Enabling An Informed Contextual Multi-Armed Bandit Framework For Stock Trading With Neuroevolution

Devroop Kar
dk7405@g.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Zimeng Lyu
zimenglyu@mail.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Alexander G. Ororbia
ago@cs.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Travis Desell
tjdvse@g.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Daniel Krutz
dxkvse@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

ABSTRACT

Multi-armed bandits and contextual multi-armed bandits have demonstrated their proficiency in a variety of application areas. However, these models are highly susceptible to volatility and often exhibit knowledge gaps due to a limited understanding of future states. In this paper, we propose a new bandit framework for what we refer to as *informed contextual multi armed bandits* (iCMABs) to mitigate these gaps, facilitating "informed" decisions based on predicted future contexts. The performance of an iCMAB is thus highly dependent on the accuracy of the forecast it uses. We examine the use of recurrent neural networks (RNNs) evolved through the EXAMM neuroevolution algorithm as compared to other time series forecasting (TSF) methods and evaluate our iCMAB framework's ability to make stock market trading decisions for the Dow-Jones Index (DJI) in comparison to other decision making strategies using these forecasts. Our results demonstrate that an iCMAB, driven by evolved RNN architectures, performs better than statistical TSF methods, fixed architecture RNNs for TSF, and other CMAB methods. Using evolved RNNs, iCMAB is able to achieve the highest return of over 21%, a ~7% improvement over not incorporating forecasted values, and a ~5% improvement over DJI's return for that time period.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Markov decision processes**; • **Applied computing** → **Economics**.

KEYWORDS

Multi-Armed Bandits, Recurrent Neural Networks, Decision Making

ACM Reference Format:
Devroop Kar, Zimeng Lyu, Alexander G. Ororbia, Travis Desell, and Daniel Krutz. 2024. Enabling An Informed Contextual Multi-Armed Bandit Framework For Stock Trading With Neuroevolution. In *Genetic and Evolutionary Computation Conference (GECCO '24 Companion), July 14–18, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3638530.3664145

GECCO '24 Companion, July 14–18, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0495-6/24/07.
https://doi.org/10.1145/3638530.3664145

1 INTRODUCTION

Multi-armed bandits address a fundamental problem in sequential decision-making where an agent must repeatedly choose actions from a set of available options – often referred to as "arms" – in order to maximize cumulative rewards over time. Each action yields an immediate but uncertain reward and the goal is to balance the trade-off between exploiting actions that have yielded high rewards in the past (exploitation) with exploring other actions to learn more about their potential rewards (exploration). The challenge lies in making optimal decisions with incomplete information about the rewards associated with each action, which requires strategies that dynamically adjust based on observed outcomes in order to efficiently allocate actions and to maximize overall rewards. Multi-armed bandit (MAB) algorithms have widespread applications in various fields, including online advertising, clinical trials, and recommendation systems, where efficient and adaptive decision-making is essential in the face of uncertainty [66, 68].

In this work, we propose the *informed Contextual Multi-Armed Bandits* (iCMAB) framework, which represents an evolution over traditional MAB models by integrating machine learning to supplement contextual information into the decision-making process. Our novel iCMAB approach enables systems to proactively make predictions regarding the anticipated context that is provided to the multi-armed bandit. Specifically, our iCMAB process will: **I)** jointly adopt a reward prediction model and generative world model of contexts, **II)** provide a measure of confidence in its predictions of both reward and context values, and **III)** utilize artificial neural network (ANN) models based on evolving recurrent neural networks (eRNNs) to predict both contextual and reward information. This updated information will serve as input to the intelligent systems that are driven by (contextual) bandits, enabling them to make informed decisions over time.

Our iCMAB framework particularly addresses the limitations of traditional MAB and contextual multi-armed bandit (CMAB) approaches in that these existing frameworks: **I)** are unable to sufficiently account for future environmental states and volatility, **II)** often exhibit knowledge gaps that inhibit their ability to make high-quality decisions, **III)** frequently encounter (in real-world scenarios) corrupt contextual and reward values that inhibit their decision-making, and **IV)** are frequently unable to observe the reward and thus cannot use this in later steps in the subsequent decision-making process. In this work, we tackle these issues by proposing the iCMAB framework, which will: 1) jointly adopt a reward prediction model and generative world model of contexts, and 2) utilize recurrent neural network (RNN) models for time series forecasting (TSF) that jointly predict both contextual and reward information, the design of which has been automatically generated using neuroevolution.

To validate our proposed iCMAB framework, this work demonstrates its effectiveness using stock trading as an application. Stock trading provides an interesting use case for iCMAB as it involves both forecasting and decision making – if the future value of a set of stocks is known, better decisions can be made about which stocks to buy, short or hold and how much of each to do so with. We hypothesize that more accurate forecasts will lead to being able to train better decision making strategies. In this problem context, we compare three leading iCMAB models driven by six different context prediction approaches against their CMAB counterparts. The proposed work makes the following contributions:

- **iCMAB introduction and evaluation:** We introduce and perform an evaluation of our novel *informed contextual multi-armed bandit* (iCMAB) framework;
- **Research Findings #1:** Our findings suggest that having a model able to predict future contexts improves the decisions made by iCMAB as compared to a traditional CMAB framing, resulting in the attainment of higher cumulative rewards.
- **Research Findings #2:** While using statistical approaches and fixed architecture RNNs to estimate future context performs better than the baseline, utilizing RNNs designed by neuroevolution provide the best and most robust results.

2 RELATED WORK

Contextual multi-armed bandits [48] have benefited various domains, facilitating improved sequential decision making. Domains include recommendation systems, healthcare, finance and dialogue systems. In Li *et al.* [41], the authors used a contextual bandit algorithm, termed LinUCB, to improve news article recommendation. It was suggested in Dimakopoulou *et al.* [20] that the integration of balancing methods from the causal inference literature would make contextual bandits more robust to bias due to improper reward estimation. In contrast to these prior efforts, our suggested iCMAB model makes use of a separate forecasting method for reward estimation, which in turn, as we will demonstrate, is capable of filtering erroneous reward feedback.

Recommendation systems based on bandits have been explored extensively. Wang *et al.* [69] suggested a factorization-based bandit to mitigate the issue where a bandit remains too focused on overly exploiting a learned model that is biased towards previously frequently recommended items; this ultimately hinders its ability to explore recommending newer items. Xu *et al.* [70] considered the situation where user interests gradually change over time and developed two models to tackle such a case. Gentile *et al.* [29] used a cluster of bandits to estimate user similarity and share feedback. While the focus of the iCMAB framework does not pertain to user preferences specifically, it relies on uncertainty estimation and generating confidence bounds to help drive its temporal generative modeling mechanisms.

