Quantum-like Decision Theory for Time-series Forecasting

Lizhi Xin [a,1], Kevin Xin [a,2,*]

[a]*XINVISIONQ, INC., Chicago, IL, USA*
*Corresponding author
[1] lizhi@xinvisionq.com
[2] kevin@xinvisionq.com

Abstract

The key to forecasting is information, almost all forecasting problems are caused by incomplete information. In this paper we propose a quantum-like evolutionary algorithm for time series forecasting from an information perspective. Based on reward learning, the quantum-like evolutionary algorithm gains valuable information from time series data to produce probabilistic forecasts. The quantum-like evolutionary algorithm utilizes operation matrixes to generate a population of virtual trajectories to simulate time series data, then compute the returns of each virtual trajectory generated, and finally by means of Genetic Programming to evolve the virtual trajectory with the maximum returns that is most approximate to the observed time series data as possible. The operation matrix with maximum returns is the one utilized to produce the probabilistic forecast. By using historical data from the Dow Jones Index and Crude Oil Prices, we show that our methodology is able to produce reasonable forecasts.

Keywords: time-series forecasting, machine learning, genetic programming, quantum superposition, natural selection, quantum-like evolutionary algorithm

1. Introduction

Time-series forecasting plays a crucial role in commercial business and academic research. In industries such as; forecasting stock market indexes [1-4], forex [5-6], cryptocurrency [7-10], electricity consumption [11-14], retail demand [15-16], wholesale prices [17], and raw data yield [18]. In applications of research topics such as; biological science [19], medicine [20-22], climate modelling [23-27], precipitation [28], semiconductor anomaly detection [29-32], neural spiking [33-35], neuronal behavior predictions [36-37], ECG readings [38-41], and earthquake seismic activity [42-45]. Other fields include predicting tidal waves [46-47] and traffic congestion data [48-49].

Of the traditional methods formulated for time-series forecasting [50], the first systematic approach is the Box-Jenkins model [51], which integrated the existing knowledge of the autoregressive and moving average methodologies. There have also been many other mathematical methods formulated such as autoregressive [52], exponential smoothing [53-55], and other structural models [56]. Many attempts to automate time series forecasting have been made [57-60], such as the development of various packages for Python [61-62] and R [63], as well as a plethora of open-source software tools [64]. Various competitions have been hosted with the aim of achieving the highest accuracy for time series forecasting [65-68]. With the increasing availability of data and access to computing power, machine and deep learning has been thrust to the forefront of many next generation and state-of-the-art forecasting methods and models [69-72]. The rising popularity of AI and generative AI tools has also seen the releases of commercial tools such as Amazon SageMaker AI DeepAR [73-74], Facebook Prophet [75], and Nixtla TimeGPT [76] for time series forecasting that all leverage some sort of machine and deep learning.

All the problems that arise for forecasting are due to incomplete information. This incomplete information phenomena are inherent in all uncertain environments (non-linear systems), such as the financial market. Because of the lack of complete information in an observed time series, we aren't able to accurately reconstruct the historical data and predict the future. Thus, the best we can to do is to "guess" what is to come – be able to make the most reasonable prediction among the many possible outcomes.

Essentially forecasting is a form of decision-making under uncertainty due to incomplete information. The main challenge for decision-making is that of a dual uncertainty: the first uncertainty (the external world) is the inherent unpredictability of mother nature; the second uncertainty (human nature) is the unpredictable irrational behavior that we sometimes act in. The uncertainty of the world clouds our judgement of exactly what action to take when making a decision, and in certain contexts, our actions influence and affect the ever-changing external world to some degree, thus presenting a significant challenge of modelling both the uncertainty of the objective and the subjective.

This work presents a quantum-like evolutionary algorithm [77-78] for time-series forecasting; the quantum-like evolutionary algorithm incorporates the quantum superposition principle to model the dualistic uncertainty of the external world and internal mind all under unified complex Hilbert Space. Based on reward learning, the quantum-like evolutionary algorithm scours out the most valuable information by studying the historical data to produce probabilistic forecasts. The quantum-like evolutionary algorithm that we've proposed does this in two parts: first, generate a population of simulated virtual trajectories and compute the returns of each simulated virtual trajectory, then second, genetic programming is applied to evolve the population of simulated virtual trajectories and output the one that has the maximum returns. This ensures that the outputted simulated virtual trajectory is the most approximate one to the actual observed time series data.

While in today's era of big data where huge amounts of data seem to be better, we believe for short horizon forecasting under uncertain environments a small data sample is good enough; the far distant past and remote future don't have an impact on the short horizon forecast; the far past doesn't influence what is happening now, and in the long term the far future is unpredictable. By studying more recent historical data the quantum-like evolutionary algorithm focuses on producing short horizon forecasts which is more in line with real world scenarios.

In this paper we used historical data of two real-world datasets: the Dow Jones Index and Crude Oil Prices. By deliberately selecting these two non-linear datasets with great uncertainty, we demonstrate the short horizon forecast capabilities of the quantum-like evolutionary algorithm with datasets that don't have complete information, which other forecasting methods can't handle well. We trained around 60 data points for each and produced a short horizon forecast of 15 points on each time series dataset, and the results of each forecast produced are reasonable.

The structure of this paper as follows: Section 2 details the methodology. Section 3 are the results. Section 4 is the discussion. Section 5 is the conclusion.

2. Methods

For any given time series, the data points recorded can be described as in (1).

$$\{(t_k, x_k, q_k)\}\ k = 1, \ldots, N \tag{1a}$$

$$q_k = \begin{cases} 0, x_k > x_{k-1} \\ 1, x_k < x_{k-1} \end{cases} \tag{1b}$$

Where t_k is time, x_k are the observed data points of what is being recorded, q_k is the trend (0: up or 1: down) of the time series.

For quantum-like evolutionary algorithm applied to time series forecasting there are three main components:

(1) Time series data $\{(t_k, x_k)\}$: the trajectory of the observed data needs to be forecasted.
(2) Operation matrix ρ_{matrix}: a density operator to simulate the trajectories of time series data.
(3) Returns R_{returns}: the evaluation metric that evaluates how effective the simulated trajectory is to the actual observed trajectory of the time series data.

