Augmented Machine Intelligence with Human Intelligence for Cryptocurrency Price Prediction

Melika Honarmand*, Arushee Saxena, Sahaj Shandilya, Abhishek Chaudhary, Nikhil Meena, and Philip Treleaven

[1] ***Abstract*—This research explores the integration of human and machine intelligence in the financial industry. While AI systems excel at analyzing data, humans possess unique traits that contribute to accurate predictions. Inspired by the concept of Augmented Financial Intelligence, the study aims to integrate human intelligence with existing models, considering the limitations of current human input methods. Natural Language Processing and Sentiment Analysis are used to enhance prediction tasks, but limited training data and suboptimal processing can introduce noise. The study introduces a framework that integrates human and machine intelligence to enhance cryptocurrency market forecasts, analyzing six cryptocurrencies and utilizing an Elo-based rating system to identify exceptional predictors. The framework aims to minimize noise and optimize human input in financial forecasting.**

***Index Terms*—Augmented Intelligence, Cryptocurrency, Elo Rating, Super forecasting**

I. Introduction

The finance industry has been revolutionized by the advancement of Artificial Intelligence (AI) systems, which are powered by large amounts of data and cloud computing capabilities. The development of algorithmic trading and time series forecasting models has been possible due to these factors. Time series forecasting models have evolved to incorporate various factors that affect the series, including trend, seasonality, cyclic variations, and other regular and irregular variations.

ARIMA (AutoRegressive Integrated Moving Average) and SARIMA (Seasonal AutoRegressive Integrated Moving Average) models are extensively utilized for analyzing time series data by considering past observations, lags, and lagged forecast errors. Exponential Smoothing is another algorithm that utilizes an exponential window function to smooth time series data, assigning lower weights to observations farther from the prediction point. RNN (Recurrent Neural Network) approaches for time series prediction include LSTM (Long Short-Term Memory), which can effectively store, transfer, and process sequential data through specialized cell units. The LSTM model often employs the Adam optimization algorithm, which combines the strengths of AdaGrad and RMSProp algorithms to handle sparse gradients and noise issues. Recent advancements in this field include Facebook's Prophet algorithm and Amazon's SageMaker DeepAR forecasting algorithm.

Previous studies on stock price prediction have relied on historical data and incorporated human input in the form of textual data from social media and news articles. However, the credibility of such sources remains uncertain, and the preprocessing of data through NLP techniques introduces additional noise. In this work, a novel approach is proposed for accurate crypto-price prediction, referred to as the ZERO-NOISE human-machine augmented intelligence system. This system intelligently integrates human input without the need for preprocessing, and a rating system is implemented to assign weights to individuals' predictions based on their capabilities, contributing to the overall prediction outcome.

II. Data Collection

For the purpose of our research, we have specifically focused on studying the price dynamics of various cryptocurrencies. To obtain the required data, we utilized the yfinance API to access information from Yahoo Finance. The dataset encompasses a significant time period, spanning from January 2020 to November 2022, and includes the price data for Bitcoin (BTC), Ethereum (ETH), Binance (BNB), Dogecoin (DOGE), Polkadot (DOT), and Cosmos (ATOM). Each dataset is composed of 1064 data points, providing a robust foundation for our comprehensive analysis of cryptocurrency price movements.

III. Evaluation Methodology

The Mean Absolute Percentage Error (MAPE) is a widely used metric for time series forecasting [3]. It is particularly useful when only the magnitude of the difference between the predicted and true values is important, while the direction is irrelevant. The MAPE metric overcomes the large deviation bias in Root Mean Squared Error (RMSE) and exhibits robustness for datasets with long tails.

IV. Experimentation for Machine Prediction

A. ARIMA

ARIMA stands for Autoregressive Integrated Moving Average. The ARIMA model is composed of the AR (AutoRegressive), I (Differencing), and MA (Moving Average) models, each with its own set of parameters p, d, and q, respectively. In short-term prediction, ARIMA consistently exhibited superior performance compared to complex structural models [19].

The utilization of historical values of the variable being forecasted to predict future values is referred to as the AR component. In order to increase the accuracy of future

---

[1]

Manuscript received XXX; revised XXX; accepted XXX.

Melika Honarmand is at Amirkabir University of Technology,Tehran Polytechnic.

Arushee Saxena is at University College London, London.

Sahaj Shandilya, Abhishek Chaudhary, and Nikhil Meena are at Indian Institute of Technology, Kanpur, India.

