller than the KNN. The KNN standard deviation at k=1(0.0.05), 2(0.04), 3(0.03), 4(0.04), 5(0.03), 6(0.05) and 7(0.03) outperformed by the SVM technique.

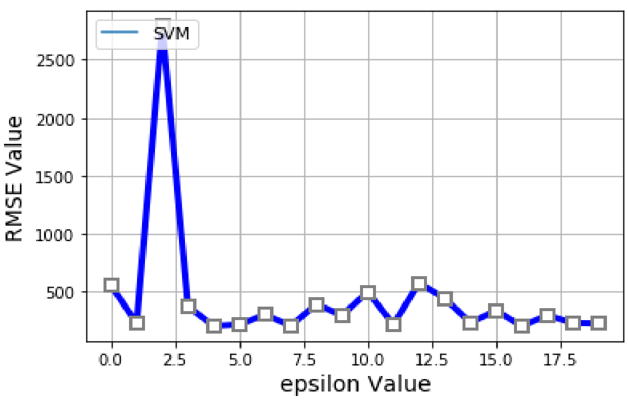

Figure 12: The SVM plot of RMSE against episolon

Figure 12 depicts the RMSE plot of SVM against epsilon hyper-parameter value. The RMSE value decreases from 0.0 to 2.5 and spiked up and decreases down in a cyclic trend as we finetune the epsilon hypaparameter value from 2.5 to 17.5.

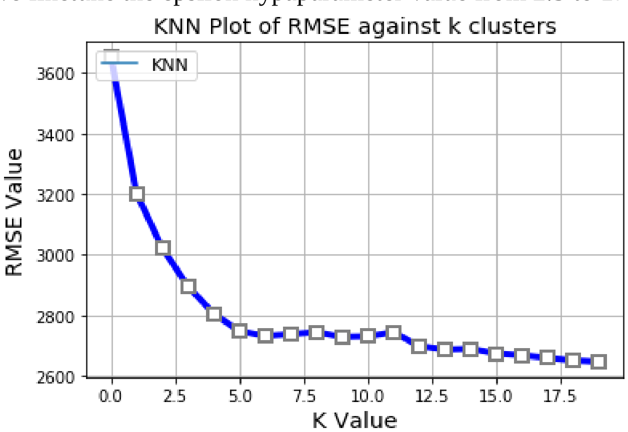

Figure 13: KNN plot of RMSE against K clusters

Figure 13 shows the Elbow plot of KNN RMSE value against k-clusters at training stage. There is a significant decrease in RMSE value across different fine-tuned hyperparameters at K-clusters ranging from 0.0, 2.5 to 17.5.

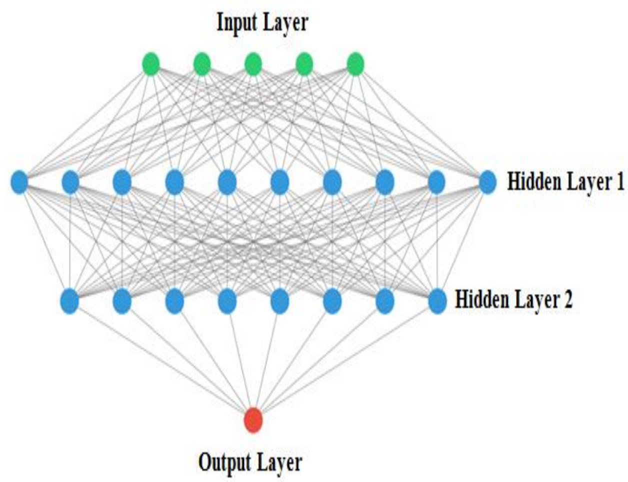

Figure 14. The MLPs neural network structure

The MLP design has 5-input nodes, 18-hidden layers(layer-1 containing 10-nodes and layer-2 with 8-nodes) and 1-output(high or low) visualized using Graphviz library in Python. Each neuron compute an activation function with two hidden layers sufficient enough to solve any problem. The increase in hidden layers does not really affect the model accuracy but depends on the complexity of the problem at hand. The input data was passed to the hidden layers through weighted channels and processed by the neuron with added bias. The trained MLP learn correlations and dependencies between target and independent variables. The learning process proceeds by way of presenting network with training set consist of input and response patterns. The validation error, accuracy and loss value of NN was computed using the "relu", and "sigmoid" activation functions at the first and second hidden layers. The error values are then used to alter the connection strength between layers in other to achieve a better network response.