For the quantum-like evolutionary algorithm, the main challenge faced from the beginning is the incomplete information presented by the observed time series. The way that the quantum-like evolutionary algorithm does so is to create a population of operation matrixes, with each operation matrix generating a single simulated virtual trajectory, and then the quantum-like evolutionary algorithm computes the returns of each operation matrix. After generations of evolution, the operation matrix with the maximum returns (most "adapted" to the observed trajectory) will be outputted and be used for the forecast of the time series.

Due to the infinite possibilities that arise from the uncertain trend of the time series ("external world"), the quantum-like evolutionary algorithm ("decision-maker") has to make an "educated guess" on whether the trajectory is up or down, thus there needs to be a way to effectively model both the trend of the "external world" and the actions that can be taken by the "decision-maker". To do so, we've called on the principle of quantum superposition, by drawing off of the concept that something can be "superposed" in multiple states "simultaneously"; we can "superpose" the trend of the "external world" (up and down) and the actions that can be taken by the "decision-maker" (believe whether trend is up and down) altogether under a unified complex Hilbert Space. Then to find the "best" most satisfactory operation matrix, we evolve the "fittest" one from the population of all generated operation matrixes by using Genetic Programming (GP), an algorithm based on Darwinian Natural Selection.

In this section, we first go over how to model the dual uncertainty of the "external world" and the "decision-maker's" actions with the concept of quantum superposition, then we detail how to apply GP to optimize the most satisfactory operation matrix to use for the forecast.

2.1 Modeling dual uncertainty with quantum superposition

The dual uncertainty of the trend of the time series and the actions that can be taken by the operation matrix both can be modeled under a unified Hilbert Space [79-80] as (2).

$$|Q\rangle = c_1|q_1\rangle + c_2|q_2\rangle \tag{2a}$$

$$|A\rangle = \mu_1|a_1\rangle + \mu_2|a_2\rangle \tag{2b}$$

Where $|q_1\rangle$ denotes at any observed point the trajectory of the time series is going up; $|q_2\rangle$ denotes the trajectory is going down; $\omega_1 = |c_1|^2$ is the frequency of the increase; $\omega_2 = |c_2|^2$ is the frequency of the decrease. $|a_1\rangle$ denotes that the operation matrix believes that the trajectory is going up; $|a_2\rangle$ denotes that the operation matrix believes the trajectory is going down; $p_1 = |\mu_1|^2$ are the degree of beliefs that the operation matrix "thinks" the trajectory is going up; $p_2 = |\mu_2|^2$ are the degree of beliefs that the operation matrix "thinks" the trajectory is going down.

The time series can be treated as a classical entity, which can be represented by a classical statistics operator as in (3).

$$\rho_{\text{world}} = \omega_1|q_1\rangle\langle q_1| + \omega_2|q_2\rangle\langle q_2| \tag{3}$$

The operation matrix is treated as a quantum-like entity because it can't "decide" whether what's being observed is up or down, it could be in a state of "believing" the trajectory of the time series is up and down "simultaneously" [81], which can be represented by a pure density operator as in (4).

$$\rho_{\text{matrix}} = |A\rangle\langle A| = p_1|a_1\rangle\langle a_1| + p_2|a_2\rangle\langle a_2| + \mu_1\mu_2^*|a_1\rangle\langle a_2| + \mu_1^*\mu_2|a_2\rangle\langle a_1| \tag{4}$$

The first term is where the operation matrix believes that the trajectory is up with probability p_1, second term is where the operation matrix believes that the trajectory is down with probability p_2, third and fourth terms are "quantum interference" terms, the operation matrix is in a "undecided" state of "simultaneously" believing that the trajectory is "up and down" [82-83].

Once the operation matrix "takes an action", it then either believes the trajectory is up or down and not both; this can be seen as the equivalent to the "collapse" when a "quantum measurement" is executed [84], as in (5).

$$\rho_{\text{matrix}} \xrightarrow{\text{decide}} p_1|a_1\rangle\langle a_1| + p_2|a_2\rangle\langle a_2| \tag{5}$$

The complex system of the "external world" and operation matrix can be described as (6).

$$\begin{aligned}\rho_{\text{world}} \otimes \rho_{\text{matrix}} = {} & \omega_1 p_1|q_1\rangle\langle q_1| \otimes |a_1\rangle\langle a_1| + \omega_1 p_2|q_1\rangle\langle q_1| \otimes |a_2\rangle\langle a_2| + \\ & \omega_2 p_1|q_2\rangle\langle q_2| \otimes |a_1\rangle\langle a_1| + \omega_2 p_2|q_2\rangle\langle q_2| \otimes |a_2\rangle\langle a_2|\end{aligned} \tag{6}$$

Where the first term denotes the trajectory of the time series ("external world") is up and the operation matrix "believes" that the trajectory is up, it guesses right and is rewarded; the second term denotes the trajectory is up and the operation matrix believes that the trajectory is down, it guesses wrong and is punished; the third term denotes that the trajectory is down and the operation matrix believes that the trajectory is up, it guesses wrong and is punished; the fourth term denotes that the trajectory is down and the operation matrix believes that the trajectory is down, it guesses right and is rewarded.