Philip Treleaven is a Professor of Computer Science and Director of the Financial Computing Centre at University College London.

*Correspondence: melikahonarmand@aut.ac.ir

forecasts, the MA component makes use of historical prediction errors (the discrepancy between actual and expected values). In order to make the time series stationary and get rid of any patterns or seasonality, the I component requires making variations to the time series. These models are useful for short-term forecasting [12].

Two key assumptions of an ARIMA model are the linearity and stationary nature of the data[28].

The parameters for each dataset were chosen based on the auto-arima module of the Pmdarima library, which automatically discovers the optimal order for the model by trying out various combinations. The optimal orders were found to be different for different currencies.

B. EXPONENTIAL SMOOTHING

Exponential smoothing is a type of fixed-model prediction method for time series that use a weighted average of earlier sequence observations to forecast future values [15]. This technique was initially known as "exponentially weighted moving average" and involves assigning the highest weight to the most recent observation while progressively decreasing the weight as older observations are integrated into the calculation [10].

The main goal of exponential smoothing is to smooth out the original sequence and then use that smoothed sequence to predict future values of the variable of concern [16]. This technique is beneficial when the parameters of the time series are gently changing over time.

For this model, we divided the dataset into the training and testing sets mentioned above. The smoothing parameter is a critical aspect of this model, as it determines the balance between the model's bias and variance. To obtain the optimal smoothing level, we have used the optimized parameter available in the SimpleExpSmoothing function from the statsmodels library. This function performs a grid search over a range of smoothing levels and selects the one that results in the best fit to the data[29].

C. LSTM

Recurrent Neural Networks (RNNs) have gained significant recognition as a powerful model for effectively processing sequential data, including sound, time series data, and written natural language [21]. Long Short-Term Memory, or LSTM, is one of the RNNs architectures that, by introducing a memory component that enables it to selectively recall or forget information over time, gets around some of the drawbacks of conventional RNNs. For tasks involving long-term dependencies or temporal correlations between data points, including time series prediction, LSTM networks are ideally suited [13]. Although they can be more computationally expensive to train, they are more versatile than ARIMA models.

To model a regression problem, we predicted the price at a specific time as a function of some previous data points[26], using a sliding window with a window size/ time step of 20 days to predict the data at the end of those points [6]. We selected the value of 20 days based on prior experimentation with different values. During model fitting, we experimented with batch size and epoch parameters and used the ADAM optimizer as the optimization model [25]. We performed min-max scaling on the data before feeding it into the model, and after prediction, we performed the inverse transform to obtain the predicted prices.

D. DeepAR

DeepAR is a variant of the recurrent neural network (RNN) architecture that is specifically designed to handle time series data, such as stock prices, weather data, and website traffic. It can handle multiple parallel time series as input, and it uses a technique called "autoregression" to model the dependencies between the observations in a time series [14]. Currently, the DeepAR model has found extensive application in forecasting various phenomena, such as sales volume, traffic occupancy rate, electricity consumption, and deformation prediction [20].

The model is trained on historical data, and it can make predictions for future time steps. It is possible to use DeepAR with other models like CNN, but its strength particularly lies in handling multivariate time series problems. The main advantage of using DeepAR over other time series forecasting methods is its ability to handle missing values and categorical features with ease, as well as its ability to generate accurate forecasts even when dealing with long-term dependencies or complex patterns in the data [14].

We employed the Gluon Time Series library [24] to train a deep learning model for time series forecasting, using the DeepAREstimator class. This model utilizes a multi-layer LSTM network to capture temporal dependencies in the data. The Optuna library was used to perform hyperparameter tuning for the deep learning model [7]. This library is particularly well-suited for use in deep learning problems where the number of trials required to find an optimal solution can be very large, and computational resources are limited. We defined a function to take a trial object from Optuna as input, which was used to define the ranges of the hyperparameters to be optimized: num_layers, num_cells, dropout_rate,patience, base_lr, decay_factor,decay_factor and num_models. Then we trained the model with the given hyperparameters and evaluated its performance using an evaluation metric. The objective function returns the negative of the evaluation metric to be minimized.

