Cryptocurrency Price Estimation Using Hyperparameterized Oscillatory Activation Functions in LSTM Networks

Pragya Mishra and Shubham Bharadwaj

VIT University

pragya.mishra2019@vitstudent.ac.in

shubham.bharadwaj2013@vitalum.ac.in

Abstract

Activation functions are critical components of neural networks, helping the model learn highly-intricate dependencies, trends and patterns. Non-linear activation functions allow the model to behave as a functional approximator, learning complex decision boundaries and multi-dimensional patterns in the data. Activation functions can be combined with one another to learn better representations with the objective of improving gradient flow, performance metrics reducing training time and computational cost. Recent work on oscillatory activation functions[25][26] showcased their ability to perform competitively on image classification tasks using a compact architecture. Our work proposes the utilization of these oscillatory activation functions for predicting volume weighted average of Bitcoin on the G-Research Cryptocurrency Dataset. We utilize a popular LSTM architecture for this task achieving competitive results when compared to popular activation functions formally used.

1 Introduction

Deep neural networks have achieved state-of-the-art results across variety of tasks such as classification[1], regression[8], generative modeling[7] and meta learning tasks[6]. A critical part of these networks are activation functions which introduce non-linearity between the layers[2]. The choice of activation functions[27] used has an important effect on the model's ability to generalize. The combination of activation functions is also an important hyper-parameter for a model. Given their non-linear form, there is a trade-off between good performance and train time or computational cost[21]. Models which utilize oscillatory activation functions in the feature representation layers, such as Growing Cosine Unit[25] function or the Non-monotonic Cubic[26] function have achieved competitive performance while improving convergence speed.

In this work, the following hypothesis are covered as follows:

- Extend the utility of oscillatory activation functions for regression task over a domain specific open source dataset: G-Research Cryptocurrency dataset.
- Present and compare results using the above architecture with the previous benchmarks on cryptocurrency pricing.

Preprint. Under review.

2 Activation Functions

The nonlinear activation functions is an essential composition of any network. Despite the critical importance of the nature of these functions in determining the performance of neural networks, simple monotonic non-decreasing nonlinear activation functions are universally used. The newly proposed oscillatory nonlinear activation functions in deep neural networks is a family of functions that hold variety of properties pertaining to zero value approximation, differential and continuity behaviour, range and monotonicity. These functions are derived and inspired from dendritic action potentials in human layer 2/3 cortical neurons[11][26] and have the capability of solving XOR problem using a single neuron. In the past oscillatory [26][25] and non-monotonic activation functions have been largely ignored. Sigmoid and tanh[28] functions work for simple neural networks but cannot be used for deep neural networks due the vanishing gradient problem. The ReLU[4] activation function tackles the vanishing gradient problem but may take infinitely large values and cannot work for models with negative outputs. Popular functions such as swish[30] and mish[23] give better performance but at the cost of computational complexity and train time.

This paper presents experiments with 13 activation functions :

BIPOLAR

$$f_1(x) = 1 - e^{-x}/1 + e^{-x} \tag{1}$$

SILU

$$f_2(x) = x/1 + e^{-x} \tag{2}$$

SOFTPLUS

$$f_3(x) = \ln\left(1 + e^{x}\right) \tag{3}$$

LRELU

$$f_4(x) = [\begin{cases} 0.01x & x < 0 \\ x & x >= 0 \end{cases}] \tag{4}$$

GELU

$$f_5(x) = 0.5 \cdot x \left(1 + \tanh\left(\sqrt{\frac{2}{\pi}} \cdot x + 0.044715 \cdot x^3\right)\right) \tag{5}$$

SWISH

$$f_6(x) = x/1 + e^{-x} \tag{6}$$

MISH

$$f_7(x) = x \cdot \tanh\left(\ln\left(1 + e^{x}\right)\right) \tag{7}$$

MONOTONIC CUBIC

$$f_8(x) = x^3 + x \tag{8}$$

GCU

$$f_9(x) = x \cdot \cos(x) \tag{9}$$

DSU

$$f_{10}(x) = \frac{\pi}{2} \cdot (\operatorname{sinc}(x - \pi) - \operatorname{sinc}(x + \pi)) \tag{10}$$

SHIFTED QUADRATIC

$$f_{11}(x) = x^2 + x \tag{11}$$

NON MONOTONIC CUBIC

$$f_{12}(x) = x - x^3 \tag{12}$$

SHIFTED SINC

$$f_{13}(x) = \pi \cdot \operatorname{sinc}(x - \pi) \tag{13}$$

3 Properties to look for in an activation function

3.1 Vanishing Gradient problem

Training a neural network involves gradient descent that uses backward propagation. This step calculates derivatives to get the change in weights so as to reduce the loss after every epoch[31]. The gradients are calculated at every layer and in a deep neural network the gradients tend to vanish because of the depth of the network as the activation shifts the value to zero. This is called the vanishing gradient (saddle point) problem[14]. To support deep networks an activation must not have vanishing gradients[19].

3.2 Zero-Centered

Gradient descent depends heavily on slope of the loss function. It is important that gradients are not biased by the position of the graph in the three dimensional space. Hence, for gradients to be unbiased the output of the activation function must be symmetrical at zero so that the gradients do not shift to a particular direction.