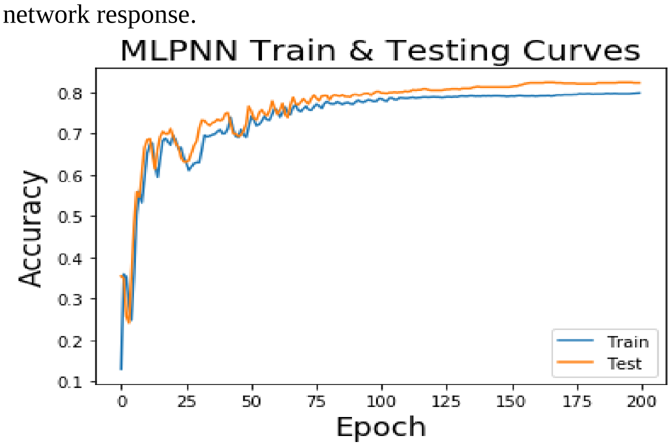

Figure 15: The learning curve of MLPs

Figure 15 is the ANN learning curve with training and testing set. The model performance is comparable in terms of accuracy on training and testing dataset. There is no consistency in the model training and testing data patterns across different iterations(Epoch).

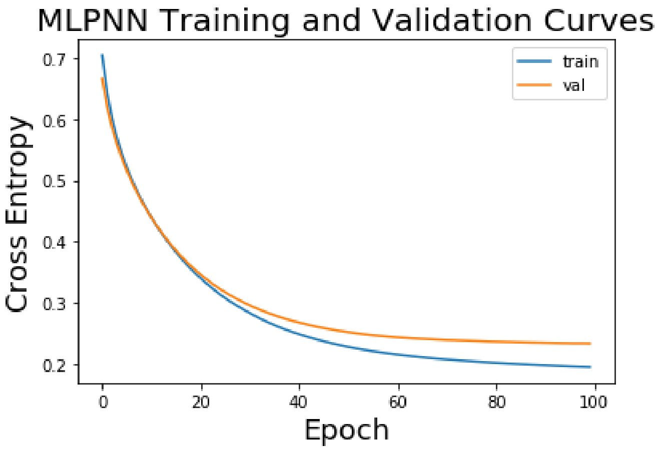

Figure 16: The learning curve of MLPs

Figure 16 is the learning curve which decreases across different Epoch(iterations) with minimal gap existing between the loss values. The gap between validation and training set is the gap of generalization which shows a learning curve with good fit. The point of training and validation loss decreases towards a point of model stability.

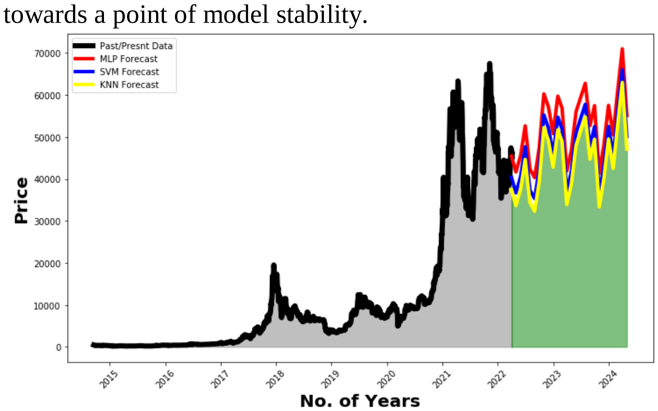

Figure 17: Forecasting Bitcoin Prices over time

Figure 17 is the future price prediction graph using past and present data with respect to time. The historical time series price plot of MLP ranges from 2015 to early 2022 represented with black color and predicted future values represented with red(MLP), blue(SVM) and green(KNN) color.