E. FBProphet

Facebook developed the time series forecasting library known as FBProphet. To analyze and forecast time series data, it employs a decomposable model with elements of trend, seasonality, and holidays [11]. Additionally, FBProphet has built-in cross-validation, automatic outlier detection, and management for missing data. The FBProphet model gives predictions in the form of trends, accompanied by upper and lower limits. This unique feature enables the data to be fitted into the optimal model, allowing for more accurate forecasting [22]. It can quickly produce accurate forecasts, requires little parameter tuning, and is simple to use.

In this experiment, we fit the Prophet model with data consisting of the crypto prices in the data set from Jan 2020 to Nov 2022. The model is a single-variate time series regression model that is able to capture seasonality effectively but cannot use any side-information in the prediction [4].

The problem of overfitting and underfitting is dealt with by the change point prior scale parameter, which has a default value of 0.05. The Prophet detects change points throughout the observation series where at, the rate is allowed to change. The performance of the model declines if the seasonality factor is disabled, or the changepoint prior scale is increased or decreased. This is evaluated using Cross Validation for various horizons[30], and the one with the least MAPE is chosen.

F. XGBoost

XGBoost [17] is a decision tree ensemble based on gradient boosting designed to be highly scalable. It is a machine learning system based on [23]. Similarly to gradient boosting, XGBoost builds an additive expansion of the objective function by minimizing a loss function. In addition, XGBoost implements several methods to increment the training speed of decision trees not directly related to ensemble accuracy. Specifically, XGBoost focuses on reducing the computational complexity for finding the best split, which is the most time-consuming part of decision tree construction algorithms [18].

In this study, in order to optimize the performance of the model, we performed a grid search over hyperparameters [8]. Here, we tune a single hyperparameter called 'n-estimators' which specifies the number of trees in the forest. The initial value of the n-estimators parameter depends on the specific problem and dataset that is being used. We used this hyperparameter to train the final model and evaluated its performance on the test set.

V. RESULTS AND CONCLUSION ON MODEL CONCLUSION

The results obtained from the experimentation for the optimum model for the machine intelligent part of the systems are presented in this section. The MAPE scores of the models on different currencies are summarized in Fig. 1. As can be seen from the table, both LSTM and ARIMA models perform better than other models that were experimented upon. The MAPE for LSTM lies between 3.2 - 5.2 that for ARIMA lies between 2.8 - 4.8.

Model/Algorithm	eth	dot	bnb	btc	doge	atom
LSTM	0.041	0.034	0.032	0.044	0.052	0.049
ARIMA	0.036	0.031	0.028	0.02	0.042	0.048
DeepAR	0.2	nan	0.1	0.2	nan	nan
Exponential Smooth	0.865	0.249	0.362	0.747	0.788	0.695
XGBoost	0.082	0.145	0.046	0.265	0.098	0.069

Fig. 1. MAPE scores

Classical time series models like Prophet and XGBoost show significantly larger MAPE values in comparison to ARIMA and LSTM. The MAPE values of both LSTM and ARIMA are near each other, however, LSTM is considered to be better suited for time series forecasting due to its nonlinear nature as compared to the ARIMA model's linearity assumptions. This is further supported by various studies that suggest that LSTM and Neural Network models in general fit better to the fluctuations of prices [5].

Additionally, it is observed that ARIMA is better than LSTM at weekly and monthly forecasting, while LSTM is better at daily forecasting for rolling forecasting models [9]. This is because ARIMA requires fewer parameters for training and thus less data.

Based on the results obtained, LSTM is finalized as the machine intelligent part of the Augmented Financial Intelligence system.

VI. EXPERIMENTATION ON HUMAN INPUT DATA

The Human Input, that comprises the Human Intelligent part of the Augmented System is focused upon identification and isolation of superforecasters. The aim of the entire system is to gather this data without noise and in a processable format. In modern Augmented Financial Intelligence Systems, this data is gathered using sentimental analysis of financial news, Twitter data, etc. The data gathered from these sources can't really be trusted and can easily mislead if proper investigation isn't done before the gathering. Also, it requires enough preprocessing before use[27], not to mention that it has enough noise to degrade the NLP model. To avoid all this, we took to gathering data directly from their side in the form of predictions from their side about the future prices. The predictions are taken once every week to ensure that they have enough time to take into account the local fluctuations before making them. Though, this data still poses the problem of being unfiltered and can invite fraudulent predictions[31]. To avoid this, we have established a rating system that rates a person over time based on the accuracy of their prediction. This, in turn, enables us to sieve out the ones who possess better skills at predicting than others and weigh their predictions more. Moreover, it is the rating and ranking system that helps us decide the weights of the predictions of the people in accordance with their rating. The weights of the predictions of all individuals, including those of the machine, are calculated and then used to weigh their predictions, which together yield the final augmented prediction.