3.3 Computational Expense

The activation functions need to be computationally inexpensive for the forward and backward propagation. They should enable seamless gradient flow, allowing networks to converge faster. Activation functions like sigmoid were used in early days, known to have higher computation time for training neural networks due to shape of the function, saturating and vanishing gradients across the range of inputs. This was countered by the use of ReLU activation and it's variants. Swish and mish activations adjusted the shortfalls of ReLU at the cost of computation, however the recently proposed oscillatory activation functions [25][26] were tested to outperform these functions on time complexity and performance.

3.4 Differentiable

An activation function must be differentiable at all points (except certain discrete points) in space for their derivative calculation during the backward pass. Dead activations are not desirable as gradient

Activation Function	Vanishing Gradient	Zero-centered	Computational Expense	Differentiable
Bipolar	Yes	Yes	No	Yes
Silu	No	No	No	Yes
Softplus	No	No	No	Yes
LRELU	No	No	No	No(x=0)
GELU	No	No	No	Yes
Swish	No	No	No	Yes
Mish	No	No	Yes	Yes
Mcubic	No	Yes	No	Yes
GCU	No	Yes	No	Yes
DSU	Yes	Yes	Yes	Yes
Shifted Quadratic	No	No	No	Yes
Non Monotonic Cubic	No	Yes	No	Yes
Shifted Sinc	Yes	No	No	Yes

Table 1: Lookup table for desirable properties of activation functions

flow is essential for the optimizer to optimize the loss function, thereby reflecting on the model performance.

In prior work [25], certain Oscillatory activation functions are shown to possess the following advantages:

- Alleviate the vanishing gradient problem. These functions have non-zero derivatives throughout their domain except at isolated points.
- Improved performance for compact network architectures.
- Computationally cheaper than the state-of-the-art Swish and Mish activation functions [24].

4 Long Short-Term Memory

Recurrent neural networks are deep neural networks that loops over the sequence of data passing a single timestamp in one iteration. It consists of repeating modules that take a single data point from the sequence of data, process it in a deep neural network and stores output as well as sends it to the next module.The following modules use an activation function to combine the previous information with the new input and then the process is repeated for the complete sequence of data.

With such large and deep neural networks the problem of vanishing gradient occurs ie. the gradient calculated during backpropagation becomes smaller every iteration and after a point the gradient is insignificant and the model cannot learn any further. LSTM[13] solves this problem by using an identity function in recurrency that has a derivative of 1, which is not less than 1 to avoid vanishing gradient and not greater than 1 to avoid exploding gradients problem. In a LSTM or a Long short-term memory[33] unit a combination of activation functions is used to store the patterns from previous data and decide if an information will be carried forward or deleted. A default combination is of tanh activation function for the cell state and the sigmoid activation function for the node output.

Adding an activation layer to any model makes it more flexible to learn patterns. Experiments on oscillatory activation functions and monotonic, non-monotonic functions have been successful in Convolutional Neural Networks. Now the question is how will they perform in a LSTM/Stacked LSTM or RNN model.

5 Experimental Setup

Everyday over $ 40 billion worth of crypto-currencies, the most popular assets for speculation and investment, are traded and marketed all around the world. The market capitalization of cryptocurrencies is estimated to be 266 billion USD and projected to have a growth of 11.9 percent by 2024

according to CAGR reports[17]. They are incredibly volatile and have fast-fluctuating prices making millionaires of a lucky few and resulting in immense losses to others. But is it possible to predict some of these movements even before they happen? Is the highly unstable crypto-exchange predictable?

The simultaneous activity of thousands of traders ensures that most signals will be transitory, persistent alpha will be exceptionally difficult to find, and the danger of overfitting will be considerable. In addition, since 2018, interest in the crypto-market has exploded, so the volatility and correlation structure in our data are likely to be highly non-stationary.

Since the model can easily overfit, the performance of each activation function can also be measured on a test dataset that will help us compare the activation functions. We have defined the G-Research Cryptocurrency dataset in training (1564924 records) and validation (195527 records) sets in observation period and test set (same size as validation set) in prediction period. All values are scaled using minimum-maximum scaler. The target variable we considered was volume weighted average price because opening and closing prices are highly correlated to this variable, and estimating this variable could possibly predict the bitcoin price. Early stopping, model checkpointing and time calculation, along with mean squared error loss function has been utilised in the trails.

5.1 Hyperparameters

The following hyper-parameters were experimented with :

- No. of nodes in LSTM layer: 32 units , 50 units
- For pooling we experimented with GlobalAveragePooling and MaxPooling
- Fully Connected Layer 1 : Dense (N1): 1024,512,256 units/neurons
- Fully Connected Layer 2 : Dense (N2): 512,256,128 units/neurons
- Optimizers compared were Adam and RMSprop
- Learning rates : 1e-3 ,1e-4, 1e-5,batch size: 1024,2048 and model(s) trained over 5, 10 number of epochs
- Various activation functions: oscillating and non-oscillating activation units

These parameters were explored extensively to deep dive into the distinctive functionalities which these oscillating activation functions bring for superior performance. We have only hyperparameterized commonly used activation functions in the non-oscillating category and compared them with the oscillating functions proposed in the previous work.