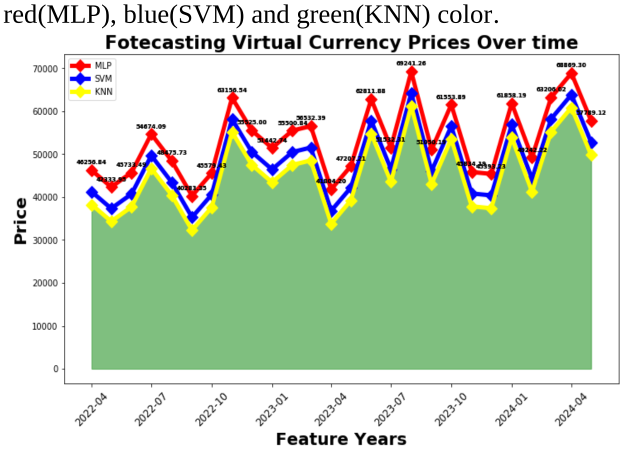

Figure 18: Future Bitcoin prices

Figure 18 depict the up and down movement of Bitcoin future price returns in the cause of time. There is a little deviation or variation recorded in future price fluctuations between SVM, KNN and MLP from late 2022 to 2024.

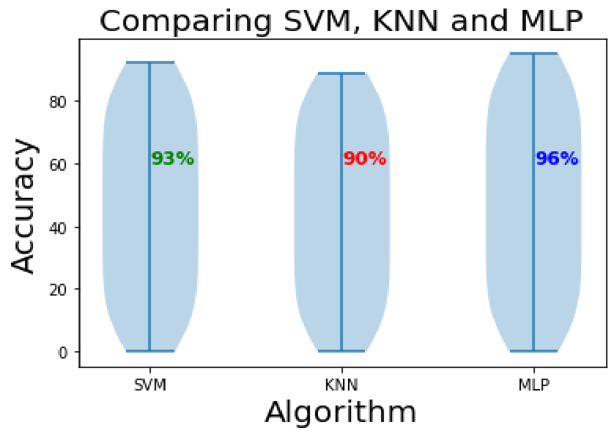

Figure 19: The Violin plot of SVM, KNN and MLP accuracy

Figure 19 shows the violin plot of MLP, SVM and KNN prediction accuracy. The SVM produced 93%, KNN gave 90% as the least and MLP recorded 96% accuracy as the best.

V. CONCLUSION

The selected MLP's and ML techniques will help investors to have a prior knowledge of feature price returns for the purpose of decision making. The SVM and KNN prediction accuracy reduces as we increase training samples in forecasting daily price returns. The performance metrics of MLP model increase along with more training dataset. The experimental results MLP proved to be highly efficient and accurate in forecasting future patterns of target variable. We therefore; evidently conclude that the MLP performed better with good fit.

REFERENCES

[1] M. A. Razi, and K. A. Athappilly, "Comparative Predictive Analysis of Neural Networks (NNs), Nonlinear Regression and Classification and Regression Tree (CART) Models," Expert Systems with Applications, vol.29, no.1, pp.65-74, 2005

[2] S. P. N. Ureshbhai, P. Bhattacharya, and S. Tanwar, "KaRuNa: A blockchain based Sentiment Analysis Framework for Fraud Cryptocurrency Schemes," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, pp.1-6, Jun. 2020.

[3] T. E. Koker, and D. Koutmos, "Cryptocurrency Trading using Machine Learning," *J. Risk Financial Manage*, vol. 13, no. 8, p. 178, Aug. 2020.

[4] N. A. Bakar, & S. Rosbi, Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction, International Journal of Advanced Engineering Research and Science, vol.4, no.11, pp.130-137, 2017.

[5] D. L. K. Chuen, L. Guo, and Y. Wang, "Cryptocurrency: A new Investment Opportunity?" *J. Alternative Investments*, vol. 20, no. 3, pp. 16-40, 2017.

[6] Y. B. Kim, J. G. Kim, J. W. Kim, J. H. Im, T. H. Kim, S. J. Kang, and C. H. Kim, "Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies," PloS One, vol. 11, no. 8, article no. e0161197, 2016.

[7] P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, "Stochastic Neural Networks for Cryptocurrency Price Prediction," *IEEE Access*, vol. 8, pp. 82804-82818, 2020.