Based on (6), the return of the operation matrix "action taken" at any time t is as (7).

$$r_t = \begin{cases} \omega_1 p_1 x_{t,t-1}, \text{trajectory is up and the operation matrix believes with probaility } p_1 \\ -\omega_1 p_2 x_{t,t-1}, \text{trajectory is up and the operation matrix doesn't with probability } p_2 \\ -\omega_2 p_1 x_{t,t-1}, \text{trajectory is down and the operation matrix doesn't with probability } p_1 \\ \omega_2 p_2 x_{t,t-1}, \text{trajectory is down and the operation matrix believes with probability } p_2 \end{cases} \tag{7}$$

Where $x_{t,t-1} = |x_t - x_{t-1}|$ is the absolute difference between the value of the current point and the previous point. The total returns as in (8) is the sum of all the individual returns as in (7).

$$R_{\text{returns}} = \sum_{t=1}^{N} r_t \tag{8}$$

2.2 Optimize operation matrix by Genetic Programming (GP)

ρ_{matrix} is just a 2x2 matrix, (5) can be described as (9).

$$\rho_{\text{matrix}} = \begin{bmatrix}\rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22}\end{bmatrix} \xrightarrow{\text{decide}} \begin{bmatrix}p_1 & 0 \\ 0 & p_2\end{bmatrix} = p_1|a_1\rangle\langle a_1| + p_2|a_2\rangle\langle a_2| \tag{9a}$$

$$|a_1\rangle = \begin{bmatrix}1 \\ 0\end{bmatrix}, |a_2\rangle = \begin{bmatrix}0 \\ 1\end{bmatrix};\ |a_1\rangle\langle a_1| = \begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix}, |a_2\rangle\langle a_2| = \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix} \tag{9b}$$

$$\begin{cases} H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\ X=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\ Y=\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\ Z=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \\ S=\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}\ D=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\ T=\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}\ I=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \end{cases} \tag{9c}$$

We can approximately construct this density operator ρ_{matrix} with the eight most basic quantum gates as (9c). After constructing an individual operation matrix, we can then construct a population of operation matrixes, and then by using the fitness function as the evaluation criteria, the most satisfactory density matrix ρ_{matrix} from the population is evolved through generations of natural selection. The fitness function is defined as the total returns in (10).

$$f_{fitness} = R_{returns} \tag{10}$$

If there are M number of individuals in a population of operation matrixes, the most satisfactory operation matrix is the one that possess the maximum fitness function that can be described as in (11).

$$\rho_{matrix}^{output} = \arg\max_{a}\{f_{fitness}^{k}, k=1,\cdots,M\} \tag{11}$$

After finding the operation matrix with the highest returns, the simulated value x'_k for the virtual trajectory can be iteratively computed as in (12). $d'_{k,k-1}$ in (12c) is the simplest way to calculate the absolute difference value between two observed points used by the quantum-like evolutionary algorithm, for a more advanced way please refer to Xin L, et al (2023) [78].

$$q'_k = \begin{cases} 0, \text{operation matrix believes trajectory is going up} \\ 1, \text{operation matrix believes trajectory is going down} \end{cases} \tag{12a}$$

$$x'_k = \begin{cases} x'_{k-1} + d'_{k,k-1}, & \text{if } q'_k = 0) \\ x'_{k-1} - d'_{k,k-1}, & \text{if } q'_k = 1) \end{cases} \tag{12b}$$

$$d'_{k,k-1} = \left(\sum_{k=0}^{N-1} |x_k - x_{k-1}|\right) \Big/ N \tag{12c}$$

GP uses random crossover, selection, and mutation to formulate an executable program that solves problems accordingly [85-87]. First by randomly generating a certain number of individuals that comprise of a population, the algorithm obtains the fitness of each individual in the group and then by utilizing the principles of natural evolution for a number of generations it will optimize a most "satisfactory" solution to be used. The fittest ones that survive are the ones utilized, in line with the theory of natural evolution that states life has evolved through generations of selection, mutation, and crossover, the ones most adapted to the environment survive long enough to pass their genes off to the next generation [88].

The GP algorithm is shown in Algorithm 1.

Algorithm 1. GP Algorithm

Input:

- Historical dataset $\{(t_k, x_k), k = 0, \cdots, N\}$;
- Setting:
  (1) Operation set $F = \{+, *, //\}$;
  (2) Dataset $T = \{H, X, Y, Z, S, D, T, I\}$;
  (3) Crossover Probability = 70%; Mutation probability = 5%.

Initialization:

- Population: randomly create 300 individuals.

Evolution:

- Loop: for i = 0 to 80 generations.
  a) Calculate fitness for each individual based on the historical dataset;
  b) According to the quality of fitness:
     i. Selection: selecting parents.
     ii. Crossover: generate a new offspring using the roulette algorithm based on crossover probability.
     iii. Mutation: randomly modify the parent based on mutation probability.
- Output: An individual of the best fitness.

3. Results

All the training and forecasting was performed on a single general-use laptop. We trained and predicted on a relatively small dataset of about 75 data points, which was completed in a few hours. The parameters were set to a population of 300 individuals, evolving 80 generations, with the crossover probability at 70%, and the mutation probability at 5%.

3.1 Datasets

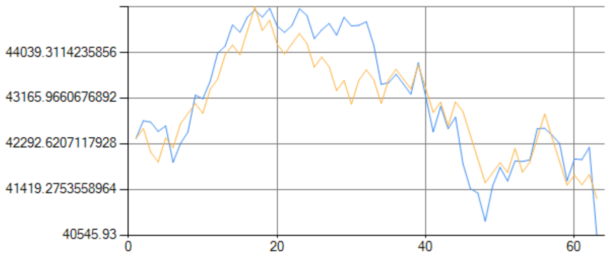
Two datasets – Dow Jones Industrial Average and Crude Oil Prices: West Texas Intermediate (WTI) – Cushing, Oklahoma – downloaded from the Federal Reserve Economic Data (FRED) and were used for all evaluations. The first 80% of the data was used for training and the remaining 20% of the data was used for verification (validation forecast). Regarding the forecast results of both datasets, the quantum-like evolutionary algorithm does two things: (1) forecast the one possibility that has the greatest chance of happening, and (2) forecast the trend curve. For (1), the quantum-like evolutionary algorithm generates 10,000 possibilities and runs each one once, then it selects the one that incurs the most frequently (yellow line). The one that is selected is then applied in the forecast process. For (2), quantum-like evolutionary algorithm generates 1,000 possibilities and runs them once each, then averages them all, which is then the result of the trend curve (red line).