6 Previous Research in Crypto-currency prediction

Predictions in cryptocurrency prices predominantly have domestic and external factors. Some of the external influences are classified as follows:

- Market popularity for crypto trades: Market trend and Speculations.
- Financial factors: Equity, exchange rates, gold price, interest rate.
- Policies; Legalization, restrictions.

Some of the main factors like supply and demand, cost of transactions, compensation scheme, hash rate, circulation of coins and forks also hold major importance in pricing. Cryptocurrency investors and financial analysts seek proper investment decisions to acquire higher profits and for reviewing markets behavior around cryptocurrencies respectively[29].

Machine learning[18] and Deep learning algorithms have often been used to assess the factors affecting crypto-trades[15]. Architectures like Gated Recurrent (GRU)[5], Neural Networks (NN)[17] and Long-Term Memory (LSTM)/Hybrid LSTM framework units predicts several results pertaining to crypto-currencies, at times specific to some alt-coins lite coin and Montero. Some of the different cryptocurrency forecasts through machine learning techniques for Bitcoin prices, are reliable in nature, while others target the Buy and Hold strategies in the vast majority and counterbalance the effects of these schemes in the data pre-processing stage.

Various feature selection methods[35] have been previously tested for the best prediction attributes. Price trends prediction using the functionalities of Artificial Neural Networks (ANN), SVM and Ensemble algorithms has been investigated (after recurrent neural networks and the clustering process for kmeans). Bitcoin maximum, minimum and closing rates are also estimated by ANN and SVM. In addition, regression results have also been used as inputs to improve price forecasts.

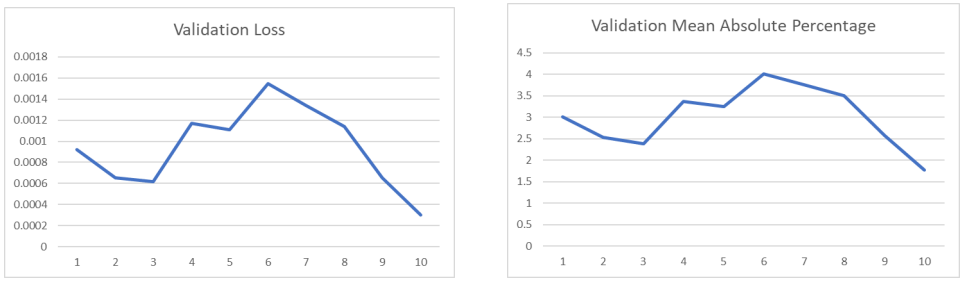
Task	Model	Metric	Value
Bitcoin Price Prediction	GRU[3]	**RMSE**	435.645
	Linear Regression[20]	**Accuracy**	0.65
	LDA[10]		0.6
	RNN[5]		0.9
	LSTM[13]		0.913
Bitcoin Exchange Rate Prediction	LSTM[13]	**RMSE**	354.5
	ANFIS[16]		430.8
	SVM[12]		546.9
Daily close price of Bitcoin Prediction	SVM[12]	**MEA**	0.1
	ANN[22]		0.2
	RNN[32]		0.19
	KNN[34]		0.18

Table 2: Performance of various models on different pricing tasks for Bitcoin

7 Results

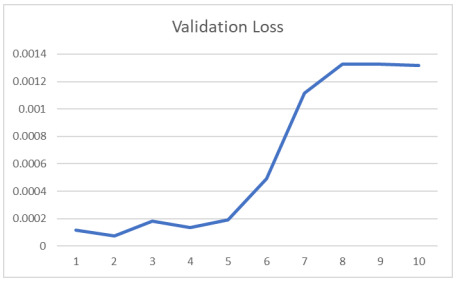
The experimental setup described above was demonstrated over a simple LSTM architecture to evaluate the prominent activation functions with the oscillating non-linear activations. Table 2 [15] shows previous benchmarks on the crypto pricing tasks. The best performance of the network was achieved by using 32 units LSTM layer along with global average pooling and N1 and N2 layers having 256 and 128 units/neurons respectively, with batch size of 1024 and learning rate of 1e-3 in adam optimizer trained over 10 epochs. It is clear from the results that Non-monotonic Cubic and Shifted Sinc have a comparable downward trend on validation loss and mean absolute percentage respectively with the frequently used LeakyReLU and GELU, suggesting a valid usecase at training time. Monotonic Cubic and Mish display similar train time behaviour as the graphs pivot at seventh epoch for both these activations. GCU, DSU and Shifted Quadratic showcase fluctuations, and could suggest saddle points or local minima issue, although computational hacks have been incorporated in the functions to handle these issues, further customization could be explored in these functions for better adaptability for task specific behaviour. However on lower learning rates, it was also observed that LeakyReLU and GCU had similar performance on this architecture. Non-monotonic cubic, DSU and Shifted Sinc also give a hint towards early convergence.