VII. THE RATING SYSTEM

1) The Augmented Financial Intelligence system utilizes a rating system that is based on the Elo rating system, which has been generalized for multiple participants [2]. This rating system is used to determine an individual's rating in comparison to others, which in turn influences the probability of their prediction being closer to the actual value. The probability of one individual being more accurate than the others is determined using Ra and Rb, which represent the ratings of individuals A and B, respectively. This probability is then used as the expected score that an individual should receive. By using this approach, the system can more accurately predict the performance of individuals and help identify those who are most likely to make accurate predictions. This rating system has been widely used in various fields and has been proven to be effective in predicting outcomes and ranking individuals in competitive environments.
2) In the Augmented Financial Intelligence system, a person's prediction is considered better if the error percentage between their prediction and the actual value is smaller. To rank individuals based on their predictions, a score is assigned to each rank, and this actual score required to make a rating change is calculated using an exponential function.

$$Score_R = h^{N-R} - \frac{1}{\sum_{r=1}^{N}(h^{N-r}-1)}$$

$$Score_R = Score\ associated\ with\ rank\ R$$

This function is used to ensure that the rating reward is greater towards the upper end of the ranking list. The use of an exponential function is justified because a linear function would have resulted in equal rating changes on all rank differences, which would have defeated the purpose of isolating the best forecasters faster. The hyperparameters of the Rating system, such as the exponent of the exponential rank-score calculation function, can be adjusted to establish the convergence of ratings faster or slower, depending on the capabilities of the predictors. This allows for greater flexibility in the rating system and enables it to adapt to changes in the environment. The AFI system leverages these techniques to accurately predict outcomes and identify the most skilled forecasters in its network.

As it only ranks forecasters based on their inputs, the rating system may still produce some erroneous outcomes. For example, if every user's prediction has errors, then the user with the lowest error rate will still receive the same amount of rewards as the user with 100% accurate predictions. In addition, since it ranks users based on their errors, two nearly accurate predictions may be rewarded quite differently despite their similarity.

So, in order to address these issues, the scores of the users are further modified based on their accuracy relative to the actual value. Thus, predictions that have fewer errors are rewarded more than those that have more errors, as they are more accurate. Therefore, distinguishing the superforecasters more clearly.

$$R_f = Score_R \times \delta \times f(\frac{user's\ error}{Max\ Error})$$

$$f(x) = e^{ax}u(x - \alpha) + e^{b(x-\beta)}u(\alpha - x)$$

$$R_f = Rewarding\ factor$$

$$\delta = Reward\ constant$$

u *is the unit step function, and* a, b, α, β *are some constant values that vary depending on the desired rewarding factor.*

Values used for calculation(upon experimentation) :

$a = ln(0.2)$
$b = ln(0.001)$
$\alpha = 0.3$
$\beta = 0.23.$

After stabilization, a rating difference of 400 signifies a ten times better prediction, and that of 200 signifies a prediction approximately 4 times better. This is because the probability of A performing better than B is set as

$$P = \frac{1}{1+10^d}$$

$$d = \frac{R_b - R_a}{400}$$

Before every round of prediction, we calculate the expected score of the person based on their rating and accuracy. And after the prediction process, we change these ratings based on these predictions.

This rating system not only rates players and creates a clear distinction between good and bad players, but the ratings also enable us to weigh their predictions as per their accuracy.

Overall, this rating system allows for a fair and objective assessment of the accuracy of predictions made by individuals in the Augmented Financial Intelligence system.

VIII. HUMAN INPUT DATA

This is the essential and final part of the prediction process. Here, we integrate the predictions that we got from the ML model and all the human predictions.

The website superforecaster.ch is used to collect predictions from users, who can provide their predicted prices for selected cryptocurrencies and optional reasoning and get rewarded accordingly.

As previously discussed, the ratings of an individual in the AFI system are a reflection of their prediction capabilities. To integrate these ratings, they are treated as weights. It is important to note that a constant difference in ratings represents a constant ratio of prediction accuracy. Therefore, when treating the ratings as weights, they are used on an exponential scale. This approach ensures that the contribution of higher-rated individuals is given greater weight in the overall prediction process. By using this method, the Augmented Financial Intelligence system is able to accurately predict outcomes and identify the most skilled forecasters in its network. This approach has been proven to be effective in various domains and has been widely adopted in competitive environments where accurate predictions are essential for success.