3.2 Forecast results

Data used for the Dow Jones was from January 2nd, 2025 to April 25th, 2025; 80% of the data (Jan. 2nd, 2025-Apr. 3rd, 2025) was used as training data, while the remaining 20% (Apr. 4th, 2025-Apr. 25th, 2025) was used as verification. The entire training time took 5 hours, 45 minutes for the Dow Jones Index.

Data used for Crude Oil was from January 2nd, 2025 to April 21st, 2025; 80% of the data (Jan. 2nd, 2025-Mar. 28th, 2025) was used as training data, while the remaining 20% (Mar. 31st, 2025-Apr. 21st, 2025) was used as verification. The entire training time took 5 hours, 21 minutes for the Crude Oil Prices.

The key to all the graphs is: the blue line is the original data points of each dataset, the yellow line in the train graph is the fitting curve while the yellow line in the forecast graph is the predicted curve, and the red line is the trend curve.



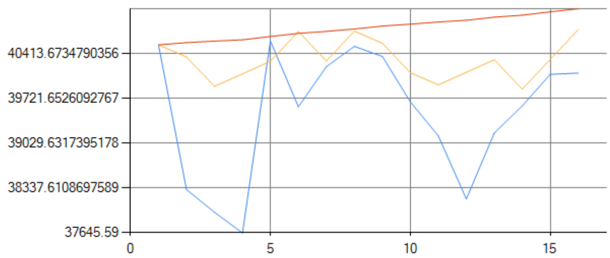


Figure 1: The training (left) and forecast (right) results of the Dow Jones Index.

Date	Observed closing price	Predicted closing price
2025-04-03	40545.93	40545.93
2025-04-04	38314.86	40361.38
2025-04-07	37965.60	39908.01
2025-04-08	37645.59	40101.56
2025-04-09	40608.45	40295.10
2025-04-10	39593.66	40754.47
2025-04-11	40212.71	40301.10
2025-04-14	40524.79	40760.47
2025-04-15	40368.96	40572.93
2025-04-16	39669.39	40119.56
2025-04-17	39142.23	39932.01
2025-04-21	38170.41	40125.56
2025-04-22	39186.98	40319.10
2025-04-23	39606.57	39865.74
2025-04-24	40093.40	40325.10
2025-04-25	40113.50	40784.47

Table 1: Result of the Dow Jones Index forecast outcomes

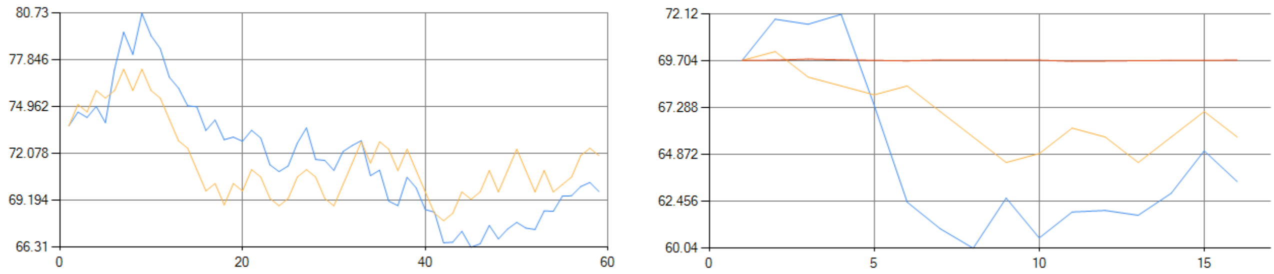


Figure 2: The training (left) and forecast (right) results of Crude Oil Prices.

Date	Observed closing price	Predicted closing price
2025-03-28	69.74	69.74
2025-03-31	71.87	70.19
2025-04-01	71.61	68.88
2025-04-02	72.12	68.42
2025-04-03	67.43	67.96
2025-04-04	62.42	68.42
2025-04-07	61.05	67.10
2025-04-08	60.04	65.78
2025-04-09	62.63	64.46
2025-04-10	60.57	64.92
2025-04-11	61.91	66.24
2025-04-14	61.99	65.78
2025-04-15	61.74	64.46
2025-04-16	62.88	65.78
2025-04-17	65.07	67.10
2025-04-21	63.48	65.78

Table 2: Result of Crude Oil Prices forecast outcomes

Figure 1 (Dow Jones Industrial Index) and Figure 2 (Crude Oil Prices) shows the training and forecast results by the quantum-like evolutionary algorithm. Table 1 shows the verification results of the Dow Jones Index; Table 2 shows the verification results of the Crude Oil Prices.

The forecast produced are reached by the principle of majority rules like in a system of court or voting procedure, the minority of people must obey the majority's opinion. In the case of the forecast, the operation matrix generates an action sequence which provides a judgement of whether the trajectory is going up or down for each predicted point; the operation matrix with the maximum returns produces 12 action sequences, then by taking all of them into account, majority rules are applied to generate a final action sequence. This final action sequence is what is applied to produce the virtual trajectory in (12).

3.3 Analysis

MAPE and RMSE metrics are used to evaluate the accuracy of the forecast results, they are defined as:

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{N} |(o_i - p_i)/o_i|}{N} \times 100$$

$$\mathrm{RMSE} = \frac{\sqrt{\sum_{i=1}^{N} (o_i - p_i)^2 / N}}{\sum_{i=1}^{N} |o_i| / N}$$

Where o_i is the observed value for item i, p_i is the predicted value for item i, and N is number of intervals that are going to be predicted. Table 3 shows the MAPE, RMSE, and Odds statistics for the Dow Jones and Crude Oil forecasts.

	Dow Jones	Crude Oil
MAPE	2%	5%
RMSE	0.03	0.06
Odds	66.67%	66.67%

Table 3: MAPE, RMSE, and Odds Metrics Accuracy

4. Discussion

The key to forecasting is information, but unfortunately historical time series datasets usually only contain incomplete information. It's difficult if not impossible to predict the absolute values for each observed data point by rigorous mathematical equations, for example to predict the closing prices of the Dow Jones Index. In this paper we proposed a methodology that first tries to simulate the up down movement at each observed point of the time series by an operation matrix, and then compute the absolute value; by studying the historical data with reward learning, the operation matrix will gain "knowledge" (valuable information) of the trend of the time series. Based on its "experience" the operation matrix will make a reasonable probabilistic forecast. Particularly for those datasets that are random, the operation matrix is more like a "mixed strategy" (with different "degrees of beliefs" to decide the whether the trajectory will be up or down), and as von Neumann said if your opponent is "randomly" playing cards then the best strategy to "counter" is to use a mixed strategy [89].