The loss and mean absolute percentage error of Activation Functions

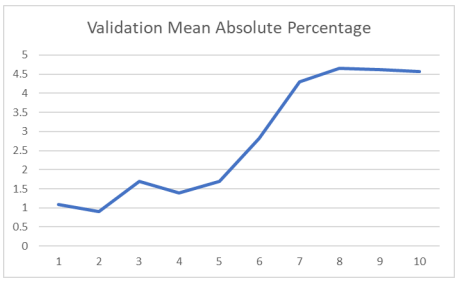


(a) (b)

Figure 1: Bipolar

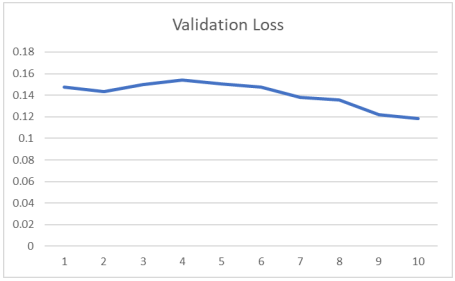


(a)

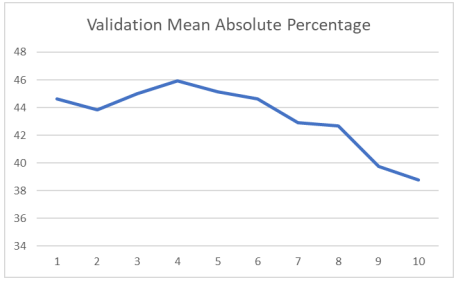


(b)

Figure 2: Silu

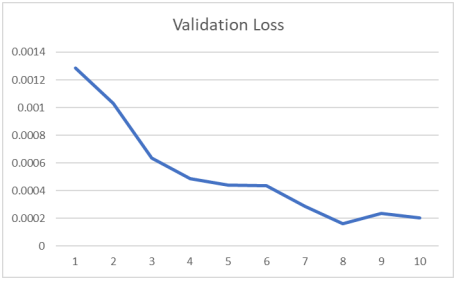


(a)

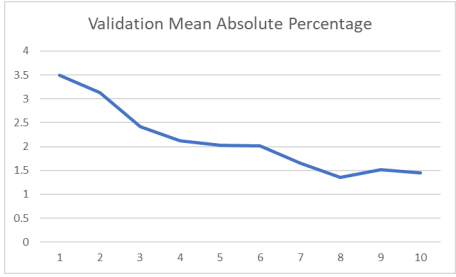


(b)

Figure 3: Softplus

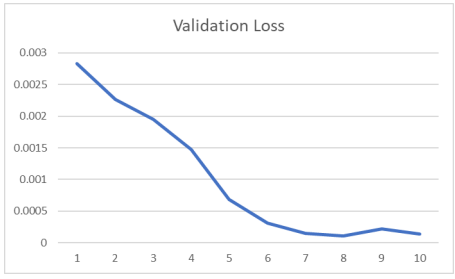


(a)

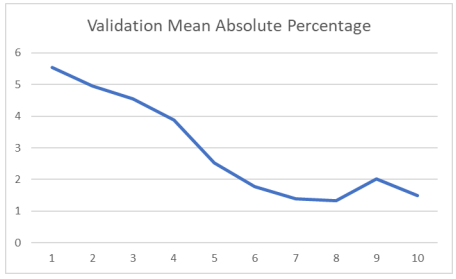


(b)

Figure 4: LRELU

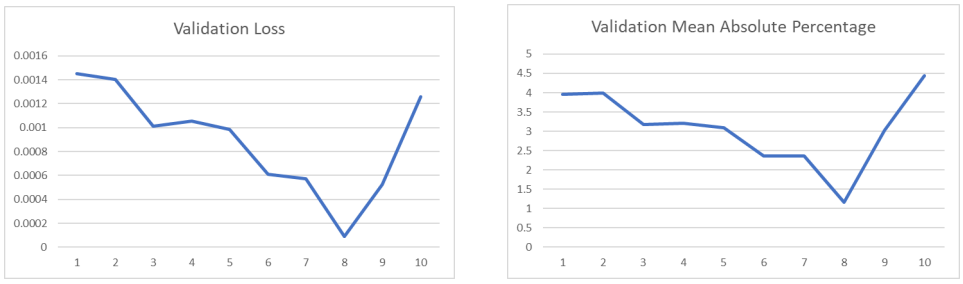


(a)



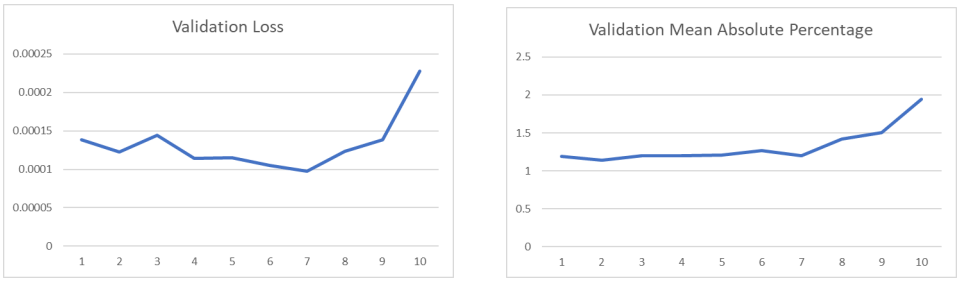
(b)

Figure 5: GELU



(a) (b)