MAB/CMAB-based approaches have also been used for addressing risk [28, 36–38, 51, 63]. Risk-awareness emphasizes various characteristics of the reward distribution, such as diversifying the impact of adverse outcomes [21]. In many real-world situations, maximizing the expected reward is not always the most desirable operation. For example, some portfolio managers may prefer portfolios with lower expected returns rather than risky portfolios with a higher accepted return [44] or less risk may be desired in specific mission or safety-critical CPS [2, 8, 57]. Operations (arms) with less variance may be desirable, therefore the risk of the reward should be considered during the decision-making process.

Apart from this, researchers have also looked into bandit operations where the environment may be less than ideal [11, 65, 73]. In a case where the reward may be missing, Bouneffouf *et al.* [12] utilized an unsupervised clustering approach to estimate the reward. Bouneffouf *et al.* [10] also proposed a methodology combining a traditional MAB with a CMAB to mitigate the effects of corrupted context. This differs from our approach where we utilize a (temporal generative) model trained on time-varying data to estimate contexts and rewards in cases of corruption and/or delay. Gajane *et al.* [27] examined a variation of the MAB problem where corrupt rewards were encountered. The objective of this work was to maximize the sum of the unobserved rewards through the transformed observations of these rewards using a stochastic corruption process with known parameters. Unlike this work, our framework fully considers the contextual case (and not the classical reward-only setup).

In this work, we apply and evaluate our proposed framework in the context of stock trading, as this problem provides a useful testing benchmark for determining the efficacy of bandit-driven approaches and TSF methods. Historically, several methods have been proposed in economics and in computer science to predict future market trends concerning stock trend direction (up or down, *i.e.,* bull market or bear market, respectively), intraday or interday stock price, associated risk and return, and so on. Usually, stock market data is represented in the form of time-series of data points collected at specific time intervals. Traditionally, statistical methods based on econometrics have been suggested in the literature [3, 13, 25] to process such data. Gradually, computational intelligence-based techniques [14, 26, 33, 67] have garnered attention for their ability to derive useful predictions quickly and efficiently. Nevertheless, these methods have their limitations – statistical methods are dependent on particular base assumptions while (many) machine learning approaches typically require hand-crafted features, suffer from limited interpretability (in terms of actual dependant factors), and tend to exhibit forms of overfitting.

The above issues have led to the design and development of neural network (NN)-based deep learning methods to enhance stock market predictions [46, 56]. Among them, recurrent neural networks (RNNs) [22, 61] have, more recently, garnered attention given that they are inherently designed to handle time series and sequential data. Important variants include the LSTM [34] and GRU [15], which have been used in stock prediction [7, 30, 40, 64]. Notably, deep reinforcement learning techniques have been also used to study stock price variations [45, 71, 72]. Methods which automate the design of stock forecasts have also been investigated, such as using genetic programming for feature selection [42] and graph based genetic programming to design equations for predicting index and currency values [39, 60, 74].

Neuroevolution systems are gradually being adopted in recent years for use in the financial markets. Qiu *et al.* [59] employed a hybrid approach using an ANN together with Genetic Algorithm (GA) to calculate the initial network weights to predict the return of the Japanese Nikkei 255 index for the next month. The prediction accuracy were significantly better than traditional backpropagation. Nadkarni and Neves [53] proposed a methodology combining principal component analysis (PCA) with the NeuroEvolution of Augmenting Topologies (NEAT) to generate a trading signal of potential high returns and daily profits with low associated risk.

Neural Architecture Search (NAS) has become popular to find optimal network architectures. Li *et al.* [43] proposed a decomposition-based memetic neural architecture search algorithm for univariate time series forecasting. The authors addressed the problem of time series information impairment when it is decomposed for pattern identification and optimized network configurations in huge search spaces. Hafiz *et al.* [31] proposed a co-evolution approach to concurrently select relevant features and topology of neural networks. They used a search framework consisting of MOEA and a posteriori decision support tool to identify neural architectures using the proposed method and evaluated their performance on NASDAQ, NYSE and S&P500 stock indices during the COVID-19 pandemic period.

3 TIME SERIES FORECASTING METHODOLOGIES

3.1 Statistical Models: ARIMA and VAR

Autoregressive integrated moving average (ARIMA) is a univariate statistical model that is used for time series forecasting. ***Vector autoregression (VAR)*** is a statistical model used to capture the relationship between multiple quantities as they change over time and is a type of stochastic process model. VAR models generalize the single-variable (univariate) autoregressive model by operating on multivariate time series patterns. Notably, VAR models are often used in economics and the natural sciences. Like the autoregressive part of ARIMA, each variable in VAR has an equation modeling its evolution over time. This equation includes the variable's lagged (past) values, the lagged values of the other variables in the model, and an error term.

3.2 Recurrent Neural Networks

Recurrent neural networks can capture both spatial and temporal relationships or patterns in data. These networks unfold across time and serve as good forecasters as well as sequence generators. Recurrent neural networks (RNNs) have played an important role in deep learning; these models are particularly effective when modeling sequential data given that they are able to operate over varying input lengths [22, 35, 52, 61]. However, in practice, RNNs are difficult to train as they suffer from the problem vanishing and exploding gradients [9, 58]. This makes it challenging to identify and learn the more complex temporal relationships within the data. To mitigate this issue, several different types of architecture and memory cells have been designed and introduced. These include orthogonal/unitary RNNs [4, 32], Δ-recurrent cells [55], as well as gating mechanisms such as long short-term memory (LSTM) [34] and gated recurrent units (GRUs) [15]. For this work we investigate using the popular fixed LSTM and GRU architectures, which were fully connected two layer architectures (a common design).

3.3 Evolution with EXAMM

In this work we utilize the Evolutionary eXploration of Augmenting Memory Models (EXAMM) [54] process, a neuroevolution algorithm that designs recurrent neural networks for TSF. As opposed to other forms of neural architecture search (NAS) and neuroevolution (NE) strategies, EXAMM was designed specifically for TSF and has demonstrated significant success in the problem area. EXAMM evolves progressively larger RNNs for large-scale, multivariate, real-world TSF problems [23, 24]. The evolved RNN architectures can consist of varying recurrent connections, simple neurons and memory cells. Recurrence/temporal reach can span multiple time steps, which was found to significantly improve predictive ability [19]. Memory cells are selected from a library of Δ-RNN units [55], gated recurrent units (GRUs) [17], long short-term memory cells (LSTMs) [34], minimal gated units (MGUs) [75], and update-gate RNN cells (UGRNNs) [18].