In the Augmented Financial Intelligence system, every individual and machine is assigned a base rating. Once all predictions are made, they are weighted and normalized appropriately to create a unanimous prediction. This is achieved by considering the ratings of each individual as a weight, with higher-rated individuals having a greater weightage. By using this method, the system is able to generate an accurate and reliable prediction.

After the true value arrives, the rating of each individual is updated, and the process is repeated. This allows for the rating of each individual to evolve over time and accurately reflect their current prediction capabilities. By using this recursive approach, the system is able to continually learn and adapt to changes in the environment. This ensures that the Augmented Financial Intelligence system remains at the forefront of predicting financial outcomes and identifying the most skilled forecasters in its network.

This system aids in our objective of separating the good estimators from the bad ones, whether they be machines or humans.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

Conclusion

We conclude with the benefits we predict upon implementation. We hope to generate highly accurate prediction results of various currencies and in future, we can move on to the prediction of stock prices and depending on the efficiency of the system, we shall take it beyond Financial Sciences and apply it to various other domains from mobility to game sciences. As the system clearly works on Human input, we hope on involving more and more people to our experiments in desire of finer results. The idea of Augmented Intelligence itself has high potential and can be exploited as a major upgrade from AI in all applications.

References

[1] A.-H. Mihov, N. Firoozye, and P. Treleaven, “Towards Augmented Financial Intelligence,” SSRN Electronic Journal, 2022.

[2] D. Cunningham, “Developing a generalized elo rating system for multiplayer games,” Medium, https://towardsdatascience.com/developing-a-generalized-elo-rating-system-for-multiplayer-games-b9b495e87802.

[3] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” International Journal of Forecasting, vol. 22, no. 4, pp. 679–688, 2006.

[4] Y. Indulkar, “Time series analysis of cryptocurrencies using Deep Learning & Fbprophet,” 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021.

[5] Q. Ma, “Comparison of Arima, ann and LSTM for stock price prediction,” E3S Web of Conferences, vol. 218, p. 01026, 2020.

[6] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[7] S. Shekhar, A. Bansode, and A. Salim, “A comparative study of hyper-parameter optimization tools,” 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2021.

[8] L. Sun, “Application and improvement of Xgboost algorithm based on multiple parameter optimization strategy,” 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2020.

[9] R. Zhang, H. Song, Q. Chen, Y. Wang, S. Wang, and Y. Li, "Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China," PLOS ONE, vol. 17, no. 1, pp. e0262009, 2022.

[10] E. Ostertagová and O. Ostertag, “Forecasting using simple exponential smoothing method,” Acta Electrotechnica et Informatica, vol. 12, no. 3, 2012.

[11] S. Dash, C. Chakraborty, S. K. Giri, and S. K. Pani, “Intelligent computing on time-series data analysis and prediction of COVID-19 pandemics,” Pattern Recognition Letters, vol. 151, pp. 69–75, 2021.

[12] “Forecasting: Principles and practice (2nd ed),” Chapter 8 ARIMA models, https://otexts.com/fpp2/arima.html..

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” International Journal of Forecasting, vol. 36, no. 3, pp. 1181–1191, 2020.

[15] S. Shastri, A. Sharma, and V. Mansotra, “A model for forecasting tourists arrival in JandK, India,” International Journal of Computer Applications, vol. 129, no. 15, pp. 32–36, 2015.

[16] S. Shastri, A. Sharma, and V. Mansotra, "Predicting Pilgrimage in Numbers to Shri Mata Vaishno Devi, Katra, J&K using Time Series Analysis," Int. J. Emerging Res. Manage. Technol., vol. 4, no. 10, pp. 102-106, Oct. 2015.

[17] T. Chen and C. Guestrin, “XGBoost,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[18] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” Artificial Intelligence Review, vol. 54, no. 3, pp. 1937–1967, 2020.

[19] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock price prediction using the Arima model,” 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, 2014.

[20] M. Dong, H. Wu, H. Hu, R. Azzam, L. Zhang, Z. Zheng, and X. Gong, "Deformation prediction of unstable slopes based on real-time monitoring and Deepar model," Sensors, vol. 21, no. 1, pp. 14, 2020.