Figure 6: Swish

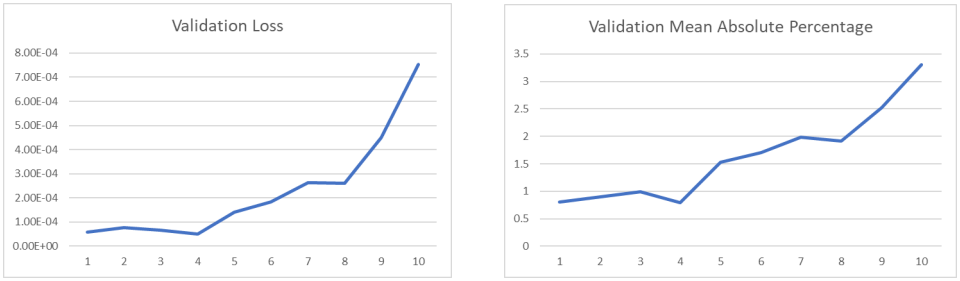


(a) (b)

Figure 7: Mish

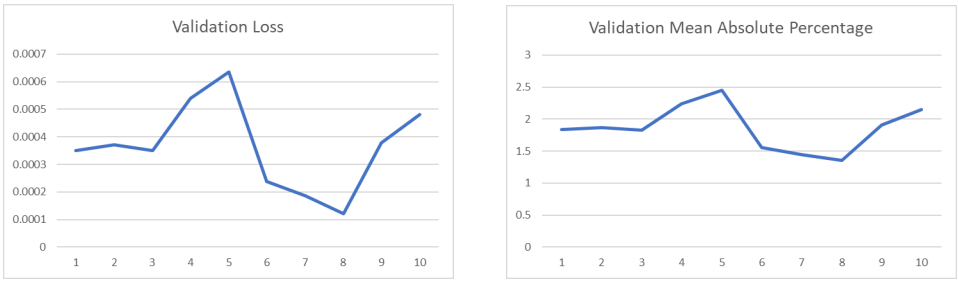
Overall, few activation functions such as Non-monotonic cubic and Shifted Sinc are suggesting superior train time performance over commonly used activation functions as per the results presented. Other oscillating functions like GCU, DSU, Monotonic cubic, and Shifted Quadratic are giving comparable performances, realising their use case along with other non-linear monotonic activation functions, The final network size for best performing network also suggests possibility of superior performance of oscillating activation functions with less number of neurons, which also remains to be explored in other commonly used backbones.

Performance of various activation functions are visualized and presented here for comparing the oscillating activation functions with other prominently used non-linear activations. These comparisons intend to demonstrate task specific usecases for these new functions, especially over sequential datapoints, similar to the time series crpyto-currency pricing data used in this study. Enhanced architectures like GRU, Autoencoders, Transformers etc. are yet to be explored with these new activations for investigating their performance gains.



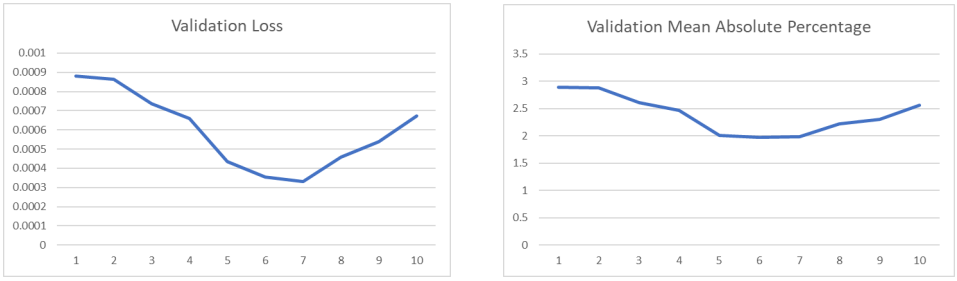
(a) (b)

Figure 8: Monotonic Cubic



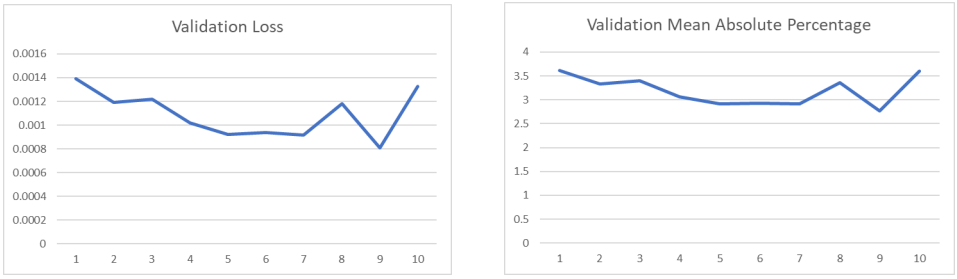
(a) (b)

Figure 9: GCU



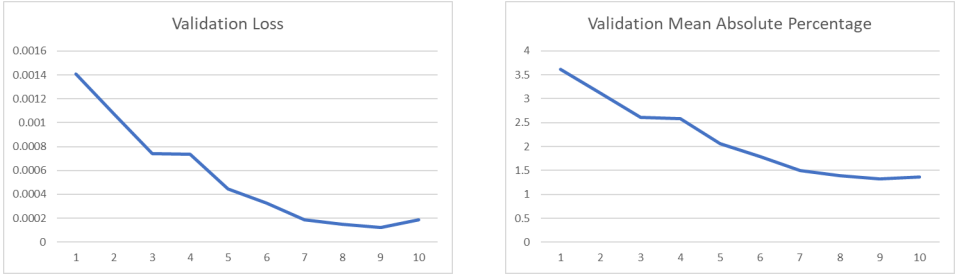
(a) (b)