EXAMM evolves and trains RNNs concurrently, using a naturally load balancing, asynchronous, steady state distributed algorithm, which further decouples the population size from the number of worker processes or threads available [54]. It also uses island-based populations, which periodically perform extinction events to remove and repopulate the worst performing island(s) to further improve performance [50]. Offspring generated by crossover or mutation inherit (parameter) weights from their parent(s) using a Lamarckian approach, which reduces the amount of backpropagation (backprop) epochs required for training the evolved RNNs; this significantly reduces the time needed for individual model training and evaluation [49].

4 BANDIT FRAMEWORKS

4.1 Contextual Multi-Armed Bandit (CMAB)

Multi-armed bandits [5, 6] (MABs) have been widely used in decision-making scenarios across various domains, including but not limited to: internet recommendations, clinical trials / courses of treatments, and for anomaly detection. The primary function of the bandit is to balance exploration and exploitation while maximizing reward. In this regard, the bandit may have access to additional information about the environment (context), which might aid it in its decision-making process. Such bandits are known as *contextual multi-armed bandits* (CMABs) [48, 76].

Many algorithms have been proposed to improve the ability of bandits to balance exploration against exploitation while yielding optimal results. For this work, we have selected three popular algorithms/schemes:

- *LinUCB*: It models a linear relationship between the reward and the context making it straightforward to implement. It provides hyperparameters which facilitate tuning the degree of exploration versus exploitation as the bandit takes actions.

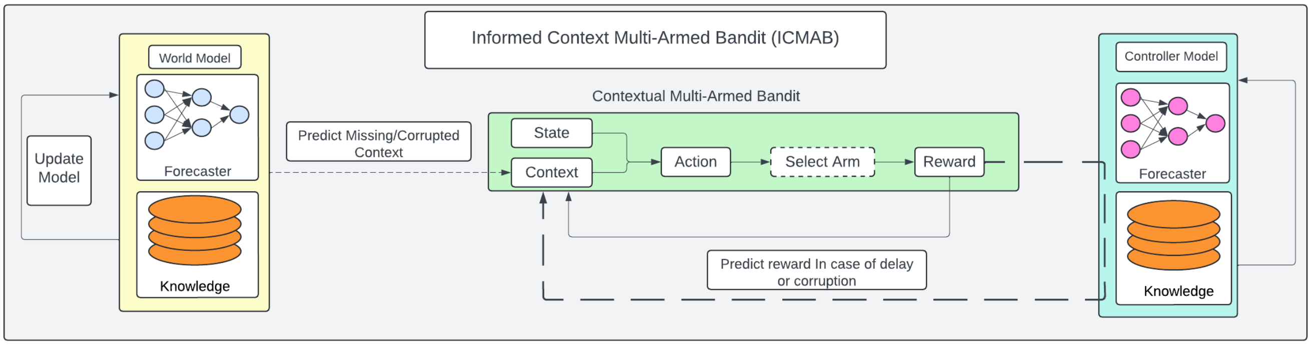
Informed Context Multi-Armed Bandit (ICMAB)
World Model
Forecaster
Knowledge
Update Model
Predict Missing/Corrupted Context
Contextual Multi-Armed Bandit
State
Context
Action
Select Arm
Reward
Predict reward In case of delay or corruption
Controller Model
Forecaster
Knowledge

Figure 1: iCMAB Process Flow

- *Thompson Sampling with Linear Payoffs/Returns*: This is a heuristic approach that has attained good empirical results on advertisements and news recommendations. Since it makes use of prior information, it is appropriate fits for time series data, given that we can incorporate forecasts as additional context.
- *EXP4.P*: This adversarial bandit scheme relies on probability vectors regarding how good an action might be for a particular context. This allows us to make the bandit select a subset of actions by tuning these probabilities. From a trading perspective, this is ideal for modeling the "buy, then sell" sequence.

4.1.1 ***LinUCB***. The LinUCB algorithm [16, 41] was formulated to extend the UCB algorithm [5] for considering contextual cases. Consider that each arm of the bandit has an associated feature vector $x_{t,a} \in R^d$. The underlying assumption of LinUCB is that the expected reward for an arm a is linearly related to its corresponding feature vector $x_{t,a} \in R^d$:

$$E[r_{t,a}|x_{t,a}] = x_{t,a}^T.\theta^\star \tag{1}$$

where $\theta^\star$ is the true coefficient vector.

4.1.2 ***Thompson Sampling with Linear Payoffs/Returns***. Agarwal *et al.* [1] proposed this contextual version of the Thompson sampling algorithm with linear payoffs. Consider a total of K arms, with each arm a associated with a d-dimensional feature vector $x_{t,a}$ at time t. A linear predictor is defined by a d-dimensional parameter $\mu \in R^d$ and predicts the mean reward of arm a by $\mu.x_{t,a}$.

$\hat{\mu} \in R^d$ is taken as a parameter such that the expected reward for arm a at time t is linearly related to the context *i.e.*,,

$$\hat{r_{t,a}} = \hat{\mu}.x_{t,a} \tag{2}$$

The real reward r_t associated with arm/action a at time t is assumed to be generated from a distribution with mean $\hat{r_{t,a}}$ (this distribution is unknown to the bandit and must learned from experience). If arm $a*$ is the optimal arm at time t and arm a is selected, then the regret for that action is $\hat{r_{t,a*}} - \hat{r_{t,a}}$. The likelihood of the reward $\hat{r_{t,a}}$, given the context $x_{t,a}$, is modeled as a Gaussian distribution $N(x_{t,a}^T\mu^*, v^2)$, where $v = R\sqrt{\frac{24}{\epsilon} d \ln \frac{1}{\delta}}$. Here $\epsilon \in (0, 1)$ and δ controls the regret bound.

4.1.3 ***EXP4.P***. This is an adversarial bandit algorithm that improves the regret-bound computation of EXP4 [6] by using aspects of both UCB [5] and EXP4. It computes the confidence interval of the reward vector estimator (hence, it bounds the cumulative reward of each expert with high probability) and then it designs a strategy to weight each expert.

Similar to the EXP4 algorithm setting, there are K arms $\{1, 2, ..., K\}$ and N experts $\{\xi_1, \xi_2, ..., \xi_N\}$. At time $t \in \{1, ..., T\}$, the world reveals context x_t and each expert i outputs an advice vector $\xi_{i,t}$, representing its recommendations on each arm. The agent then selects an arm a_t based on the advice and an adversary chooses a reward vector r_t. Finally, the world reveals the reward of the chosen arm r_{t,a_t}.