[21] Z. C. Lipton, "A Critical Review of Recurrent Neural Networks for Sequence Learning," arXiv:1506.00019, 2015.

[22] N. K. Chikkakrishna, C. Hardik, K. Deepika, and N. Sparsha, “Short-term traffic prediction using sarima and FbPROPHET,” 2019 IEEE 16th India Council International Conference (INDICON), 2019.

[23] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Ann. Stat., vol. 29, no. 5, pp. 1189-1232, 2001.

[24] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, and Y. Wang, "Gluonts: Probabilistic time series models in python," arXiv:1906.05264, 2019.

[25] Z. Zhang, “Improved adam optimizer for Deep Neural Networks,” 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018.

[26] Wen Tian, Ying Zhang, Yinfeng Li, and Huili Zhang, "Probabilistic Demand Prediction Model for En-Route Sector," International Journal of Computer Theory and Engineering vol. 8, no. 6, pp. 495-499, 2016.

[27] Tsungnan Chou and Mingmin Lo, "Predicting Credit Card Defaults with Deep Learning and Other Machine Learning Models," International Journal of Computer Theory and Engineering vol. 10, no. 4, pp. 105-110, 2018.

[28] Zahra Mahdavi , Maryam Khademi, "Prediction of Oil Production with: Data Mining, Neuro-Fuzzy and Linear Regression," International Journal of Computer Theory and Engineering vol. 4, no. 3, pp. 446-447, 2012.

[29] Y.Radhika and M.Shashi, "Atmospheric Temperature Prediction using Support Vector Machines," International Journal of Computer Theory and Engineering vol. 1, no. 1, pp. 55-58, 2009.

[30] Jing Hong, "An Improved Prediction Model based on Fuzzy-rough Set Neural Network," International Journal of Computer Theory and Engineering vol. 3, no. 1, pp. 158-162, 2011.

[31] Farzin Owramipur, Parinaz Eskandarian, and Faezeh Sadat Mozneb, "Football Result Prediction with Bayesian Network in Spanish League-Barcelona Team," International Journal of Computer Theory and Engineering vol. 5, no. 5, pp. 812-815, 2013.

Melika Honarmand, born in Tehran, Iran, in 2001, is a final-year undergraduate student pursuing a bachelor's degree in computer science at Amirkabir University of Technology, Tehran Polytechnic.
Throughout her academic journey, Melika has actively engaged in various programs and fellowships to enrich her knowledge and skills. Notably, she successfully participated in the highly esteemed Summer@EPFL program, which facilitated her professional growth and proficiency in the field of computer science. Furthermore, her selection for the E3 program (the EPFL Excellence in Engineering Program) enabled her to broaden her expertise within this field. In recognition of her dedication and valuable contributions to research, Melika was honored with the prestigious NCCR Marvel Fellowship.

Arushee Saxena, born in 2001, is a final-year undergraduate student pursuing a bachelor's degree in computer science at University College London. She has been actively engaged in societies at her university and has been in the committee for the TEDxUCL and Tech society. She has recently finished an internship at Visa as a software developer intern.

Sahaj Shandilya, born in Dhanbad, Jharkhand, India, in 2003. He is a second year undergraduate student currently pursuing a Bachelor in Technology in Electrical and electronics engineering from the Indian Institute of Technology, Kanpur. He has been part of the ShARE Future Leaders organization since 2021. He has worked as software development intern in various startups and has keen interest in the field of Algorithmic Finance, Internet of Things, and Federated Learning. He has been an intern to the team TRiBE of INRIA.

Abhishek Chaudhary, born in Faridabad, India, in 2004 is a second year undergraduate student pursuing a bachelor's degree in chemical engineering at Indian Institute of Technology, Kanpur. He has worked on a variety of projects related to computer science, machine learning, and robotics. He has wide research interests mainly including computer vision, data mining, Artificial Intelligence, object detection and robotics.

Nikhil Meena, who was born in 2005, is a second-year undergraduate electrical engineering student at the Indian Institute of Technology, Kanpur. He is actively involved in the programming club at his university, where he, along with other members, organizes sessions, workshops, and competitions for his fellow students and works on related campus projects. Recent projects include Transpiler for IITK-Traveller (an esoteric programming language designed for IITK).

Philip Treleaven, is Professor of Computer Science and Director of the Financial Computing Centre at University College London. His research in computational finance and algorithmic trading has had a transformative impact on the financial services industry.