

PREDICTION INFUSED DEEP REINFORCEMENT LEARNING FOR AUTONOMOUS TRADING SYSTEMS

TECHNICAL FIELD

[0001] Various aspects of this disclosure generally relate to machine learning, and more particularly, to prediction infused deep reinforcement learning for financial portfolio management.

BACKGROUND

[0002] Machine learning is a subfield of computer science that explores the study and development of algorithms that can learn from data so as to either make predictions or produce other actionable insights. Such algorithms are aimed at building a model through learning from historic data inputs and make predictions or decisions on new data based on the model. Machine learning is employed in a range of computing tasks where deriving rules and insights from the past experiences, and/or continuous human inspection is infeasible. Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction. These analytical models allow researchers, data scientists, engineers, and analysts to produce reliable, repeatable decisions and results and uncover hidden insights through generalizing from historical relationships and trends in the data.

[0003] In finance, time series machine learning models are widely used to try to predict or detect future asset behaviours. For example, auto-encoders and long short-term memory (LSTM) units are used to model financial data. In some instances, neural network based methods are used to make asset predictions based on news articles (i.e., event based trading). In some instances, Bayesian procedure based on Gaussian process is introduced for nonstationary time series prediction.

[0004] Financial portfolio management is the process of constant redistribution of a fund into different financial products. It is desirable to create an automatic trading system such that these investment decisions and actions are made periodically and correctly.

[0005] Reinforcement (RL) learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize expected

total return. The problem, due to its generality, is studied in many disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In the operations research and control literature, reinforcement learning may be called approximate dynamic programming, or neuro-dynamic programming.

[0006] In the financial domain, people are often mindful of risks. Risk-aversion comes from a situation where a probability can be assigned to each possible outcome of a situation. In the sequential decision-making setting, the return is a random variable due to the stochastic transitions/rewards for a given Markov Decision Process (MDP). Such uncertainty becomes important when there is significant stochasticity in the MDP transitions, which may lead to significant variability in the return. The natural method for dealing with this uncertainty, motivated by classical studies in the financial literature, is through the notion of risk, such as its variance, value-at-risk (VaR), conditional value-at-risk (CVaR), or exponential-utility. Such measures capture the variability of the return, or quantify the effect of rare but potentially disastrous outcomes. Based on the multi-period or temporal nature, the risk in a MDP can be classified into two types: static risk measure and dynamic risk measure. In particular, risk-sensitive MDP considers stochastic optimization problems in which the objective involves a risk measure of the random cost, in contrast to the typical expected cost objective. Such problems are important when the decision-maker wishes to manage the variability of the cost, in addition to its expected outcome, and are standard in various applications in finance and operations research. Risk-sensitive MDP with dynamic risk measure and ambiguity-averse MDPs are closely connected. Similar to classical MDP, risk-sensitive MDP suffers from the “curse of dimensionality” and requires the full knowledge of the model.

[0007] Risk-sensitive reinforcement learning is proposed to alleviate the curse, and is able to produce good solutions in a model-free environment. For instance, dynamic quantile-based risk measures are considered, and a simulation-based approximate dynamic programming algorithm was proposed. In another instance, a simulation-based approximate fitted value iteration for dynamic risk measures was proposed. In yet another instance, dynamic coherent risk measures are studied, and actor-critic algorithms are provided with value function approximation. In yet another instance, linear function approximation for variance-related criteria in MDPs was developed. In such an instance, temporal difference and least-squares temporal difference algorithms were developed for

this problem class. A parametric policy gradient technique for static-CVaR minimizing reinforcement learning was proposed in some instances. In another instance, a cutting plane algorithm for time-consistent multistage linear stochastic programming problems was given. Moreover, different models studied exponential-utility functions, mean-variance models, policy gradient for coherent static risk measures, and dynamic coherent risk for systems with linear dynamics. All these models use the historical data/experience as inputs.

[0008] Recently, reinforcement learning methods are increasingly being investigated to determine their effectiveness in the field of finance. For example, reinforcement learning could be applied to finance to train an agent to manage financial portfolios autonomously. However, traditional models are risk-neutral, thus leading to high-risk asset portfolio.

SUMMARY

[0009] The following presents a simplified summary in order to provide a basic understanding of various aspects of the disclosed invention. This summary is not an extensive overview of all contemplated aspects, and is intended to neither identify key or critical elements of all aspects nor delineate the scope of any or all aspects. The sole purpose is to present some concepts of one or more aspects in a simplified form as a prelude to the more detailed description that is presented later.