4.2 Informed Contextual Multi-Armed Bandits (iCMAB)

The motivation behind the iCMAB framework is to generalize the CMAB to a time-varying architecture that jointly adapts a model for deciding actions and estimating rewards/returns as well as a world model for CPS. Two key functions characterize iCMAB, namely:

- **Controller/Action Model**: The controller f_c is responsible for estimating the reward values μ_r for each action a_t, given the current context c_t and an encoding $E_t(.)$ of other signals that might help inform what action to take next.

$$f_c(c_t, a_t, E_t(.)) = \mu_r \tag{3}$$

- **World Model** - The world model f_w is responsible for forecasting the context c_t itself given the previously encountered context c_{t-1} and encoding $E_t(.)$.

$$f_r(c_{t-1}, E_t(.)) = c_t \tag{4}$$

The process flow of iCMAB has been depicted in Figure 1, including the functions of the world and action controller. To design the world model (the "forecaster"), we leverage the forecasting power of an RNN to drive the decision-making ability of a contextual bandit using estimated future data points as context. The role of the forecaster not only suggests possible future states, but it can also be leveraged to identify and replace possible corruptions or even missing contextual data. The trained forecaster can then communicate the updated context to the bandit, which will select an action to maximize the reward. The feedback loop nature of the architecture ensures that the forecaster models are updated based on the observations and ground truth, thus improving the predictions in subsequent iterations. The definition of reward will depend on the domain to which this framework is applied. For the scope of this paper, reward is assumed to be the profit or loss incurred during trading.

Unlike a traditional MAB, the iCMAB framework not only integrates contextual information into the decision-making process, but it also incorporates historical behaviors, to make more informed decisions. In addition, in contrast to the traditional CMAB, the iCMAB is able to account for future contexts and possible corruption in contexts and reward values.

Stock trading was selected as the problem context for evaluating the iCMAB because: **I)** there is a dependency between future returns on past trends, and **II)** reward information is readily available. This makes designing the controller model easier as the number of shares and the selling price of each share will, in turn, give the reward value for a specific company's stock. Making a trade decision "today" requires a prediction about how the market will behave "tomorrow"; the system's forecaster helps to estimate a possible return on the concerned stock, which drives the iCMAB to decide whether to `BUY, SELL` or `HOLD OFF` on that day.

In preliminary tests, an interesting scenario occurred when the bandit was allowed access to all of its arms (*i.e.*,, the entire set of possible actions). Since it could `BUY, SELL` or `HOLD OFF`, theoretically it could opt to simultaneously both `SELL` and `BUY` a stock. Additionally, in most cases, if the bandit encountered loss in the beginning stages of its training, it got stuck choosing `HOLD OFF` to ensure the net reward did not fall below zero. This led to a sub-optimal policy which did not contribute to achieving profits. To mitigate this issue, we modified the bandit to only have access to alternating subsets. Initially, the bandit could only either choose `BUY` or `HOLD OFF`. Once it had bought any stock, it would then be constrained to choose from the subset `SELL, HOLD OFF`. After a stock was sold, the options reverted back to `BUY` or `HOLD OFF`. A similar topic was investigated by [62] where the authors introduced an approach based on the `EXP3` algorithm; in effect, they re-distributed the probabilities of the entire action set among the probable actions at a particular time intervals.

In general, iCMAB differs from traditional reinforcement learning (RL) techniques primarily in their incorporation of contextual information during decision-making. Unlike traditional RL, where decisions are often based exclusively on historical experiences and feedback, iCMAB considers additional contextual features associated with each action. This explicit inclusion of context allows iCMAB to adapt its decision-making strategy, balancing exploration and exploitation by leveraging information about the environment. iCMAB algorithms are specifically designed to generalize knowledge across different contexts, making them particularly suitable for applications where decision-making depends on nuanced contextual factors. Conversely, traditional RL techniques typically lack this explicit consideration of context, leading to differences in their adaptability and performance in context-dependent scenarios.

5 EXPERIMENTS

Research Questions: This work, beyond developing the iCMAB framework, seeks to address the following research questions:

RQ1. Can a time series forecasting model improve the decision-making capacity of a CMAB? *Our research findings indicate that a suitable forecasting method does improve the decision-making ability of a CMAB beyond, and that better forecasts tend to produce better CMAB decisions.*

RQ2. How does the iCMAB compare to the CMAB in maximizing rewards? *Our research findings indicate that in the context of stock trading iCMAB using adversarial bandits perform extremely well in maximizing profits as compared to just the CMAB variant.*

RQ3. How does an evolved RNN network compare to other generative models such as VAR and other fixed architecture RNNs to account for future states and drive decision-making? *Our research findings indicate that using evolved RNNs for forecasting provides better context than ARIMA, VAR or fixed architecture RNNs.*

5.1 Details of Selected Data

The stock data used for this project comes from the Center for Research in Security Prices, LLC (CRSP) [47]. The most comprehensive collection of security price, return, and volume data for the NYSE, AMEX and NASDAQ stock markets is maintained by CRSP. From this, we used historical data of the 30 companies in the Dow-Jones Index (DJI). As the companies in the DJI are selected due to their prominance and well known ability to provide a good returns, simply buying an equal amount of each stock and holding it for a time period (as many equity firms do) provides a strong baseline to determine if our iCMAB strategy can "beat the market". Daily CRSP data for the DJI from January 1, 1992 to December 31, 2023 was extracted and divided into training, validation and testing datasets. We selected 7 economic predictors that are strongly associated with stock returns as input parameters for stock return prediction: *Stock Return*, *Volumn Change*, *Stock Price*, *Bid-Ask Spread*, *Illiquidity*, *DJI Index Return* and *S&P 500 Index Return.* The output parameter for time series forecasting was *Stock Return (RET).* This data was used to evaluate the performance of the iCMAB for stock trading, with respect to the different decision making (bandit) algorithms and time series forecasting methods.

5.2 Design and Evaluation

Given the DJI company data extracted from CRSP, we generated a training dataset using data from January 1, 1992 to December 31, 2021, a validation dataset from January 1, 2022 to December 31, 2022, and a testing dataset from January 1, 2023 to December 31, 2023. The time series forecasting methods were trained on the training dataset, and then used to provide forecasts for the validation and testing datasets. EXAMM required a validation dataset to determine fitness of evolved RNNs, which utilized the validation dataset. The best evolved RNN was used for forecasts (on the same validation dataset) to be provided to the decision making strategies and the testing dataset. The decision making strategies which required training were trained on the validation dataset along with the forecasts of the different TSF methods. TSF methods were trained separately for each stock and their forecasting results to train bandits for separately for each stock. Results were finally determined by evaluating the trained decision making strategies on the test dataset – which was not used in training either the TSF or bandit methods.

We used percent gains as our evaluation metric for the various methodology combinations, as we are concerned with the net profit across trades. An oracle was created which was responsible to determine amount of gain or loss incurred based on each trade. The initial pool for investment was set at $100 and based on sequential decisions, the return was calculated. The final performance of forecasting and decision making strategy combinations was determined by the amount of amount of money made (or lost) at the end of the test data period. As we were concerned with percentage return, fractional stock purchases and sells were allowed.