[8] S. Tandon, S. Tripathi, P. Saraswat, and C. Dabas, "Bitcoin Price Forecasting using LSTM and 10-Fold Cross Validation," in *Proc. Int. Conf. Signal Process. Commun. (ICSC)*, pp. 323-328, Mar. 2019.

[9] H. Sebastiao, "Forecasting and Trading Cryptocurrencies with Machine Learning Under Changing Market Conditions," *Financial Innov.*, vol.7, p.3, Jan. 2021.

[10] J. Barkatullah, and T. Hanke, "Goldstrike 1: CoinTerra's First-Generation Cryptocurrency Mining Processor for Bitcoin," IEEE Micro, vol.35, no.2, pp.68-76, 2015.

[11] M. H. Yuneline, "Analysis of Cryptocurrency's Characteristics in Four Perspectives," *J. Asian Bus. Econ. Stud.*, vol. 26, no. 2, pp.206-219, Dec. 2019.

[12] M. Sivaram, E. L. Lydia, I. V. Pustokhina, D. A. Pustokhin, M. Elhoseny, G. P. Joshi, and K. Shankar, "An Optimal Least Square Support Vector Machine Based Earnings Prediction of Blockchain Financial Products," *IEEE Access*, vol.8, pp.21-30, 2020.

[13] C. Leopoldo, S. Grassi, and F. Ravazzolo, "Forecasting Cryptocurrencies Under Model and Parameter Instability," International Journal of Forecasting vol.35, no.2, pp.485-501, 2019.

[14] H. Jang, and J. Lee, "An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information," IEEE Access, 6, PP.5427-5437, 2018.

[15] S. Ha, and B. Moon, (2018). Finding Attractive Technical Patterns in Cryptocurrency Markets. Memetic Computing, vol.10, no.3, pp.301-306, 2018.

[16] J. S. Heo, D. H. Kwon, J. B. Kim, Y. H. Han, and C. H. An, "Prediction of Crypto-Currency Price Trend using Gradient Boosting," KIPS Transactions on Software and Data Engineering, vol.7, no.10, pp.387-396, 2018.

[17] T. Awoke, M. Rout, L. Mohanty, S. C. Satapathy, "Bitcoin Price Prediction and Analysis using Deep Learning Models," Springer: Communication Software and Networks, Pp.631–640, 2021.

[18] W. Yiying, and Z. Yeze, "Cryptocurrency Price Analysis with Artificial Intelligence, in Proc. 5th Int. Conf. Inf. Manage. (ICIM)," pp.97–101, Mar. 2019.

[19] H. S. Narman, and A. D. Uulu, "Impacts of Positive and Negative comments of Social Media users to Cryptocurrency," in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), pp.187–192, Feb. 2020.

[20] T. Rothman, and C. Yakar, "Empirical Analysis Towards the Effect of Social Media on Cryptocurrency Price and Volume," Eur. Sci. J. ESJ, vol. 15, no. 31, p. 52, Nov. 2019.

[21] S. Garg, "Autoregressive Integrated Moving Average Model based Prediction of Bitcoin Close Price," in Proc. International Conference Smart System, Inventive Technol. (ICSSIT), 473–478, 2018.

[22] A. Begum, A. Tareq, M. Sultana, M. Sohel, T. Rahman, and A. Sarwar, "Blockchain Attacks Analysis and A Model to Solve Double Spending Attack," *Int. J. Mach. Learn. Comput.*, vol.10, no.2, pp. 352-357, 2020.

[23] T. Zheng, A. A. Girgis, and E. B. Makram, "A hybrid wavelet Kalman Filter Method for Load Forecasting, Electric Power Systems Research, vol.54, PP.11-17, 2000.

[24] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system, " Available via DIALOG. http://bitcoin.org/bitcoin.pdf. Accessed 21 May. 2022.

[25] A. A. Saad, "Predicting the Price of Cryptocurrency Using Support Vector Regression Methods, Journal of Mechanics of Continua and Mathematical Sciences," vol.15, no.4, pp.313-322, 2020.