[0010] This disclosure aims to solve the portfolio management problem, which is the decision making process of continuously reallocating an amount of fund into a number of different financial investment products (also referred to as assets), aiming to maximize the return while restraining the risk. Some embodiments of the disclosure may include the creation of an autonomous algorithmic trading artificial intelligence system (hereinafter referred to as the agent) that constantly allocates fund over the assets portfolio at each time interval by not just conditionalizing its actions on past data provided by the market but also internally generated hypotheses of future performances of each asset in the portfolio.

[0011] In some embodiments, such Agent may be designed by formulating the portfolio management problem as a Markov decision process and proposing deep reinforcement learning (RL) algorithms to solve it. These reinforcement learning algorithms are able to incorporate the prediction of markets' movement and yield decisions (trading signals) to maximize the total investment return as well as to minimize the risk of

return. Specially, some embodiments may achieve high annualized return, Sharpe ratio and Sortino ratio while obtaining low strategy risks such as value-at-risk (VaR), conditional value-at-risk (CVaR) and maximum drawdown (MDD). Some embodiments include an autonomous algorithmic trading system that is built upon machine learning algorithms with prediction features. This robust framework consistently works well regardless of specific asset class type.

[0012] In one aspect of the disclosure, a method, a computer-readable medium, and an apparatus for portfolio management are provided. The apparatus may retrieve the most recent transaction data of one or more financial markets. The apparatus may forecast, using at least one prediction model, one or more prediction features based on the most recent transaction data. The apparatus may determine using a reinforcement learning based intelligent agent, based on the one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets. The apparatus may initiate a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution.

[0013] To the accomplishment of the foregoing and related ends, the aspects disclosed include the features hereinafter fully described and particularly pointed out in the claims. The following description and the annexed drawings set forth in detail illustrate certain features of the aspects of the disclosure. These features are indicative, however, of but a few of the various ways in which the principles of various aspects may be employed, and this description is intended to include all such aspects and their equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a diagram illustrating an example of portfolio management.

[0015] FIG. 2 is a diagram illustrating an example of the architecture of a completely autonomous end-to-end trading system.

[0016] FIG. 3 is a block diagram illustrating an example of the architecture of the agent described above in FIG. 2.

[0017] FIG. 4 is a flowchart of a method of training a trading system (e.g., the trading system described above in FIG. 2).

[0018] FIG. 5 is a flowchart of a method of running a trading system (e.g., the trading system described above in FIG. 2) during actual trade execution.

[0019] FIG. 6 is a flowchart of a method of training or testing a trading system (e.g., the trading system described above in FIG. 2).

[0020] FIG. 7 is a conceptual data flow diagram illustrating the data flow between different means/components in an exemplary apparatus.

[0021] FIG. 8 is a diagram illustrating an example of a hardware implementation for an apparatus employing a processing system.

DETAILED DESCRIPTION

[0022] The detailed description set forth below in connection with the appended drawings is intended as a description of various possible configurations and is not intended to represent the only configurations in which the concepts described herein may be practiced. The detailed description includes specific details for the purpose of providing a thorough understanding of various concepts. However, it will be apparent to those skilled in the art that these concepts may be practiced without these specific details. In some instances, well known structures and components are shown in block diagram form in order to avoid obscuring such concepts.

[0023] Several aspects of a prediction infused deep reinforcement learning technique will now be presented with reference to various apparatus and methods. The apparatus and methods will be described in the following detailed description and illustrated in the accompanying drawings by various blocks, components, circuits, processes, algorithms, etc. (collectively referred to as “elements”). These elements may be implemented using electronic hardware, computer software, or any combination thereof. Whether such elements are implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system.

[0024] By way of example, an element, or any portion of an element, or any combination of elements may be implemented as a “processing system” that includes one or more processors. Examples of processors include microprocessors, microcontrollers, graphics processing units (GPUs), central processing units (CPUs), application processors, digital signal processors (DSPs), reduced instruction set computing (RISC) processors, systems on a chip (SoC), baseband processors, field programmable gate arrays (FPGAs),

programmable logic devices (PLDs), state machines, gated logic, discrete hardware circuits, and other suitable hardware configured to perform the various functionality described throughout this disclosure. One or more processors in the processing system may execute software. Software shall be construed broadly to mean instructions, instruction sets, code, code segments, program code, programs, subprograms, software components, applications, software applications, software packages, routines, subroutines, objects, executables, threads of execution, procedures, functions, etc., whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise.

[0025] Accordingly, in one or more example embodiments, the functions described may be implemented in hardware, software, or any combination thereof. If implemented in software, the functions may be stored on or encoded as one or more instructions or code on a computer-readable medium. Computer-readable media includes computer storage media. Storage media may be any available media that can be accessed by a computer. By way of example, and not limitation, such computer-readable media may include a random-access memory (RAM), a read-only memory (ROM), an electrically erasable programmable ROM (EEPROM), optical disk storage, magnetic disk storage, other magnetic storage devices, combinations of the aforementioned types of computer-readable media, or any other medium that can be used to store computer executable code in the form of instructions or data structures that can be accessed by a computer.