Figure 10: DSU



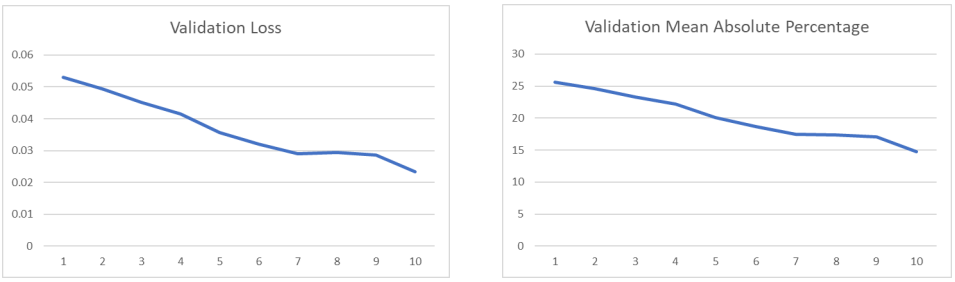
(a) (b)

Figure 11: Shifted Quadratic



(a) (b)

Figure 12: Non Monotonic Cubic



(a) (b)

Figure 13: Shifted Sinc

8 Metrics

Mean absolute percentage error

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}|\frac{Actual - Forecast}{Actual}| \tag{14}$$

Mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Actual - Forecast)^2 \tag{15}$$

Activation Function	MAPE	MSE
Bipolar	1.009817719	7.83E-05
Silu	4.744802475	0.000977032
Softplus	31.92780113	0.052854434
LRELU	0.90362674	5.13E-05
GELU	2.288985968	0.000210299
Swish	4.656949997	0.000958191
Mish	2.703600407	0.000307792
Monotonic Cubic	3.176057816	0.000471466
GCU	1.05540657	0.000100883
DSU	1.290743947	0.00013247
Shifted Quadratic	1.970283151	0.000326757
Non Monotonic Cubic	0.74698925	4.13E-05
Shifted Sinc	7.973043919	0.005605693

Table 3: Performance of oscillating activations in comparison with prominently used activations on test dataset

Table 3 presents the testing dataset performance of the activation functions under performance. As stated above, similar test time behaviour is observed, Non-monotonic Cubic beating LeakyReLU's performance on this dataset by a 20 percent gain in Mean squared error value while GCU,DSU and Shifted Quadratic Unit are giving comparable results and in-line with other activations like GELU, Swish, Mish and SiLU. Shifted Sinc did not perform as per the expected test time behaviour, suggesting likelihood of overfitting on functions using this activation on feature representation/extraction layers. To our surprise, bipolar[9] function gave a promising result comparable to Non-monotonic Cubic and LeakyReLU on test set, this could be due to it's combination of two soft sets one of them represents the positive side where the other represents the negative, thereby increasing the range of values[9]. Figure 14 provides the train time performance of various activations at training time. The graph suggests lower time for most of the activation functions when compared to LeakyReLU, GELU, Swis, Mish and SiLU. This suggests that performance along with time complexity makes some of the oscillatory activation functions ideal for training time and test environment gains.