[26] A. Bisht, and P. Agarwal, "Analysis of Bitcoin using Linear Regression and Data Mining Techniques," International Journal of Advanced Research in Computer and Communication Engineering, vol.6, no.11, pp.43-46, 2017.

[27] M. Kenney, "Investigating the Predicting of a Chaotic Time-Series Data Using Reservoir Computing Deep-Learning and Machine-Learning on the Short; \Medium and Long Term Pricing of Butcoin and Ethereum," Dissertation, Technological University Dublin, pp.1-78, 2020, doi 10.2142//42ks-q868.

[28] H. Jen-Peng and S. D. Genesis, "Forecasting Bitcoin Returns: A Data Mining Approach, Review of Integrative Business and Economics Research," vol.10, no.1, pp.51-68, 2020.

[29] P. Patil, "Bitcoin Price Prediction Using Machine Learning and Neural Network," International Journal of Advances in Engineering and Management(IJAEM), vol.4, no.1, pp.847-852, 2022.

[30] Z. Wu, Predictions of Crypto-currency Prices Based on Inherent Interrelationships, *Advances in Economics, Business and Management Research:* International Conference on Financial Innovation and Economic Development (ICFIED), vol.648, PP.1877- 188, 2022.

[31] P. Sriramya, and L. S. Reddy, "A Research on Bitcoin Price Prediction Using Machine Learning Algorithms," International Journal of Scientific and Technology Research, vol.9, no.04, pp.1600-1604, 2020.

[32] S. A. Alahmari, "Using Nonlinear Machine Learning Algorithms to Predict The Price of Cryptocurrencies," International Journal of Future Generation Communication and Networking, vol.13, no.1, pp.745-752, 2020.

[33] P. Jaquart, D. Dann, and C. Weinhardt, "Short-Term Bitcoin Market Prediction Via Machine Learning," The Journal of Finance and Data Science, vol.7, pp.45-66, 2021.

[34] P. Mehta, and E. Sasikala, "Prediction of Bitcoin Using Recurrent Neural Network, International Journal of Recent Technology and Engineering(IJRETE)," vol.8, no.6, pp.1303- 1307, 2020.

[35]. B. Aygün , and E. G. Kabakçı, "Comparison of Statistical and Machine Learning Algorithms for Forecasting Daily Bitcoin Returns," European Journal of Science and Technology, no.21, pp.444-454 Jan. 2021.

[36]. H. Pabuccu, S. Ongan and A. Ongan, "Forecasting the Movements of Bitcoin Prices: An Application of Machine Learning Algorithms," AIMS Quantitative Finance and Economics, vol.4, no.4, pp.679–692, Nov. 2020.

[37]. R. K. Jane, I. Ghosh, and D. Das, "A diferential evolution-based regression framework for forecasting Bitcoin price," Springer Nature: Annals of Operations Research, pp.1-26, March 2021.

[38]. A. Dutta, S. Kumar, and M. Basu, A Gated Recurrent Unit Approach to Bitcoin Price Prediction, MDPI: Journal of Risk and Financial Management, vol.13, no.23, pp.1-16, Feb. 2020.

[39]. I. M. Yassin, A. Zabidi and Z. I. Rizman, "Non-linear Autoregressive with Exogeneous Input(NARX) Bitcoin Price Prediction Model Using PSO-Optimization Parameters and Moving Average technical Indicators," Journal of Fundamental and Applied Sciences, 9(3S), pp.791-808, Sept. 2017

[40]. W. Sin, and L. Wang, "Bitcoin Price Prediction Using Ensembles of Neural Networks," 2017 13th International

Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 666-671, 2017.

[41]. K. Kalpanasonika, S. M. Sayasri, A. Vinothini, and P. H. Suga, Bitcoin Cost Prediction Using Deep Neural Network Technique, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol.5, no.5, pp.96-101, April. 2019.

[42]. A. Ho, R. Vatambeti, S. K. Ravichandran, "Bitcoin Price Prediction Using Machine Learning and Artificial Neural Network Model," Indian Journal of Science and Technology, vol.14, no.27, pp.2300-2308, Aug. 2021.