Majority rules according to May's theorem as outlined by Kenneth O. May, who "proved that the simple majority rule is the only "fair" ordinal decision rule, in that majority rule does not let some votes count more than others or privilege an alternative by requiring fewer votes to pass." The quantum-like evolutionary algorithm applies majority rules to increase the accuracy of forecasting, basically the operation matrix will produce a group of action sequences consisting of whether the trajectory is up or down at any given point, similarly in a rule of law court, from the group of action sequences the one with "majority rules" is the one that becomes the final action sequence. From the phenomena observed in the experiment, every single action sequence's accuracy is around 50%, but by applying majority rules the accuracy of the final action sequence can increase to 60% or even 80%.

Majority of traditional methods tend to focus on breaking down forecasting into three steps: (1) find a trend, (2) apply an interval cycle, and (3) eliminate external noise (uncertainty) as much as possible. If the said dataset can't be broken down into these three parts, then that dataset is treated as an unpredictable data series filled with random external noise that can't be reduced. Basically, traditional methods strive to reduce or completely eliminate external noise (uncertainty) by treating it as a bad thing to find a certain possible trend.

Compared to traditional methods, our methodology doesn't treat external noise (uncertainty) necessarily as a bad thing, we don't strive to reduce or eliminate noise and uncertainty but quite the contrary, we attempt to utilize uncertainty to find valuable information from the constantly changing environment to formulate a trend. Thus, we don't set out with the mindset of attempting to break up the data into trend, interval cycle, and noise, instead we seek to embrace uncertainty as a factor to help us find possible outcomes; based on reward learning the operation matrix with the maximum returns formulates a way to "randomly play dice" to "counter" the random walk of time series' trajectories.

5. Conclusion

In this paper, the quantum-like evolutionary algorithm utilizes an operation matrix to generate a group of simulated virtual trajectories, then computes the returns of each virtual trajectory compared to the actual trajectory of the observed time series, and finally genetic programming is applied to evolve the operation matrix with the maximum returns for forecasting. After applying majority rules, the forecast accuracy was increased based on the experiment results. MAPE, RMSE, and accuracy odds were used to evaluate the accuracy of the forecast results. The accuracy based on the evaluation metrics of MAPE and RMSE were both 90%+ and the accuracy odds were 60%+.

Further research work will include: increase the training time and adjust the parameters of the GP algorithm (crossover, mutation, selection) to see if forecast accuracy will improve; conduct more training and forecasts on the vast amounts of other time series datasets; a comparison to traditional time series forecasting methods and Machine Learning methods as well as other time series tools.

Because the future is inherently unpredictable, a perfect universally all-encompassing crystal ball method can't be found to "predict the future", thus if a black swan suddenly alighted then no one can foresee it no matter what. In this paper we deliberately chose a small sample of historical data to forecast a short horizon, assuming the time series will "keep" its "relative" trend of the recent past in the near future. If a black swan does actually show up then our method is not omnipotent and fail-proof, it won't be able to predict the landing of a black swan either, which in that case we'll just have to "trust our gut" to throw the dice back and hope for the best.

Data Availability
The data that was used for the purposes of this study are publicly available for download from the Federal Reserve Economic Data website. The authors confirm that the data supporting the findings of this study are available within the main manuscript. All data that support the findings of this study are also available from the corresponding author, K.X., upon reasonable request.
Author Contributions
All authors conducted the research and contributed to the development of the model. L.X. contributed to the research from the aspects of machine learning, decision theory, and quantum theory. K.X. prepared the data, did data analysis, and wrote the main manuscript. All authors reviewed the manuscript.
Conflicts of Interests
The authors are affiliated with XINVISIONQ, INC., the developer of a commercial forecasting tool, but for transparency of the experiments conducted in this study, all data generated and produced are included in the paper's main text. The results and conclusions presented in this paper are in accordance with the ethical requirements of academic research as no data was manipulated or manually adjusted. The authors have no other competing academic interests to declare, as the methodology formulated and findings reached in this paper are the authors' original research work.
Funding
This work received no funding.

References

[1] Andersen TG, Bollerslev T, Christoffersen PF, Diebold FX. 2005 Volatility forecasting. Natl bur. Econ. Res. 11188. (doi:10.3386/w1188)

[2] Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, Jari Porras, 2022 Machine learning techniques and data for stock market forecasting: A literature review, Expert Systems with Applications, Volume 197, 116659. (doi:10.1016/j.eswa.2022.116659)

[3] Chopra, R.; Sharma, G.D. 2021 Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. J. Risk Financial Manag. 14, 526. (doi:10.3390/jrfm14110526)

[4] Huang H, Chen M, Qiao X. 2024 Generative Learning for Financial Time Series with Irregular and Scale-Invariant Patterns. The Twelfth International Conference on Learning Representations.