[0026] In one aspect of the disclosure, a novel method of improving the actions of RL agents in the field of financial portfolio management via the use of prediction models is provided. In some embodiments, such prediction based RL algorithms may be incorporated together to build a system for stable, profitable autonomous trading. In particular, such embodiments are both theoretically sound and practically workable to produce financial trading strategy with high return and low risks.

[0027] Some embodiments may include two inter-related components. The first is an algorithm of making predictions and incorporating the predictions into an autonomous agent's decision making process. The second is the incorporation of the algorithm into a completely autonomous end-to-end trading system.

[0028] In period t , price relative vector y_t can be defined as,

$$y_t \triangleq v_t \oslash v_{t-1} = \left(1, \frac{v_{1,t}}{v_{1,t-1}}, \dots, \frac{v_{m,t}}{v_{m,t-1}} \right)^T.$$

[0029] Portfolio value is $p_t = p_{t-1}y_t \cdot w_{t-1}$.

[0030] Rate of return is,

$$\rho_t \triangleq \frac{p_t}{p_{t-1}} - 1 = y_t \cdot w_{t-1} - 1.$$

[0031] Logarithmic rate of return is,

$$r_t \triangleq \ln\left(\ln \frac{p_t}{p_{t-1}}\right) = \ln(\ln(y_t \cdot w_{t-1}))$$

where,

m: number of assets;

t: trading period;

$v_t \in \mathbb{R}_+^{m+1}$: closing price vector of assets with first element $v_{0,t} \triangleq 1, \forall t$;

$v_t^{(hi)}, v_t^{(lo)} \in \mathbb{R}_+^{m+1}$: highest and lowest price vectors of period t ;

$w_{t-1} \in \mathbb{R}_+^{m+1}$: portfolio vector at the beginning of period t , $\|w_t\|_1 = 1, \forall t$, and $w_0 \triangleq (1, 0, \dots, 0)^T$.

[0032] The objective is to maximize

$$p_f = p_0 \exp\left(\sum_{t=1}^{t_f+1} r_t\right) = p_0 \prod_{t=1}^{t_f+1} y_t \cdot w_{t-1}$$

where $t_f + 1$ is the total number of trading periods.

[0033] FIG. 1 is a diagram 100 illustrating an example of portfolio management. In the example, the commission rate for purchasing and selling is assumed to be $0 < c < 1$. Over the time period t , weights change is,

$$w'_t = \frac{y_t \odot w_{t-1}}{y_t \cdot w_{t-1}}$$

where the operator in the denominator is inner product, while the operator in the numerator denotes the element-wise product.

[0034] Portfolio value change $p_t = \mu_t p'_t$, where $\mu_t \in (0, 1]$ is the transaction remainder factor $\mu_t = \mu_t(w_{t-1}, w_t, y_t)$,

$$\mu_t = c \sum_{i=1}^m |w'_{t,i} - w_{t,i}|.$$

[0035] Considering the transaction costs, the objective becomes

$$p_f = p_0 \exp\left(\sum_{t=1}^{t_f+1} r_t\right) = p_0 \prod_{t=1}^{t_f+1} \mu_t y_t \cdot w_{t-1}.$$

[0036] In some embodiments, during period t , price relative vector y_t may be provided to prediction models 102 and agent 104. With the help of the output of the prediction

models 102, the agent 104 may be able to generate buy or sell signals in order to achieve portfolio weight vector w_t . Similarly, during period $t+1$, price relative vector y_{t+1} may be provided to the prediction models 102 and the agent 104. With the help of the output of the prediction models 102, the agent 104 may be able to generate buy or sell signals in order to achieve portfolio weight vector w_{t+1} .

[0037] FIG. 2 is a diagram illustrating an example of the architecture of a completely autonomous end-to-end trading system 200. In the example, the trading system 200 may include an agent 216, prediction models 204, a data fetcher module 206, a trade execution engine 210, and a broker 208. In some embodiments, the agent 216 may be the agent 104 described above in FIG. 1; and the prediction models 204 may be the prediction models 102 described above in FIG. 1.