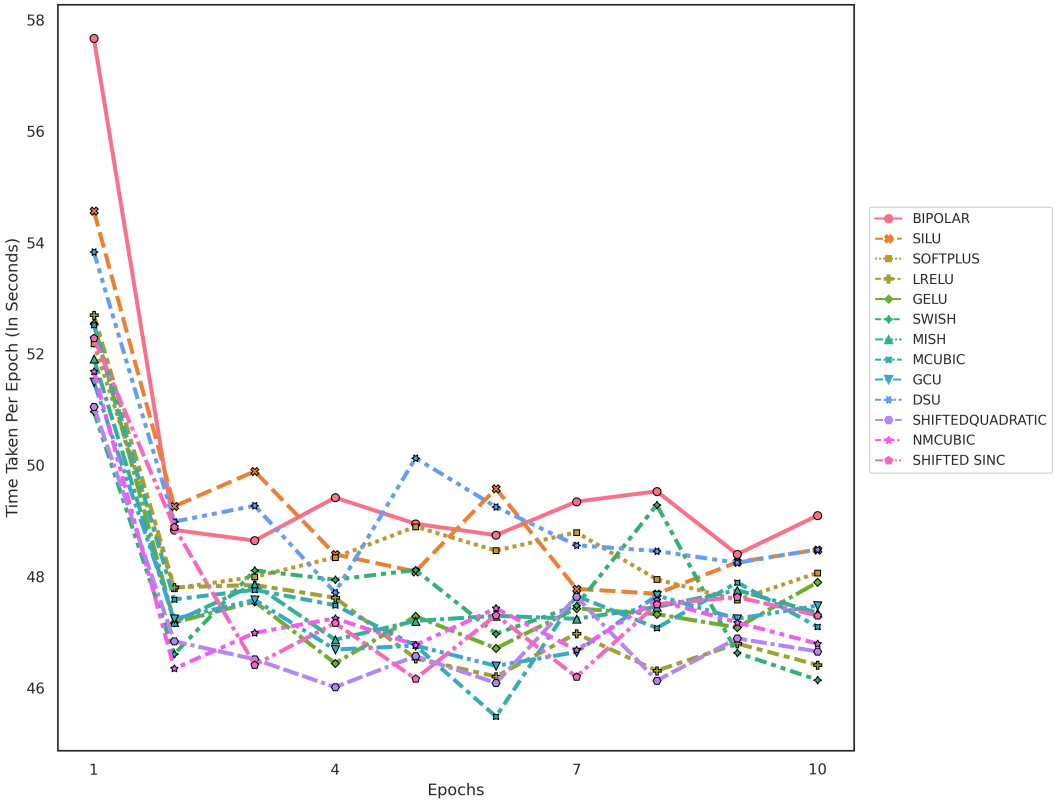


Figure 14: Time Taken Per Epoch For Various Activation Functions

9 Conclusion and Future Work

From the preliminary results a future work remains to test out other activation functions such as Shifted Quadratic Unit, Monotonic Cubic, Non Monotonic Cubic Unit, Shifted Sinc Unit, Decaying Sine Unit [26] etc. and see their impact on accuracy of neural architecture. We can extend this to other architectures such as VGG, ResNets, efficientnets etc. which could potentially improve performance. Other backbones like GRU's, Autoencoders and Transformer networks still remain unexplored and could potentially make use of these activation functions in future studies in NLP, computer vision and hybrid (image captioning, text-to-speech) tasks. The scope of usage of these activation functions could possibly be explored over other domain specific tasks to get a better understanding of their benefits over some of the predominant activation functions like ReLU, LeakyReLU, PReLU, GELU, Swish and Mish. The discovery of these new oscillating activations could possibly pave way for enhanced architectures capable of solving extremely non-linearly separable problems, however exploring general purpose and task specific usecases is also of the essence in order to have a value proposition along with other activation functions.

References

[1] Shubham Bharadwaj. Using convolutional neural networks to detect compression algorithms, 2021.

[2] Alon Brutzkus and Amir Globerson. Why do larger models generalize better? a theoretical perspective via the xor problem. In *International Conference on Machine Learning*, pages 822–830. PMLR, 2019.

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[4] Arun Kumar Dubey and Vanita Jain. Comparative study of convolution neural network's relu and leaky-relu activation functions. In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pages 873–880. Springer, 2019.

[5] Aniruddha Dutta, Saket Kumar, and Meheli Basu. A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, 13(2):23, Feb 2020.

[6] Praneet Dutta, Man Kit Cheuk, Jonathan S. Kim, and Massimo Mascaro. Automl for contextual bandits. *CoRR*, abs/1909.03212, 2019.

[7] Praneet Dutta, Bruce Power, Adam Halpert, Carlos Ezequiel, Aravind Subramanian, Chanchal Chatterjee, Sindhu Hari, Kenton Prindle, Vishal Vaddina, Andrew Leach, Raj Domala, Laura Bandura, and Massimo Mascaro. 3d conditional generative adversarial networks to enable large-scale seismic image enhancement, 2019.

[8] Praneet Dutta, Sparsh Sharma, and Pranav A Rathnam. Engine performance optimization using machine learning techniques. In *2015 SAI Intelligent Systems Conference (IntelliSys)*, pages 120–126, 2015.

[9] Asmaa Fadel and Syahida Che Dzul-Kifli. Bipolar soft functions. *AIMS Mathematics*, 6(5):4428–4446, 2021.

[10] Benyamin Ghojogh and Mark Crowley. Linear and quadratic discriminant analysis: Tutorial, 2019.

[11] Albert Gidon, Timothy Adam Zolnik, Pawel Fidzinski, Felix Bolduan, Athanasia Papoutsi, Panayiota Poirazi, Martin Holtkamp, Imre Vida, and Matthew Evan Larkum. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 367(6473):83–87, 2020.

[12] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[14] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. Overcoming the vanishing gradient problem in plain recurrent networks. *arXiv preprint arXiv:1801.06105*, 2018.

[15] Mahir Iqbal, Muhammad Iqbal, Fawwad Hassan Jaskani, Khurum Iqbal, and Ali Hassan. Time-series prediction of cryptocurrency market using machine learning techniques. *EAI Endorsed Transactions on Creative Technologies*, 4:6, 07 2021.

[16] J.-S.R. Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.

[17] Patel Jay, Vasu Kalariya, Pushpendra Parmar, Sudeep Tanwar, Neeraj Kumar, and Mamoun Alazab. Stochastic neural networks for cryptocurrency price prediction. *IEEE Access*, 8:82804–82818, 2020.

[18] Seçkin Karasu, Aytaç Altan, Zehra Saraç, and Rifat Hacioğlu. Prediction of bitcoin prices with machine learning methods using time series data. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2018.

[19] Janusz Kolbusz, Pawel Rozycki, and Bogdan M Wilamowski. The study of architecture mlp with linear neurons in order to eliminate the "vanishing gradient" problem. In *International Conference on Artificial Intelligence and Soft Computing*, pages 97–106. Springer, 2017.

[20] Walter Krämer and Harald Sonnberger. *The linear regression model under test*. Springer Science & Business Media, 2012.

[21] Alex Krizhevsky et al. Learning multiple layers of features from tiny images, 2009.