The following evaluation criteria was used in our analysis -

- **Ability to maximize profits:** The decision to buy, sell or hold was driven by the predicted return as estimated by the forecasting model. The profits and losses were calculated based on the share price on each day. As DJI is a well known index of well performing companies, the baseline strategy used was *buy and hold* where the initial funds were divided equally to buy stocks from each of the 30 companies in the DJI, and then held (no other buying or selling) for the entire test period. Performing better than this is "beating the market" and the overall goal of this work. This provided a suitable benchmark to compare how the combination of different forecasters and decision making strategies performed.
- **Ability to restrict actions based on past state:** Unlike a traditional bandit problem, in stock trading a bandit cannot utilize the entire action space at every timestep. For a particular stock, it is only possible to sell it if some shares are already owned. Similarly, a stock cannot be purchased unless others are sold or there are available funds from previous sales. While it is hard to evaluate such restrictions, we evaluate the policies generated by decision making strategies by tracking how the sequence of decisions lead to profit or loss.

We evaluated six different forecasting strategies. The first results in a CMABs as it did not use intelligent forecasts from trained models, and the latter result in iCMABs as they utilize trained models for forecasting. These strategies are:

(1) A simple strategy where we simply use the return at time t for a stock as the forecast for that stock at time $t + 1$, $RET(t + 1) = RET(t)$. While simple, this a challenging strategy to beat for time series forecasting of noisy real world data, as it is the optimal forecasting strategy when the data is a random walk. We denote this strategy *trivial-ret*.
(2) A VAR forecasting model. The observations from all attributes up to time t are used to predict $RET(t + 1)$.
(3) An ARIMA model using observations $0, \cdots, RET(t)$ to predict `RET(t+1)`.
(4) A fixed (un-evolved) fully connected two-layer LSTM RNN, trained for each stock, to predict the stock's return at the next time step, RET.
(5) A fixed (un-evolved) fully connected two-layer GRU RNN, trained for each stock, to predict the stock's return at the next time step.

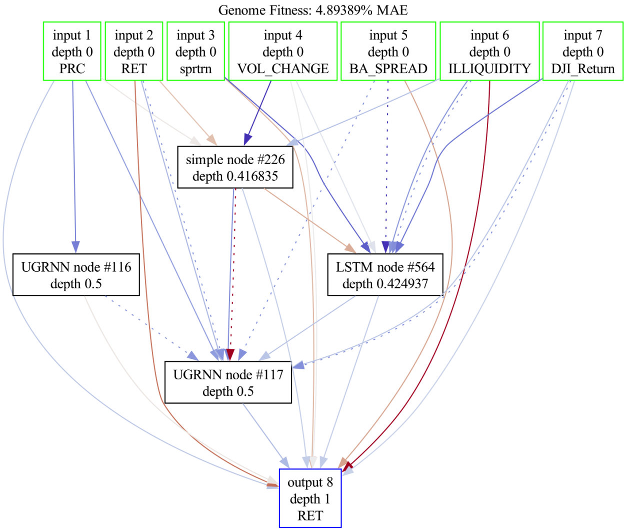

Figure 2: Best Evolved Genome for MSFT data using EXAMM. Dotted lines represent recurrent connections, darker blue lines represent edges with more positive weight values, and red lines represent edges with more negative weight values.

(6) Using EXAMM to evolve RNNs for each stock, and selecting the best performing RNNs for each stock to predict the stock's return at the next time step.

5.2.1 Forecasting Strategy Hyperparameters. The hyperparameters for EXAMM were set in the following manner. The number of islands and island size were both set as 10. The maximum number of genomes (or networks) to evaluate was 2000 and the training datasets were divided into subsequences of maximum sequence length 50 (as this has been shown to improve training and prediction performance). EXAMM used UGRNN, MGU, GRU and Δ−LSTM cells to expand the networks. Back propagation iterations was set to 5, with Adagrad used for weight update optimization having $\beta = 0.99$. Apart from this, we trained two 2-layered GRU and LSTM networks separately for 1000 epochs with learning rate 0.0001 on the split training data. The GRU and LSTM models were each trained 10 times with randomly initialized weights (for each stock) and the model with the best mean squared error (MSE), on validation data, was selected to make predictions. EXAMM was run 10 times for each stock, and the evolved RNN with the best MSE on the validation data was selected for forecasting. As an example, the best evolved RNN for the MSFT stock data can be found in Figure 2.

The VAR model required the lag order as the only hyperparameter (assuming that the data is stationary). As stock returns are percentage based on how much the stock increases or decreases this is a fairly valid assumption. VAR models were trained utilizing lag orders between 1 and 100 and the results with the best AIC (Akaike Information criteria) metric were selected, which for the data was a lag order of 20.

The ARIMA model consists of three hyperparameters: p, q, d. p is the order (number of time lags) of the autoregressive model, d is the degree of "differencing" (the number of times the data have had past values subtracted to make it stationary), and q is the order of the moving-average model. We used the `auto_arima` model from `pmdarima` library to select best possible values for each company stock data. Training ARMIA with `auto_arima` is deterministic so this resulted in a single best ARIMA model which was used.

5.2.2 Decision Making Strategy Hyperparameters. As outlined in Section 4.1, we studied three bandit algorithms. For LinUCB, we set the exploration factor α to 2.5 to enable the scheme to explore significantly in the initial stages. For the Thompson sampling approach, we set its three hyperparameters as: $R = 0.1$, $\epsilon = 0.5$, and $\delta = 0.5$. For EXP.4P five advisors under a value of $\delta = 0.5$.

Given the predictions of the TSF methods, each bandit method was trained 20 times on the validation data. The best-trained bandit on the validation data was then applied on the test data (see Figure 3 and Table 2). The results of each trained bandit on the test data was used in tests of statistical significance (see Table 3).

5.3 Performance of iCMAB framework

Each of the six TSF methods were evaluated using each of the three decision making strategies, and then compared to the baseline *buy & hold* strategy. Table 1 provides the average MSE of each forecasting method averaged across the 30 DJI stocks. Figure 3 provides the per-stock returns for each TSF method and bandit. Table 2 summarizes these results by providing the resulting average profit attained on a pool of 100$ being utilized each stock (with 100$ as the initial pool, the resulting profit is also the percentage gain), *e.g.,* if 100$ was allocated to each of the 30 stocks, utilizing EXP4.P combined with EXAMM-RNN would have resulted in a profit of 21.95$ × 30 = 658.50$.

Table 3 highlights Mann-Whitney U test results from each of the bandit models with the EXAMM evolved RNNs. At significance level $p = 0.05$, the results from EXP4.P are statistically significant improvement over both Thompson Sampling (TS) and LinUCB, however the TS and LinUCB results are not. This suggests that the EXP4.P algorithm learns a significantly better distribution than the others, which better suits this kind of task.