[0038] During training, at each training iteration (i.e. at each training time step), the following operations may be performed. The broker 208 may be configured to observe only historical data of the market 214, which may include one or more markets on which financial investment products are traded. The data fetcher module 206 may retrieve historical data from the broker 208 and down sample the retrieved data to the desired frequency for the agent 216 to output portfolio weight distribution (i.e., action). The prediction models 204 may include a neural network or any other prediction algorithm. The prediction models 204 may be trained using the historical data provided by the broker 208 (i.e., data provided by the broker 208 up to the training time instant). The trained prediction models 204 may be used to forecast price or sentiment related features in the next time step (or next several time steps). The agent 216 may be trained using both past price data provided by the data fetcher module 206 and the output of prediction models 204, so that the agent 216 may output desired portfolio weight distribution over assets under consideration. The desired portfolio weight distribution may be passed to the trade execution engine 210, which will interact with the broker 208 to try to achieve the desired portfolio weight distribution. Subsequently, the data fetcher module 206 may retrieve the actual portfolio weight distribution after the trades done by the broker 208, together with other relevant trade data. The data fetcher module 206 may pass the retrieved information back to the agent 216 and the prediction models 204. The operations described above may repeat for several training iterations (i.e., training time steps).

[0039] At test time (i.e., at actual trade execution), in some embodiments, the prediction models 204 and the agent 216 may be frozen (i.e., the prediction models 204

and the agent 216 are no longer trained/updated). In other embodiments, the prediction models 204 and the agent 216 may be constantly updated/trained by new data in an online fashion at test time.

[0040] FIG. 3 is a block diagram 300 illustrating an example of the architecture of the agent 216 described above in FIG. 2. In some embodiments, the agent 216 may include several neural networks. In the example, data from the data fetcher module 206 are passed to the agent 216, which may have multiple feature extraction (FE) networks (e.g., FE networks 306 and 314), feature analysis (FA) networks (e.g., FA networks 308 and 316), and network improvement (NI) networks (e.g., NI networks 310 and 318). The agent 216 may further include a memory 304.

[0041] In the example, only two of each type of networks are shown. But one of ordinary skill in the art would recognize that any number of each type of networks could be used if required in order to improve exploration via parameter noise perturbation, improve stability through the use of target networks, etc. In some embodiments, by making use of outputs from the prediction models 204, fed directly to either the FE or FA networks, a significant improvement in the performance of the agent 216 may be achieved.

[0042] In some embodiments, the prediction models 204 may use temporal convolutional neural networks (TCNN), e.g., WaveNet based TCNN models. In some embodiments, the prediction models 204 may use recurrent neural network based models or other types of memory enabled deep neural networks. In some embodiments, the prediction models 204 may use Bayesian based models, e.g., deep Gaussian processes. In some embodiments, the prediction models 204 may use any other prediction model for either price or sentiment related features. In some embodiments, the prediction models 204 may use a combination of two or more models described above.

[0043] FIG. 4 is a flowchart 400 of a method of training a trading system (e.g., the trading system 200 described above in FIG. 2). In one embodiment, the method may be performed by an apparatus (e.g., the apparatus 702/702' shown in FIG. 7/FIG. 8). In some embodiments, the operations performed by the method may correspond to some of the operations described above with reference to FIG. 2.

[0044] At 402, the apparatus may initialize neural networks of the trading system. In some embodiments, some of the neural networks being initialized may be part of the agent 216. In such embodiments, the agent 216 may be configured with an actor-critic

configuration. In some embodiments, some of the neural networks being initialized may be part of the prediction models 204.

[0045] At 404, the apparatus may fetch current market data (i.e., the most up-to-date market data for the current time step) via the data fetcher module 206. In some embodiments, the market data may include prices (e.g., relative prices) of financial investment products and technical indicators. The apparatus may then proceed to both 406 and 420.

[0046] At 420, the apparatus may store the fetched market data in the prediction model's training memory.

[0047] At 422, the apparatus may update the prediction model (e.g., the TCNN prediction model) based on the stored data.

[0048] At 424, the apparatus may make a prediction of one or more steps into the horizon (i.e., of one or more future time steps) with the prediction model. The apparatus may then send the prediction to operations performed at 412 and 414.

[0049] At 406, the apparatus may convert the fetched market data to neural network's states.

[0050] At 408, the apparatus may store necessary data in the agent replay buffer. In some embodiments, unlike typical actor-critic setups, the following may be stored in the replay buffer: state; action; reward; next state; and external features.

[0051] At 410, the apparatus may sample data in replay buffer via prioritized replay (e.g., based on temporal difference errors).

[0052] At 412, the apparatus may update the critic network conditioned on the sampled data and the prediction provided by the prediction model.

[0053] At 414, the apparatus may update the actor network based on the updated critic network and the prediction provided by the prediction model.

[0054] At 416, the apparatus may perform parameter noise perturbation of the updated actor network.

[0055] At 418, the apparatus may determine an action (e.g., the desired portfolio weights) for the next time step based on the updated agent. The apparatus may then proceed to 430.

[0056] At 430, the apparatus may feed the action to a trade execution engine (e.g., the trade execution engine 210) and subsequently the broker (e.g., the broker 208).