[5] Ayitey Junior, M., Appiahene, P., Appiah, O. et al. 2023 Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis. J Big Data 10, 9. (doi:10.1186/s40537-022-00676-2)

[6] Lina Ni, Yujie Li, Xiao Wang, Jinquan Zhang, Jiguo Yu, Chengming Qi. 2019 Forecasting of Forex Time Series Data Based on Deep Learning. Procedia Computer Science, 147, 647-652. (doi:10.1016/j.procs.2019.01.189)

[7] Murray, K.; Rossi, A.; Carraro, D.; Visentin, A. 2023 On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles. Forecasting 5, 196-209. (doi:10.3390/forecast5010010)

[8] Chevallier, J.; Guégan, D.; Goutte, S. 2021 Is It Possible to Forecast the Price of Bitcoin? Forecasting 3, 377-420. (doi:10.3390/forecast3020024)

[9] Mingxi Liu, Guowen Li, Jianping Li, Xiaoqian Zhu, Yinhong Yao. 2021 Forecasting the price of Bitcoin using deep learning, Finance Research Letters, 40, 101755. (doi:10.1016/j.frl.2020.101755)

[10] Jiyang Cheng, Sunil Tiwari, Djebbouri Khaled, Mandeep Mahendru, Umer Shahzad. 2024 Forecasting Bitcoin prices using artificial intelligence: Combination of ML, SARIMA, and Facebook Prophet models, Technological Forecasting and Social Change, 198, 122938. (doi:10.1016/j.techfore.2023.122938)

[11] Leite Coelho da Silva, F., da Silva Cordeiro, J., da Costa, K. et al. 2025 Time series forecasting via integrating a filtering method: an application to electricity consumption. Comput Stat. (doi:10.1007/s00180-024-01595-x)

[12] Qureshi, M., Arbab, M.A. & Rehman, S. 2024 Deep learning-based forecasting of electricity consumption. Sci Rep 14, 6489. (doi:10.1038/s41598-024-56602-4)

[13] Xuerong Li, Yiqiang Zhong, Wei Shang, Xun Zhang, Baoguo Shan, Xiang Wang. 2022 Total electricity consumption forecasting based on Transformer time series models, Procedia Computer Science, Volume 214, 312-320, (doi:10.1016/j.procs.2022.11.180)

[14] Ali Hussein, Mohammed Awad. 2024 Time series forecasting of electricity consumption using hybrid model of recurrent neural networks and genetic algorithms, Measurement: Energy, Volume 2, 100004, (doi:10.1016/j.meaene.2024.100004)

[15] Böse JH, Flunkert V, Gasthaus J, Januschowski T, Lange D, Salinas D, Schelter S, Seeger M, Wang Y. 2017 Probabilistic demand

forecasting at scale. Proc. VLDB Endow. 10, 1694–1705. (doi:10.14778/3137765.3137775)

[16] Giri, C.; Chen, Y. 2022 Deep Learning for Demand Forecasting in the Fashion and Apparel Retail Industry. Forecasting, 4, 565-581. (doi:10.3390/forecast4020031)

[17] Menculini, L.; Marini, A.; Proietti, M.; Garinei, A.; Bozza, A.; Moretti, C.; Marconi, M. 2021 Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices. Forecasting, 3, 644-662. (doi:10.3390/forecast3030040)

[18] Baierle, I.C.; Haupt, L.; Furtado, J.C.; Pinheiro, E.T.; Sellitto, M.A. 2024 Forecasting Raw Material Yield in the Tanning Industry: A Machine Learning Approach. Forecasting, 6, 1078-1097. (doi:10.3390/forecast6040054)

[19] Stoffer DS, Ombao H. 2012 Editorial: special issue on time series analysis in the biological sciences. J. Time Ser. Anal. 33, 701–703. (doi:10.1111/j.1467-9892.2012.00805.x)

[20] Topol EJ. 2019 High-performance medicine: the convergence of human and artificial intelligence. Nat. Med. 25, 44–56. (doi:10.1038/s41591-018-0300-7)

[21] Tropsha, A., Isayev, O., Varnek, A. et al. 2024 Integrating QSAR modelling and deep learning in drug discovery: the emergence of deep QSAR. Nat Rev Drug Discov 23, 141–155. (doi:10.1038/s41573-023-00832-0)

[22] Golbraikh, A., Wang, X.S., Zhu, H., Tropsha, A. 2016 Predictive QSAR Modeling: Methods and Applications in Drug Discovery and Chemical Risk Assessment. Leszczynski, J. (eds) Handbook of Computational Chemistry. Springer, Dordrecht. (doi:10.1007/978-94-007-6169-8_37-3)

[23] Mudelsee M. 2019 Trend analysis of climate time series: a review of methods. Earth Sci. Rev. 190, 310–322. (doi:10.1016/j.earscirev.2018.12.005)

[24] Mansfield, L.A., Nowack, P.J., Kasoar, M. et al. 2020 Predicting global patterns of long-term climate change from short-term simulations using machine learning. npj Clim Atmos Sci 3, 44. (doi:10.1038/s41612-020-00148-5)

[25] Bochenek, B.; Ustrnul, Z. 2022 Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives. Atmosphere, 13, 180. (doi:10.3390/atmos13020180)

[26] Chen, L.; Han, B.; Wang, X.; Zhao, J.; Yang, W.; Yang, Z. 2023 Machine Learning Methods in Weather and Climate Applications: A Survey. Appl. Sci., 13, 12019. (doi:10.3390/app132112019)

[27] Eyring, V., Collins, W.D., Gentine, P. et al. 2024 Pushing the frontiers in climate modelling and analysis with machine learning. Nat. Clim. Chang. 14, 916–928. (doi:10.1038/s41558-024-02095-y)

[28] Tang Y, Zhou J, Pan X, Gong Z, Liang J. 2024 PostRainBench: A Comprehensive Benchmark and a New Model for Precipitation Forecasting. (https://arxiv.org/abs/2310.02676)

[29] Samia MELLAH, Youssef TRARDI, Guillaume GRATON, Bouchra ANANOU, El Mostafa EL ADEL, Mustapha OULADSINE. 2022 Semiconductor Multivariate Time-Series Anomaly Classification based on Machine Learning Ensemble Techniques*, IFAC-PapersOnLine, 55, 6, 476-481, (doi:10.1016/j.ifacol.2022.07.174)

[30] Pendyala V K, Sarjoughian H S, Potineni B S, Yellig E J. 2024 A Benchmark Time Series Dataset for Semiconductor Fabrication Manufacturing Constructed using Component-based Discrete-Event Simulation Models. (https://arxiv.org/abs/2408.09307)

[31] Lee S, Choi J, Kim M S. 2024 Generative Pre-Training of Time-Series Data for Unsupervised Fault Detection in Semiconductor Manufacturing. (https://arxiv.org/abs/2309.11427)