Table 1: Forecasting method MSE averaged across the 30 DJI companies on the validation dataset.

EXAMM	ARIMA	VAR	Trivial	GRU	LSTM
0.000410	0.000412	0.00047	0.00078	0.09949	0.08816

Table 2: Comparison of Bandit Algorithms with respect to average % gain earned across all DJI company stocks. Results for each forecasting method are present in the respective row. The Trivial-RET strategy covers a CMAB example, while the others strategies represent iCMABs.

Bandit Algorithm	Forecasting Strategy	Avg. Gain%
EXP4.P	**EXAMM-RNN**	**21.95**
	GRU	20.59
	LSTM	18.71
	VAR	14.95
	ARIMA	17.77
	Trivial-RET	15.80
Thompson Sampling with Linear Payoff	**EXAMM-RNN**	4.67
	GRU	5.23
	LSTM	4.48
	VAR	4.33
	ARIMA	4.28
	Trivial-RET	4.42
LinUCB	**EXAMM-RNN**	0.70
	GRU	0.07
	LSTM	-0.55
	VAR	1.44
	ARIMA	-0.22
	Trivial-RET	0.65
Buy & Hold	-	16.78

Table 3: Mann–Whitney U test p-values for iCMAB Bandit algorithms using EXAMM evolved RNNs.

Bandits	**EXP4.P**	**TS**	**LinUCB**
EXP4.P	-	$2.13e-05$	$2.19e-08$
TS	$2.13e-05$	-	0.13
LinUCB	$2.19e-08$	0.13	-

5.3.1 ***RQ1. Can a forecaster model improve the decision-making capacity of a CMAB?*** . EXAMM evolved RNNs showed the best performance, with ARIMA coming in very close as second and VAR also quite similar. Interestingly, the LSTM and GRU RNNs did not beat the trivial forecaster, even though those models are supposed to strongly handle time dependencies. In no cases did the trivial forecaster provide better performance than EXAMM evolved RNNs, however in some cases VAR or ARMIA performed worse than the trivial forecaster – so even though the MSE of EXAMM, ARIMA and VAR were similar, EXAMM appears to provide more robust forecasting results. Also interestingly, the GRU and LSTM networks did result in good results in combination with the bandits, except for the LinUCB bandit. It may be possible that the more advanced bandits are learning to overcome some of the TSF model inaccuracies.

Outcome *Our findings show a forecaster (world) model does indeed improve the decision making power of a CMAB, and that the EXAMM evolved RNNs provide the most robust results.*

5.3.2 ***RQ2. How does the iCMAB compare to the CMAB in maximizing rewards?*** Since the trivial method makes use of the $t - 1$ th data point for predicting t timestep, *combining the different bandit strategies with the trivial method essentially behaves as a CMAB.* In most of cases the approaches using a trained TSF method performs as well as if not better than the CMAB counterpart, and in no cases was the CMAB combination the best performing.

Outcome *Our findings show that iCMAB methods almost always outperform CMAB methods.*

5.3.3 ***RQ3. How does an evolved RNN network compare to other generative models such as VAR and other fixed architecture RNNs to account for future states and drive decision-making?*** For LinUCB and Thompson sampling, the neruoevolution based iCMAB approach performs relatively better than CMABs,

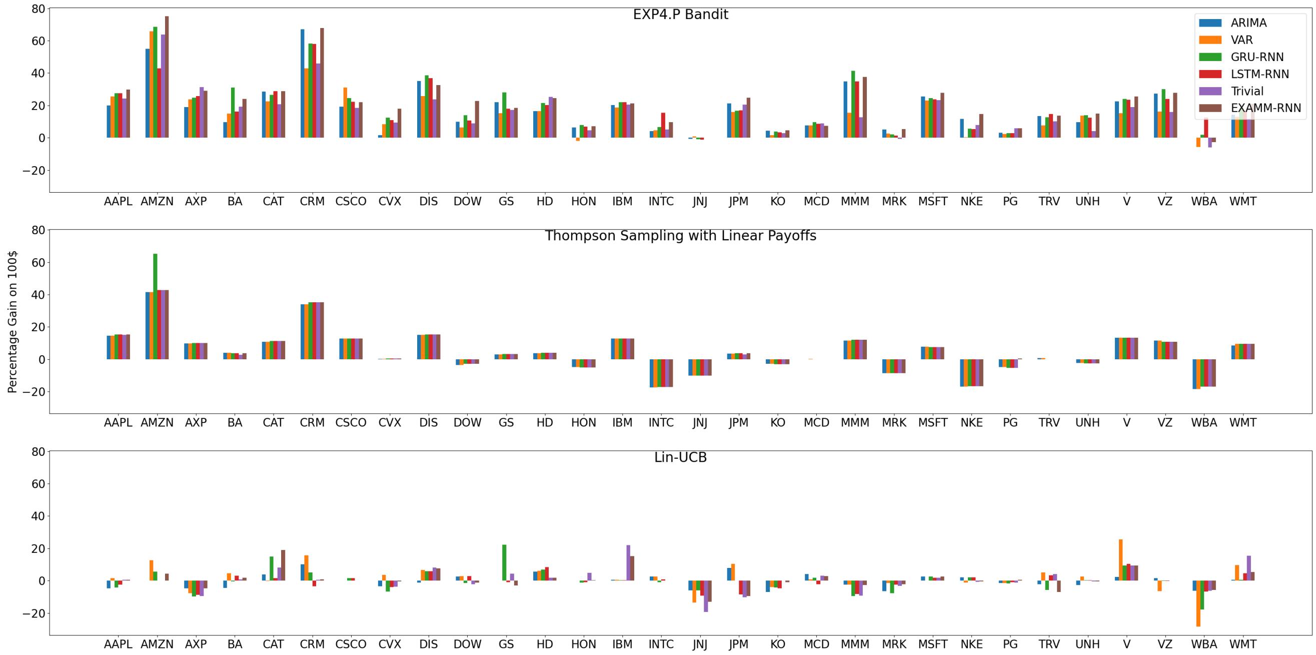

Figure 3: Bar plots of depicting percentage gain on each DJI company stock for each bandit algorithm. Each bar denotes the return on 100$ using a specific methodology for forecasting. The Y-axis denotes the return on 100$ (which is also the percentage gain) and the X-axis contains the company designations iCMAB strategies give the highest return with evolved RNN architectures in most cases.

but not better than "Buy and Hold". However, neuroevolution still provides the second best performance (behind a GRU network for Thompson sampling, and behind VAR for LinUCB) in these cases. For the EXP4.P bandit, RNNs evolved with EXAMM provide the best results, generating a policy that achieves profits near 22% on average (considering all 30 companies), improving over the *buy & hold* strategy by over 5%.