[0057] At 432, the apparatus may get actually realized portfolio weights after the desired actions are submitted for execution. The actually realized portfolio weights may be different from the desired portfolio weights, as some of the actions may be executed successfully while the others may fail to be executed. The apparatus may then loop back to 404.

[0058] FIG. 5 is a flowchart 500 of a method of running a trading system (e.g., the trading system 200 described above in FIG. 2) during actual trade execution. In one embodiment, the method may be performed by an apparatus (e.g., the apparatus 702/702' shown in FIG. 7/FIG. 8). In some embodiments, the operations performed by the method may correspond to some of the operations described above with reference to FIG. 2. In some embodiments, the trading system may be trained using the method described above with reference to FIG. 4.

[0059] At 502, the apparatus may initialize neural networks with trained parameters (e.g., weights). In some embodiments, some of the neural networks being initialized may be part of the agent 216. In such embodiments, the agent 216 may be configured with an actor-critic configuration. In some embodiments, some of the neural networks being initialized may be part of the prediction models 204.

[0060] At 506, the apparatus may fetch current market data (i.e., the most up-to-date market data for the current time step), as well as the sentiment information, via the data fetcher module 206. In some embodiments, the market data may include prices (e.g., relative prices) of financial investment products, technical indicators, and sentiment information. The apparatus may then proceed to both 508 and 520.

[0061] At 520, the apparatus may make a prediction of one or more steps into the horizon (i.e., of one or more future time steps) with the prediction model (e.g., the TCNN prediction model) based on the fetched market data. The apparatus may then send the prediction to operations performed at 510.

[0062] At 508, the apparatus may convert the fetched market data to neural network's internal states.

[0063] At 510, the apparatus may determine an action (e.g., the desired portfolio weights) for the next time step via the trained actor network. The action may be determined based on the fetched market data and the prediction provided by the prediction model. The apparatus may then proceed to 530.

[0064] At 530, the apparatus may feed the action to a trade execution engine (e.g., the trade execution engine 210) and subsequently the broker (e.g., the broker 208).

[0065] At 532, the apparatus may get actually realized portfolio weights after the desired actions are submitted for execution. The actually realized portfolio weights may be different from the desired portfolio weights, as some of the actions may be executed successfully while the others may fail to be executed. The apparatus may then loop back to 506.

[0066] In some embodiments, by incorporating predictions into the agent's neural network, the agent is able to make use of this additional information, as though it was an additional feature, together with historical data to make a better decision in terms of portfolio weights allocation. As can be seen in the table below, the addition of predictions into the agent's decision process significantly improve reward to risk ratios.

	Memory based FE with Prediction Inputs	Memory based FE without Prediction Inputs	Memoryless FE with Prediction Inputs	Memoryless FE without Prediction Inputs
Percentage Portfolio Gains over trading period/ % (trading period??)	19.3	15.6	16.3	14.5
Annualized Return/ %	13.14	10.76	11.11	9.99
Annualized Volatility/ %	11.98	10.94	10.94	10.67
VAR / %	1.21	1.02	0.99	1.00
cVAR/ %	1.89	1.73	1.72	1.69
MaxDD/ %	7.9	7.6	7.8	7.7
Sharpe Ratio	1.10	0.98	1.02	0.94
Sortino Ratio	6.16	5.41	5.60	5.13

[0067] In some embodiments, one of the neural network models incorporated in the prediction model may have a WaveNet based temporal convolutional neural network (TCNN) architecture. In some embodiments, the FE layers in FIG. 3 may either be a memory-based neural network or a memoryless neural network. In some embodiments, the FA layer in FIG. 3 could either be a fully connected dense network or a convolutional neural network. Some examples of preferred embodiments are shown in the table below.

Preferred Embodiment	FE Layer	FA Layer	Prediction Model
1	Memory-based Neural Networks	Dense Neural Network	TCNN Wavenet Model
2	Memory-based Neural Networks	Convolutional Neural Network	TCNN Wavenet Model
3	Memoryless Neural Networks	Dense Neural Network	TCNN Wavenet Model
4	Memoryless Neural Networks	Convolutional Neural Network	TCNN Wavenet Model

[0068] Deep Deterministic Policy Gradients (DDPG) is an off-policy actor-critic-style reinforcement learning algorithm. Some embodiments of the disclosure may be integrated with DDPG. Below is an example of the pseudocode for a prediction infused DDPG agent:

```

1:   Initialize critic, actor, and perturbed actor networks;
2:   Initialize target critic, target actor, and perturbed target actor networks;
3:   Initialize replay buffer;
4:   for episode = 1,..., M do
5:       Receive initial observation state
6:       for t = 1,..., T do
7:           Predict next state using Prediction Models with observed state;
8:           Make use of observed state and predictions to produce action (i.e.,
              select portfolio weights) based on the perturbed actor network;
9:           Execute action, obtain reward, and observe new state;
10:          Store the transition, i.e., state, action, reward, next state, and
              prediction, in the replay buffer;
11:          Sample a minibatch of  $N$  transitions from replay buffer via
              prioritized replay, according to temporal difference error;
12:          Update the critic network by annealing the prioritized replay bias
              while minimizing its loss using the sampled states, actions, rewards,
              and predictions;
13:          Update the actor network using the sampled policy gradient;
14:          Update the target critic and target actor networks;
15:          Update the perturbed actor and perturbed target actor networks;
16:       end for;
17:   end for.