[43]. S. Das, "Investment Using Cryptocurrency for a Machine Learning Based Stock Prediction Model," International Research Journal of Computer Science(IRJCS), vol.7, no.5, pp.70-112, May. 2020.

[44] D. H. Kwon, J. B. Kim, J. S. Heo, C. M. Kim, and Y. H. Han, "Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network, Journal of Information Process System," Vol.15, No.3, pp.694~706, June 2019.

[45] N. A. Hitam, A. R. Ismail, and F. Saeed, "An Optimized Support Vector Machine (SVM) Based on Particle Swarm Optimization (PSO) for Cryptocurrency Forecasting," Proc. Comput. Sci., vol.163, pp.427–433, Jan. 2019.

[46] S. Biswas, S. M. Pawar, S. Badole, N. Galande, and S. Rathod, "Cryptocurrency Price Prediction using Neural Networks and Deep Learning," in Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), pp.408–413, Mar. 2021.

[47] A. Radityo, Q. Munajat, and I. Budi, "Prediction of Bitcoin Exchange Rate to American Dollar using Artificial Neural Network Methods," in Proceedings of 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Bali, Indonesia, pp.433-438, 2017

[48] I. E. Livieris, S. Stavroyiannis, E. Pintelas, T. Kotsilieris, and P. Pintelas, "A Dropout Weight-Constrained Recurrent Neural Network Model for Forecasting the Price of Major Cryptocurrencies and CCi30 Index," Evolving Syst., Early Access, 1–16, Jan.2021.

[49] M. M. Patel, Tanwar, S., Gupta, R. and N. Kumar, "A Deep Learning-Based Cryptocurrency Price Prediction Scheme for Financial Institutions," Journal of Information Security Applications, vol.55, Art. no.102583, Dec.2020..

[50] A. Saxena, and A. Sukumar, "Predicting Bitcoin Price using LSTM And Compare its Predictability with ARIMA Model," International Journal of Pure and Applied Mathematics," vol.119, no.17, pp.25912600, 2018.

[51] E. A. D. Oliveira, I. E. Zarate, M. D. A. Reis, and C. N. CNobre, "The Use of Artificial Neural Networks in the Analysis and Prediction of Stock Prices," in IEEE International Conference on Systems, Man, and Cybernetics, 2011.

[52] G. Gomes, L. Dias, and M. Correia, "CryingJackpot: Network Flows and Performance Counters Against Cryptojacking," in *Proc. IEEE 19th Int. Symp. Netw. Comput. Appl. (NCA)*, pp.1-10, Nov.2020..

[53] I. A. Hashish, F. Forni, "Andreotti, G., Facchinetti, T. and Darjani, S. "A hybrid model for bitcoin prices prediction using hidden Markov models and optimized LSTM networks," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, pp.721-728, Sept.2019.

Authors Profile

Mr. Stanley Ziweritin, is a lecturer in the Department of Estate Management and Valuation, School of Environmental Design and Technology, Akanu Ibiam Federal Polytechnic, Unwana-Afikpo, Ebonyi State. He holds HND (Computer Science) from Rivers (Now Kenule Benson Saro-Wiwa) State Polytechnic, Bori. PGD(Computer Science) and M.Sc(Computer Science) from University of Port Harcourt(UPH) respectively. His research interest revolves around: Artificial Intelligence(AI)/Machine Learning(ML), Data Mining, and Visualization, Algorithms. He has published in several international journals. He is a registered member of Computer Professionals Registration Council of Nigeria(MCPN).

Dr. Emmanuel Nwaeze is a senior lecturer at Alex Ekwueme Federal University, Ndufu-Alike, Nigeria. He holds B.Sc Mathematics(UNIJOS), M.Sc(Mathematics FUTMIN) and PhD Mathematics(UNILORIN). He has published in several international journals.

Mr. Joshua Oluwasegun Jatto, holds BSc. (Computer Science) and MSc. (Computer Science) from University of Port Harcourt (UPH) respectively. He is presently working as an ICT Officer/Administrative Assistant in Our Kings International School, Port Harcourt, Rivers State, Nigeria. His research interest resolves around: Software Engineering, Database Management, Networking Security, Artificial Intelligent (AI), Expert System, Data Communication.