[22] Manish Mishra and Monika Srivastava. A view of artificial neural network. In *2014 International Conference on Advances in Engineering Technology Research (ICAETR - 2014)*, pages 1–3, 2014.

[23] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 4:2, 2019.

[24] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *CoRR*, abs/1908.08681, 2019.

[25] Mathew Mithra Noel, Arunkumar L, Advait Trivedi, and Praneet Dutta. Growing cosine unit: A novel oscillatory activation function that can speedup training and reduce parameters in convolutional neural networks, 2021.

[26] Matthew Mithra Noel, Shubham Bharadwaj, Venkataraman Muthiah-Nakarajan, Praneet Dutta, and Geraldine Bessie Amali. Biologically inspired oscillating activation functions can bridge the performance gap between biological and artificial neurons, 2021.

[27] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *CoRR*, abs/1811.03378, 2018.

[28] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[29] Emmanuel Pintelas, Ioannis E. Livieris, Stavros Stavroyiannis, Theodore Kotsilieris, and Panagiotis Pintelas. Investigating the problem of cryptocurrency price prediction: A deep learning approach. In Ilias Maglogiannis, Lazaros Iliadis, and Elias Pimenidis, editors, *Artificial Intelligence Applications and Innovations*, pages 99–110, Cham, 2020. Springer International Publishing.

[30] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a self-gated activation function. *arXiv: Neural and Evolutionary Computing*, 2017.

[31] Matías Roodschild, Jorge Gotay Sardiñas, and Adrián Will. A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4):351–360, 2020.

[32] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, Mar 2020.

[33] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019.

[34] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260, 2019.

[35] Αθανάσιος Ι Ζουμπέκας. Ανάλυση κρυπτονομισμάτων & προβλέψεις με τη χρήση μηχανικής μάθησης. B.S. thesis, 2018.

10 Appendix 1

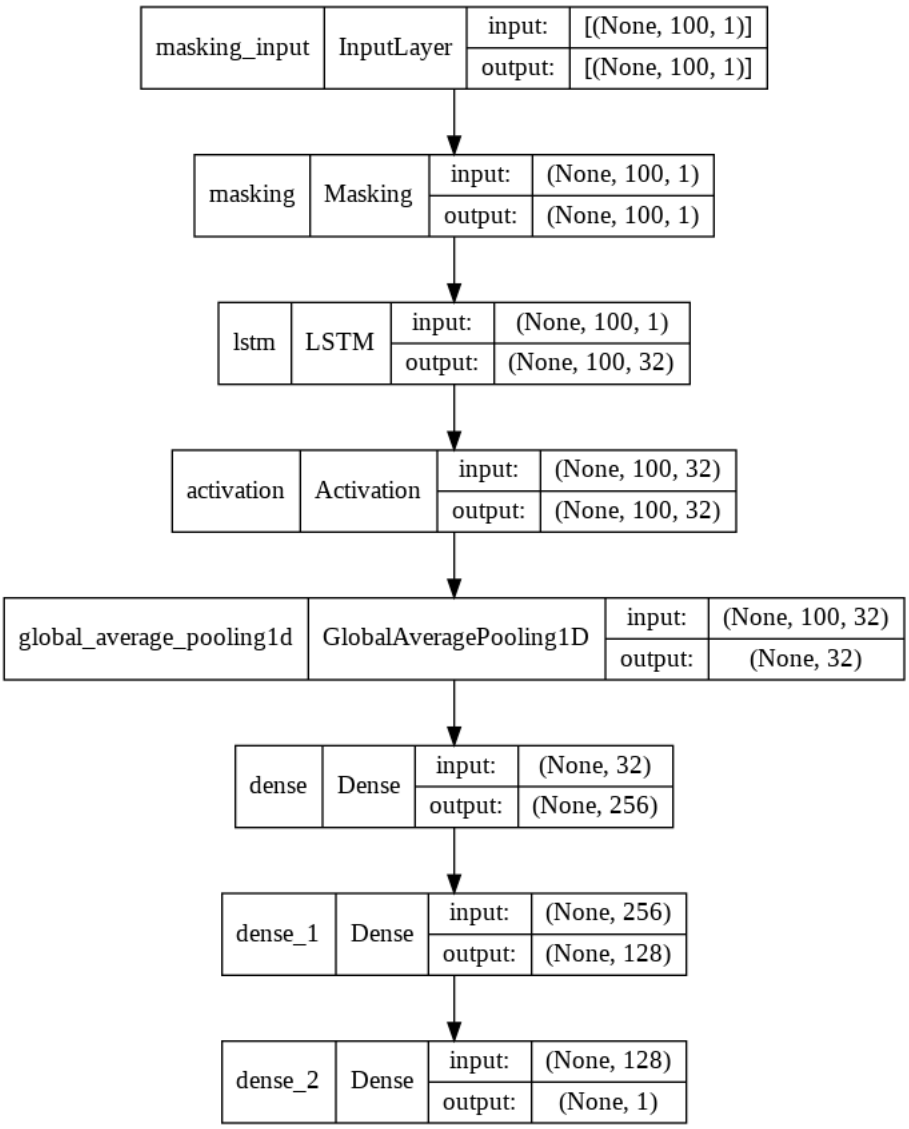


Figure 15: Architecture of the Crypto-market prediction model