```

[0069] Even though an off-policy actor-critic-style reinforcement learning algorithm is described in the example above, one of ordinary skill in the art would recognize that some embodiments of the disclosure may be applicable to on-policy reinforcement learning algorithms as well. Even though an actor-critic Deep Deterministic Policy Gradient (DDPG) architecture for the agent is described as an example in the disclosure, other algorithms such as trust region based algorithms like Proximal Policy Optimisation (PPO) may be used instead without changing the essence of the disclosure. Further, even though TCNN WaveNet model is used as an example of the prediction model in the disclosure, an ensemble of other models could also be used as the prediction model without changing the essence of the disclosure.

[0070] In some embodiments, multiple agents may be used in the trading system, each providing a recommended portfolio allocation. The multiple outputs of the agents may then be averaged in some way to give the final desired portfolio allocation. In some embodiments, multiple prediction models may be used in the trading system to provide multiple predictions, which act as additional information for the agent.

[0071] Some embodiments of the end-to-end system is primarily for autonomous algorithmic trading. Nevertheless, the architecture described in the disclosure may be used in general reinforcement learning algorithms to improve the performance and robustness of the reinforcement learning algorithms. Thus, embodiments of the disclosure are applicable not only to the financial domain, but also to general reinforcement learning domains.

[0072] Embodiments of the disclosure do not detrimentally affect convergence rate of the Agent. In fact, due to the larger amount of data being used by the agent, the policy gradients are larger, indicating that the network is actually adapting itself to make use of the predictions.

[0073] FIG. 6 is a flowchart 600 of a method of training or testing a trading system (e.g., the trading system 200 described above in FIG. 2). In one embodiment, the method may be performed by an apparatus (e.g., the apparatus 702/702' shown in FIG. 7/FIG. 8). In some embodiments, the operations performed by the method may correspond to some of the operations described above with reference to FIGS. 2, 4, and 5.

[0074] At 602, the apparatus may retrieve the most recent transaction data of one or more financial markets on which financial investment products are traded. In some embodiments, the one or more financial markets may include at least one of one or more

crypto currency markets, one or more foreign exchange markets, or one or more equity markets.

[0075] At 604, the apparatus may forecast, using at least one prediction model, one or more prediction features based on the most recent transaction data. In some embodiments, the at least one prediction model may include a temporal convolutional neural network. In some embodiments, the at least one prediction model may be trained using historical transaction data of the one or more financial markets. In some embodiments, the one or more prediction features may include at least one of one or more price related features or one or more predicted sentiments.

[0076] At 606, the apparatus may determine using a reinforcement learning based intelligent agent, based on the forecasted one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets. In some embodiments, the intelligent agent may include a set of feature extraction networks, a set of feature analysis networks, and a set of network improvement networks. In some embodiments, outputs from the at least one prediction model may be fed directly to at least one of the set of feature extraction networks or the set of feature analysis networks. In some embodiments, the intelligent agent may be trained using historical transaction data of the one or more financial markets and historical outputs from the at least one prediction model. In some embodiments, the intelligent agent may include a replay buffer for training of the intelligent agent. In such embodiments, data stored in the replay buffer may be sampled via prioritized replay based on temporal difference errors.

[0077] At 608, the apparatus may initiate, completely autonomously, a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution. The apparatus may then loop back to 602.

[0078] FIG. 7 is a conceptual data flow diagram 700 illustrating the data flow between different means/components in an exemplary apparatus 702. The apparatus 702 may be a computing device or a system including multiple computing devices.

[0079] The apparatus 702 may include a data fetcher component 708 that retrieves market data from a broker 750. In some embodiments, the data fetcher component 708 may be the data fetcher module 206 described above with reference to FIG. 2. In one embodiment, the data fetcher component 708 may perform the operations described above with reference to 404 in FIG. 4, or 506 in FIG. 5, or 602 in FIG. 6.

[0080] The apparatus 702 may include a prediction component 710 that forecasts price or sentiment related features in the next one or more time steps based on the market data provided by the data fetcher component 708. In some embodiments, the prediction component 710 may be the prediction modules 204 described above with reference to FIG. 2. In one embodiment, the prediction component 710 may perform the operations described above with reference to 422 or 424 in FIG. 4, or 520 in FIG. 5, or 604 in FIG. 6.