Outcome *The evolved RNNs provided the best results for the best bandit algorithm (EXP.4P) and and the second best results for the other two bandits, as such we find that neuroevolution can be a robust method for providing the TSF networks for an iCMAB.*

6 CONCLUSIONS AND FUTURE WORK

This work presents a novel iCMAB framework, and evaluates the use of neuroevolution to provide time series forecasting (TSF) models to improve its performance. This work provides a comparison between different TSF methods that can be used as the *world controller* component of an iCMAB as they operate in conjunction with varying bandit based decision making strategies. Results shown that statistical and neural network-based forecaster models generally perform relatively better than a trivial baseline method providing better context for the bandit decision making strategies, and that RNNs evolved by the EXAMM neuroevolution algorithm obtain the best result in conjunction with the best bandit strategy, and second best results on the others.

We compare these iCMAB methods to a baseline `Buy & Sell` strategy for the 30 companies in the Dow-Jones Index. This baseline is actually quite challenging to beat, as these 30 companies have been chosen to act as robust selection of stocks to optimize returns over time. Four of the five iCMAB strategies improve over this baseline utilizing the best bandit strategy, with the neuroevolution based iCMAB strategy providing the highest return at 21.95% over `Buy & Sell`'s 16.78%, with "beating the DJI index" as a significant result. This work lays the groundwork of the iCMAB framework and provides experimental verification of its superiority over the simple CMAB, and highlights the benefits of utilizing neuroevolution for TSF in stock return forecasting as one of its components.

This work, while already providing significant initial results, also opens up a number of avenues for future work. In particular, the TSF methods and bandits were trained separately for each stock. The stock prices in the DJI (or other indexes) do contain correlations, so utilizing the data for each stock as a whole to perform forecasts could lead to better forecasts. Additionally, designing newer more advanced strategies that can determine the quantities of which stock to buy, sell or hold across all stocks could significantly improve returns as such strategies could learn better methods to hedge holding and selling stocks to reduce risk and increase reward.

Acknowledgements

This material is based upon work supported by the United States National Science Foundation under grant #2225354.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 127–135.

[2] Mohammad Al Faruque, Francesco Regazzoni, and Miroslav Pajic. 2015. Design methodologies for securing cyber-physical systems. In *2015 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*. IEEE, 30–36.

[3] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*. IEEE, 106–112.

[4] Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary evolution recurrent neural networks. In *International conference on machine learning*. PMLR, 1120–1128.

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47 (2002), 235–256.

[6] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[7] Yujin Baek and Ha Young Kim. 2018. ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications* 113 (2018), 457–480.

[8] Ayan Banerjee, Krishna K Venkatasubramanian, Tridib Mukherjee, and Sandeep Kumar S Gupta. 2011. Ensuring safety, security, and sustainability of mission-critical cyber–physical systems. *Proc. IEEE* 100, 1 (2011), 283–299.

[9] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[10] Djallel Bouneffouf. 2021. Corrupted contextual bandits: Online learning with corrupted context. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3145–3149.

[11] Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. 2017. Context attentive bandits: Contextual bandit with restricted context. *arXiv preprint arXiv:1705.03821* (2017).

[12] Djallel Bouneffouf, Sohini Upadhyay, and Yasaman Khazaeni. 2020. Contextual bandit with missing rewards. *arXiv preprint arXiv:2007.06368* (2020).

[13] Yahya Eru Cakra and Bayu Distiawan Trisedya. 2015. Stock price prediction using linear regression based on sentiment analysis. In *2015 international conference on advanced computer science and information systems (ICACSIS)*. IEEE, 147–154.

[14] Luo Chao, Jiang Zhipeng, and Zheng Yuanjie. 2019. A novel reconstructed training-set SVM with roulette cooperative coevolution for financial time series classification. *Expert Systems with Applications* 123 (2019), 283–298.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[16] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 208–214.

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[18] Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. 2016. Capacity and Trainability in Recurrent Neural Networks. *arXiv preprint arXiv:1611.09913* (2016).

[19] Travis Desell, AbdElRahman ElSaid, and Alexander G. Ororbia. 2020. An Empirical Exploration of Deep Recurrent Connections Using Neuro-Evolution. In *The 23nd International Conference on the Applications of Evolutionary Computation (EvoStar: EvoApps 2020)*. Seville, Spain.

[20] Maria Dimakopoulou, Zhengyuan Zhou, Susan Athey, and Guido Imbens. 2019. Balanced linear contextual bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3445–3453.

[21] Kevin Dowd. 2007. *Measuring market risk*. John Wiley & Sons.

[22] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[23] AbdElRahman ElSaid, Joshua Karnas, Zimeng Lyu, Daniel Krutz, Alexander G Ororbia, and Travis Desell. 2020. Neuro-Evolutionary Transfer Learning through Structural Adaptation. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 610–625.

[24] AbdElRahman ElSaid, Joshua Karns, Zimeng Lyu, Daniel Krutz, Alexander Ororbia, and Travis Desell. 2020. Improving neuroevolutionary transfer learning of deep recurrent neural networks through network-aware adaptation. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 315–323.

[25] Robert F Engle and Clive WJ Granger. 2003. Time-series econometrics: cointegration and autoregressive conditional heteroskedasticity. *Advanced information on the Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel* 95 (2003), 98.

[26] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research* 270, 2 (2018), 654–669.

[27] Pratik Gajane, Tanguy Urvoy, and Emilie Kaufmann. 2018. Corrupt bandits for preserving local privacy. In *Algorithmic Learning Theory*. PMLR, 387–412.

[28] Nicolas Galichet, Michele Sebag, and Olivier Teytaud. 2013. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Asian Conference on Machine Learning*. PMLR, 245–260.

[29] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online clustering of bandits. In *International conference on machine learning*. PMLR, 757–765.

[30] Umang Gupta, Vandana Bhattacharjee, and Partha Sarathi Bishnu. 2022. StockNet—GRU based stock index prediction. *Expert Systems with Applications* 207 (2022), 117986.

[31] Faizal Hafiz, Jan Broekaert, Davide La Torre, and Akshya Swain. 2023. Co-evolution of neural architectures and features for stock market forecasting: A multi-objective decision perspective. *Decision Support Systems* 174 (2023), 114015.

[32] Kyle Helfrich, Devin Willmott, and Qiang Ye. 2018. Orthogonal recurrent neural networks with scaled Cayley transform. In *International Conference on Machine Learning*. PMLR, 1969–1978.

[33] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, and Herbert Kimura. 2019. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications* 124 (2019), 226–251.

[34] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[35] John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79, 8 (1982), 2554–2558.