[32] R. Hwang, S. Park, Y. Bin and H. J. Hwang. 2023 Anomaly Detection in Time Series Data and its Application to Semiconductor Manufacturing. IEEE Access, 11, 130483-130490. (doi: 10.1109/ACCESS.2023.3333247)

[33] Lv C, Wang Y, Han D, Zheng X, Huang X, Li D. 2024 Efficient and Effective Time-Series Forecasting with Spiking Neural Networks. Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[34] Wu S, Wang Y. 2023 Applying Neural Manifold Constraint on Point Process Model for Neural Spike Prediction. 45th Annual International Conference of the IEEE EMBC (doi: 10.1109/EMBC40787.2023.10340489)

[35] S. Wu, X. Zhang and Y. Wang, 2024 Neural Manifold Constraint for Spike Prediction Models Under Behavioral Reinforcement, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 32, 2772-2781. (doi: 10.1109/TNSRE.2024.3435568)

[36] Mestre G, Barbulescu R,Oliveira A L, Silveira M L. 2021 Modelling Neuronal Behaviour with Time Series Regression: Recurrent Neural Networks on C. Elegans Data. (https://arxiv.org/abs/2107.06762)

[37] Atanas AA, Kim J, Wang Z, Bueno E, Becker M, Kang D, Park J, Kramer TS, Wan FK, Baskoylu S, Dag U, Kalogeropoulou E, Gomes MA, Estrem C, Cohen N, Mansinghka VK, Flavell SW. 2023 Brain-wide representations of behavior spanning multiple timescales and states in C. elegans. Cell. 186(19):4134-4151.e31. (doi: 10.1016/j.cell.2023.07.035)

[38] U. Sree Katikala and R. Ranjith. 2023 Detecting Anomaly in ECG Signal Using Time Series Forecast Approach. ICCCIS, 323-328, (doi:

10.1109/ICCCIS60361.2023.10425516)

[39] Zacarias, H.; Marques, J.A.L.; Felizardo, V.; Pourvahab, M.; Garcia, N.M. 2024 ECG Forecasting System Based on Long Short-Term Memory. Bioengineering, 11, 89. (doi:10.3390/bioengineering11010089)

[40] M. Ali, P. W. Moore, T. Barkouki and L. J. Brattain. 2024 Zero-Shot Forecasting for ECG Time Series Data Using Generative Foundation Models," IEEE 20th BSN, 1-4, (doi:10.1109/BSN63547.2024.10780646)

[41] J. Patil, X. Wu and Z. Rezki. 2023 A Novel CNN-Informer Model for Electrocardiogram Time Series Forecasting. IEEE MIT URTC, 1-5. (doi: 10.1109/URTC60662.2023.10534983)

[42] Beroza, G.C., Segou, M. & Mostafa Mousavi, S. 2021 Machine learning and earthquake forecasting—next steps. Nat Commun 12, 4761. (doi:10.1038/s41467-021-24952-6)

[43] Jafari, A.; Fox, G.; Rundle, J.B.; Donnellan, A.; Ludwig, L.G. 2024 Time Series Foundation Models and Deep Learning Architectures for Earthquake Temporal and Spatial Nowcasting. GeoHazards, 5, 1247-1274. (doi:10.3390/geohazards5040059)

[44] S. LÖK and M. Karabatak. 2021 Earthquake Prediction by Using Time Series Analysis. 9th ISDFS, pp. 1-6, (doi: 10.1109/ISDFS52919.2021.9486358)

[45] D. Shah, J. Shah, V. Parwanda, S. Vyas and S. Mishra. 2024 Earthquake Forecasting using Time Series Data. 3rd ICONAT, pp. 1-7, (doi:10.1109/ICONAT61936.2024.10774958)

[46] Edward Roome, David Christie, Simon Neill. 2024 Predicting coastal wave conditions: A simple machine learning approach, Applied Ocean Research, 153, 104282. (doi:10.1016/j.apor.2024.104282)

[47] Scott C. James, Yushan Zhang, Fearghal O'Donncha. 2018 A machine learning framework to forecast wave conditions, Coastal Engineering, 137, 1-10, (doi:10.1016/j.coastaleng.2018.03.004)

[48] Yi Yin, Pengjian Shang. 2016 Forecasting traffic time series with multivariate predicting method, Applied Mathematics and Computation, 291, 266-278. (doi:10.1016/j.amc.2016.07.017)

[49] L. Wang, J. Chen, W. Wang, R. Song, Z. Zhang and G. Yang. 2022 Review of Time Series Traffic Forecasting Methods, 4th ICCR, 1-5. (doi: 10.1109/ICCR55715.2022.10053870)

[50] Jan G. De Gooijer, Rob J. Hyndman, 25 years of time series forecasting, International Journal of Forecasting, Volume 22, Issue 3, 2006, Pages 443-473.

[51] Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control. San Francisco: Holden Day (revised ed. 1976).

[52] Borkin, D., Nemeth, M., Nemethova, A. 2019 Using Autoregressive Integrated Moving Average (ARIMA) for Prediction of Time Series Data. Advances in Intelligent Systems and Computing, 1046. Springer, Cham. (10.1007/978-3-030-30329-7_42)

[53] Gardner Jr., E. S. (1985). Exponential smoothing: The state of the art. Journal of Forecasting, 4, 1–38.

[54] Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted averages. O.N.R. Memorandum 52/1957, Carnegie Institute of Technology. Reprinted with discussion in 2004. International Journal of Forecasting, 20, 5–13.

[55] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6, 324–342.

[56] Harvey AC. 1990 Forecasting, structural time series models and the kalman filter. Cambridge, UK: Cambridge University Press.