[0081] The apparatus 702 may include an agent component 704 that determines a desired portfolio allocation based on the prediction received from the prediction component 710 and the market data provided by the data fetcher component 708. In some embodiments, the agent component 704 may be the agent 216 described above with reference to FIG. 2. In one embodiment, the agent component 704 may perform the operations described above with reference to 410, 412, 414, 416, or 418 in FIG. 4, or 510 in FIG. 5, or 606 in FIG. 6.

[0082] The apparatus 702 may include a trade execution component 712 that interacts with the broker 750 to try to achieve the desired portfolio allocation provided by the agent component 704. In some embodiments, the trade execution component 712 may be the trade execution engine 210 described above with reference to FIG. 2. In one embodiment, the trade execution component 712 may perform the operations described above with reference to 430 or 432 in FIG. 4, or 530 or 532 in FIG. 5, or 608 in FIG. 6.

[0083] The apparatus 702 may include additional components that perform each of the blocks of the algorithm in the aforementioned flowcharts of FIGS. 4-6. As such, each block in the aforementioned flowcharts of FIGS. 4-6 may be performed by a component and the apparatus may include one or more of those components. The components may be one or more hardware components specifically configured to carry out the stated processes/algorithm, implemented by a processor configured to perform the stated processes/algorithm, stored within a computer-readable medium for implementation by at least one processor (e.g., CPUs/GPUs), or some combination thereof.

[0084] FIG. 8 is a diagram 800 illustrating an example of a hardware implementation for an apparatus 702' employing a processing system 814. The apparatus 702' may be the apparatus 702 described above with reference to FIG. 7. The processing system 814 may be implemented with a bus architecture, represented generally by the bus 824. The bus 824 may include any number of interconnecting buses and bridges depending on the specific application of the processing system 814 and the overall design constraints. The bus 824

links together various circuits including one or more processors and/or hardware components, represented by the processor 804, the components 704, 708, 710, 712, and the computer-readable medium / memory 806. The bus 824 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0085] The processing system 814 includes a processor 804 coupled to a computer-readable medium / memory 806. The processor 804 may include multiple processors. The processor 804 is responsible for general processing, including the execution of software stored on the computer-readable medium / memory 806. The software, when executed by the processor 804, causes the processing system 814 to perform the various functions described *supra* for any particular apparatus. The computer-readable medium / memory 806 may also be used for storing data that is manipulated by the processor 804 when executing software. The processing system 814 further includes at least one of the components 704, 708, 710, 712. The components may be software components running in the processor 804, resident/stored in the computer readable medium / memory 806, one or more hardware components coupled to the processor 804, or some combination thereof.

[0086] In the following, various aspects of this disclosure will be illustrated:

[0087] Example 1 is a method or apparatus for portfolio management. The method or apparatus may: retrieve most recent transaction data of one or more financial markets; forecast, using at least one prediction model, one or more prediction features based on the most recent transaction data or sentiment information; determine using a reinforcement learning based intelligent agent, based on the one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets; and initiate a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution.

[0088] In Example 2, the subject matter of Example 1 may optionally include that the one or more financial markets may include at least one of one or more crypto currency markets, one or more foreign exchange markets, or one or more equity markets.

[0089] In Example 3, the subject matter of any one of Examples 1 to 2 may optionally include that the at least one prediction model may include a temporal convolutional neural network.

[0090] In Example 4, the subject matter of any one of Examples 1 to 3 may optionally include that the at least one prediction model may be trained using historical transaction data of the one or more financial markets.

[0091] In Example 5, the subject matter of any one of Examples 1 to 4 may optionally include that the one or more prediction features may include at least one of one or more price related features or one or more sentiment related features.

[0092] In Example 6, the subject matter of any one of Examples 1 to 5 may optionally include that the reinforcement learning based intelligent agent may include a set of feature extraction networks, a set of feature analysis networks, and a set of network improvement networks.

[0093] In Example 7, the subject matter of Example 6 may optionally include that the outputs from the at least one prediction model may be fed directly to at least one of the set of feature extraction networks or the set of feature analysis networks.

[0094] In Example 8, the subject matter of any one of Examples 1 to 7 may optionally include that the reinforcement learning based intelligent agent may be trained using historical transaction data of the one or more financial markets and historical outputs from the at least one prediction model.

[0095] In Example 9, the subject matter of any one of Examples 1 to 8 may optionally include that the reinforcement learning based intelligent agent may include a replay buffer for training of the reinforcement learning based intelligent agent, wherein data stored in the replay buffer is sampled via prioritized replay based on temporal difference errors.