[36] Xiaoguang Huo and Feng Fu. 2017. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science* 4, 11 (2017), 171377.

[37] Anmol Kagrecha, Jayakrishnan Nair, and Krishna Jagannathan. 2022. Statistically robust, risk-averse best arm identification in multi-armed bandits. *IEEE Transactions on Information Theory* (2022).

[38] Anmol Kagrecha, Jayakrishnan Nair, and Krishna P Jagannathan. 2019. Distribution oblivious, risk-aware algorithms for multi-armed bandits with unbounded rewards.. In *NeurIPS*. 11269–11278.

[39] Gul Mummad Khan and Durr e Nayab. 2018. Learning Trends on the Fly in Time Series Data Using Plastic CGP Evolved Recurrent Neural Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer International Publishing, Cham, 199–207.

[40] Ha Young Kim and Chang Hyun Won. 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications* 103 (2018), 25–37.

[41] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[42] Qi Li, Norshaliza Kamaruddin, Siti Sophiayati Yuhaniz, and Hamdan Amer Ali Al-Jaifi. 2024. Forecasting stock prices changes using long-short term memory neural network with symbolic genetic programming. *Scientific reports* 14, 1 (2024), 422.

[43] Yifan Li, Jing Liu, and Yingzhi Teng. 2022. A decomposition-based memetic neural architecture search algorithm for univariate time series forecasting. *Applied Soft Computing* 130 (2022), 109714.

[44] Yifan Lin, Yuhao Wang, and Enlu Zhou. 2023. Risk-averse contextual multi-armed bandit problem with linear payoffs. *Journal of Systems Science and Systems Engineering* 32, 3 (2023), 267–288.

[45] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. 2020. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 2128–2135.

[46] Wen Long, Zhichen Lu, and Lingxiao Cui. 2019. Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems* 164 (2019), 163–173.

[47] James H. Lorie. 1960. Center for Research in Security Prices, LLC. https://www.crsp.org/

[48] Tyler Lu, Dávid Pál, and Martin Pál. 2010. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 485–492.

[49] Zimeng Lyu, AbdElRahman ElSaid, Joshua Karns, Mohamed Mkaouer, and Travis Desell. 2021. An Experimental Study of Weight Initialization and Lamarckian Inheritance on Neuroevolution. *The 24th International Conference on the Applications of Evolutionary Computation (EvoStar: EvoApps)* (2021).

[50] Zimeng Lyu, Joshua Karnas, AbdElRahman ElSaid, Mohamed Mkaouer, and Travis Desell. 2021. Improving Distributed Neuroevolution Using Island Extinction and Repopulation. *The 24th International Conference on the Applications of Evolutionary Computation (EvoStar: EvoApps)* (2021).

[51] Odalric-Ambrym Maillard. 2013. Robust risk-averse stochastic multi-armed bandits. In *International Conference on Algorithmic Learning Theory*. Springer, 218–233.

[52] Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5 (1943), 115–133.

[53] João Nadkarni and Rui Ferreira Neves. 2018. Combining NeuroEvolution and Principal Component Analysis to trade in the financial markets. *Expert Systems with Applications* 103 (2018), 184–195.

[54] Alexander Ororbia, AbdElRahman ElSaid, and Travis Desell. 2019. Investigating Recurrent Neural Network Memory Structures Using Neuro-evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. ACM, New York, NY, USA, 446–455. https://doi.org/10.1145/3321707.3321795

[55] Alexander G. Ororbia II, Tomas Mikolov, and David Reitter. 2017. Learning Simpler Language Models with the Differential State Framework. *Neural Computation* 0, 0 (2017), 1–26. https://doi.org/10.1162/neco_a_01017 arXiv:https://doi.org/10.1162/neco_a_01017 PMID: 28957029.

[56] Felipe Dias Paiva, Rodrigo Tomás Nogueira Cardoso, Gustavo Peixoto Hanaoka, and Wendel Moreira Duarte. 2019. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications* 115 (2019), 635–655.

[57] Jeffrey Palmerino, Qi Yu, Travis Desell, and Daniel Krutz. 2019. Improving the decision-making process of self-adaptive systems by accounting for tactic volatility. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 949–961.

[58] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*. Pmlr, 1310–1318.

[59] Mingyue Qiu, Yu Song, and Fumio Akagi. 2016. Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos, Solitons & Fractals* 85 (2016), 1–7.

[60] Mehreen Rehman, Gul Muhammad Khan, and Sahibzada Ali Mahmud. 2014. Foreign currency exchange rates prediction using cgp and recurrent neural network. *IERI Procedia* 10 (2014), 239–244.

[61] David E Rumelhart and David Zipser. 1985. Feature discovery by competitive learning. *Cognitive science* 9, 1 (1985), 75–112.

[62] Aadirupa Saha, Pierre Gaillard, and Michal Valko. 2020. Improved sleeping bandits with stochastic action sets and adversarial rewards. In *International Conference on Machine Learning*. PMLR, 8357–8366.

[63] Patrick Saux and Odalric Maillard. 2023. Risk-aware linear bandits with convex loss. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 7723–7754.

[64] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 497–504.

[65] Shubham Sharma, Yunfeng Zhang, Jesús M Ríos Aliaga, Djallel Bouneffouf, Vinod Muthusamy, and Kush R Varshney. 2020. Data augmentation for discrimination prevention and bias disambiguation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 358–364.

[66] Aleksandrs Slivkins. 2019. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272* (2019).

[67] Mehak Usmani, Syed Hasan Adil, Kamran Raza, and Syed Saad Azhar Ali. 2016. Stock market prediction using machine learning techniques. In *2016 3rd international conference on computer and information sciences (ICCOINS)*. IEEE, 322–327.

[68] Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*. Springer, 437–448.

[69] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization bandits for interactive recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[70] Xiao Xu, Fang Dong, Yanghua Li, Shaojian He, and Xin Li. 2020. Contextual-bandit based personalized recommendation with time-varying user interests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6518–6525.

[71] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance*. 1–8.

[72] Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1112–1119.

[73] Seyoung Yun, Jun Hyun Nam, Sangwoo Mo, and Jinwoo Shin. 2017. *Contextual multi-armed bandits under feature uncertainty*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[74] Faheem Zafari, Gul Muhammad Khan, Mehreen Rehman, and Sahibzada Ali Mahmud. 2014. Evolving recurrent neural network using cartesian genetic programming to predict the trend in foreign currency exchange rates. *Applied Artificial Intelligence* 28, 6 (2014), 597–628.

[75] Guo-Bing Zhou, Jianxin Wu, Chen-Lin Zhang, and Zhi-Hua Zhou. 2016. Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing* 13, 3 (2016), 226–234.

[76] Li Zhou. 2015. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326* (2015).