[57] Zöller, MA., Lindauer, M., Huber, M.F. 2025 Auto-sktime: Automated Time Series Forecasting. In: Festa, P., Ferone, D., Pastore, T., Pisacane, O. (eds) Learning and Intelligent Optimization. LION 2024. Lecture Notes in Computer Science, vol 14990. Springer, Cham. (doi:10.1007/978-3-031-75623-8_35)

[58] Meisenbacher, S., Turowski, M., Phipps, K., Rätz, M., Müller, D., Hagenmeyer, V., & Mikut, R. 2022 Review of automated time series forecasting pipelines. WIREs Data Mining and Knowledge Discovery, 12(6), e1475. (doi:10.1002/widm.1475)

[59] Westergaard, G.; Erden, U.; Mateo, O.A.; Lampo, S.M.; Akinci, T.C.; Topsakal, O. 2024 Time Series Forecasting Utilizing Automated Machine Learning (AutoML): A Comparative Analysis Study on Diverse Datasets. Information, 15, 39. (doi:10.3390/info15010039)

[60] Alsharef, A., Aggarwal, K., Sonia et al. Review of ML and AutoML Solutions to Forecast Time-Series Data. 2022 Arch Computat Methods Eng 29, 5297–5311. (doi:10.1007/s11831-022-09765-0)

[61] M. Peixeiro, Time Series Forecasting in Python, Manning, New York, 2022.

[62] F. Lazzeri, Machine Learning for Time Series Forecasting with Python, Wiley, New Jersey, 2020.

[63] Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27(3), 1–22.

[64] Faniband Y P, Ishak I, Sait S. 2022 A review of open source software tools for time series analysis. (https://arxiv.org/pdf/2203.05195)

[65] Hyndman R. 2020 A brief history of forecasting competitions. Int. J. Forecast. 36, 7–14. (doi:10.1016/j.ijforecast.2019.03.015)

[66] 2020 The M4 Competition: 100000 time series and 61 forecasting methods. Int. J. Forecast. 36, 54–74. (doi:10.1016/j.ijforecast.2019.04.014)

[67] Makridakis S, Hibon M. 2000 The M3-competition: results, conclusions and implications. Int. J. Forecast. 16, 451–476. The M3-Competition. (doi:10.1016/S0169-2070(00)00057-1)

[68] FryC, Brundage M. 2019 The M4 forecasting competition–a practitioner's view. Int. J. Forecast. 36, 156–160. (doi:10.1016/j.ijforecast.2019.02.013)

[69] Lim, B, Zohren S. 2021 Time-series forecasting with deep learning: a survey. Phil. Trans. R. Soc. A 379: 20200209. (doi:10.1098/rsta.2020.0209)

[70] Casolaro, A.; Capone, V.; Iannuzzo, G.; Camastra, F. 2023 Deep Learning for Time Series Forecasting: Advances and Open Problems. Information, 14, 598. (doi:10.3390/info14110598)

[71] Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. 2010 An empirical comparison of machine learning models for time series forecasting. Econ. Rev. 29, 594–621. (doi:10.1080/07474938.2010.481556)

[72] Makridakis S, Spiliotis E, Assimakopoulos V. 2018 Statistical and machine learning forecasting methods: concerns and ways forward. PLoS ONE 13, 1–26. (doi:10.1371/journal.pone.0194889)

[73] David Salinas, Valentin Flunkert, Jan Gasthaus, Tim Januschowski. 2020 DeepAR: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting, 36, 3, 1181-1191. (doi: 10.1016/j.ijforecast.2019.07.001)

[74] Tim Januschowski, Yuyang Wang, Jan Gasthaus, Syama Rangapuram, Caner Türkmen, Jasper Zschiegner, Lorenzo Stella, Michael Bohlke-Schneider, Danielle Maddix, Konstantinos Benidis, Alexander Alexandrov, Christos Faloutsos, and Sebastian Schelter. 2024 A Flexible Forecasting Stack. Proc. VLDB Endow. 17, 12, 3883–3892. (doi:10.14778/3685800.3685813)

[75] Taylor SJ, Letham B. 2017 Forecasting at scale. PeerJ Preprints 5:e3190v2 (doi:10.7287/peerj.preprints.3190v2)

[76] Garza A, Challu C, Mergenthaler-Canseco M. 2024 TimeGPT-1. (https://arxiv.org/pdf/2310.03589)

[77] Xin, L.; Xin, H. 2023 Decision-making under uncertainty—A quantum value operator approach. Int. J. Theor. Phys., 62, 48. (doi:10.1007/s10773-023-05308-w)

[78] Xin, L.; Xin, K.; Xin, H. 2023 On Laws of Thought—A Quantum-like Machine Learning Approach. Entropy, 25, 1213. (doi:10.3390/e25081213)

[79] Feynman, Richard P. The Feynman Lectures on Physics. Volume III, Quantum Mechanics. Basic Books, 2015.

[80] Dirac, P.A.M. The Principles of Quantum Mechanics; Oxford University Press: Oxford, UK, 1958.

[81] Trimmer, John D. "The Present Situation in Quantum Mechanics: A Translation of Schrödinger's 'Cat Paradox' Paper." Proceedings of the American Philosophical Society 124, no. 5 (1980): 323–38.

[82] N. S. Yanofsky, M. A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press, Cambridge, 2008.

[83] C. Rovelli, Helgoland: Making Sense of the Quantum Revolution, Riverhead Books, New York, NY, USA, 2022.

[84] Von Neumann, J. Mathematical Foundations of Quantum Theory; Princeton University Press: Princeton, NJ, USA, 1932.

[85] Holland, J. Adaptation in Natural and Artificial System; University of Michigan Press: Ann Arbor, MI, USA, 1975.

[86] Koza, J.R. Genetic Programming, on the Programming of Computers by Means of Natural Selection; MIT Press: Cambridge, MA, USA, 1992.

[87] Koza, J.R. Genetic Programming II, Automatic Discovery of Reusable Programs; MIT Press: Cambridge, MA, USA, 1994.

[88] Darwin, Charles, 1809-1882. On the Origin of Species by Means of Natural Selection, or Preservation of Favoured Races in the Struggle for Life. London: John Murray, 1859.

[89] Von Neumann, J.; Morgenstern, O. Theory of Games and Economic Behavior; Princeton University Press: Princeton, NJ, USA, 1944.