[0096] A person skilled in the art will appreciate that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0097] It is understood that the specific order or hierarchy of blocks in the processes / flowcharts disclosed is an illustration of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of blocks in the processes / flowcharts may be rearranged. Further, some blocks may be combined or omitted. The

accompanying method claims present elements of the various blocks in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[0098] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean “one and only one” unless specifically so stated, but rather “one or more.” The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects. Unless specifically stated otherwise, the term “some” refers to one or more. Combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” include any combination of A, B, and/or C, and may include multiples of A, multiples of B, or multiples of C. Specifically, combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” may be A only, B only, C only, A and B, A and C, B and C, or A and B and C, where any such combinations may contain one or more member or members of A, B, or C. All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the claims. The words “module,” “mechanism,” “element,” “device,” and the like may not be a substitute for the word “means.” As such, no claim element is to be construed as a means plus function unless the element is expressly recited using the phrase “means for.”

CLAIMS

WHAT IS CLAIMED IS:

1. A method of portfolio management, the method comprising:
retrieving most recent transaction data of one or more financial markets;
forecasting, using at least one prediction model, one or more prediction features based on the most recent transaction data;
determining using a reinforcement learning based intelligent agent, based on the forecasted one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets; and
initiating, completely autonomously, a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution.
2. The method of claim 1, wherein the at least one prediction model comprises a temporal convolutional neural network.
3. The method of claim 1, wherein the at least one prediction model is trained using historical transaction data of the one or more financial markets.
4. The method of claim 1, wherein the one or more prediction features comprise at least one of one or more price related features or one or more sentiment related features.
5. The method of claim 1, wherein the reinforcement learning based intelligent agent comprises a set of feature extraction networks, a set of feature analysis networks, and a set of network improvement networks.
6. The method of claim 5, wherein outputs from the at least one prediction model are fed directly to at least one of the set of feature extraction networks or the set of feature analysis networks.

7. The method of claim 1, wherein the reinforcement learning based intelligent agent is trained using historical transaction data of the one or more financial markets and historical outputs from the at least one prediction model.
8. The method of claim 1, wherein the reinforcement learning based intelligent agent comprises a replay buffer for training of the reinforcement learning based intelligent agent, wherein data stored in the replay buffer is sampled via prioritized replay based on temporal difference errors.
9. An apparatus for portfolio management, comprising:
 - a memory; and
 - at least one processor coupled to the memory and configured to:
 - retrieve most recent transaction data of one or more financial markets;
 - forecast, using at least one prediction model, one or more prediction features based on the most recent transaction data;
 - determine using a reinforcement learning based intelligent agent, based on the forecasted one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets; and
 - initiate, completely autonomously, a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution.
10. The apparatus of claim 9, wherein the at least one prediction model comprises a temporal convolutional neural network.
11. The apparatus of claim 9, wherein the one or more prediction features comprise at least one of one or more price related features or one or more sentiment related features.
12. The apparatus of claim 9, wherein the reinforcement learning based intelligent agent comprises a set of feature extraction networks, a set of feature analysis networks, and a set of network improvement networks.

13. The apparatus of claim 12, wherein outputs from the at least one prediction model are fed directly to at least one of the set of feature extraction networks or the set of feature analysis networks.
14. The apparatus of claim 9, wherein the reinforcement learning based intelligent agent comprises a replay buffer for training of the reinforcement learning based intelligent agent, wherein data stored in the replay buffer is sampled via prioritized replay based on temporal difference errors.
15. An apparatus for portfolio management, comprising:
means for retrieving most recent transaction data of one or more financial markets;
means for forecasting one or more prediction features based on the most recent transaction data;
means for determining, based on the forecasted one or more prediction features and the most recent transaction data, a portfolio weight distribution over a plurality of financial investment products that are traded on the one or more financial markets; and
means for initiating, completely autonomously, a set of transactions on the plurality of financial investment products to achieve the portfolio weight distribution.
16. The apparatus of claim 15, wherein the means for forecasting comprises a temporal convolutional neural network.
17. The apparatus of claim 15, wherein the one or more prediction features comprise at least one of one or more price related features or one or more sentiment related features.
18. The apparatus of claim 15, wherein the means for determining the portfolio weight distribution comprises a set of feature extraction networks, a set of feature analysis networks, and a set of network improvement networks.
19. The apparatus of claim 18, wherein outputs from the means for forecasting are fed directly to at least one of the set of feature extraction networks or the set of feature analysis networks.

20. The apparatus of claim 15, wherein the means for determining the portfolio weight distribution comprises a replay buffer for training of the means for determining the portfolio weight distribution, wherein data stored in the replay buffer is sampled via prioritized replay based on temporal difference errors.