＜ 修 士 論 文 ＞

強化学習を利用した暗号資産の価格予測

滋 賀 大 学 大 学 院
デ ー タ サ イ エ ン ス 研 究 科
デ ー タ サ イ エ ン ス 専 攻

修了年度：２０２２年度
学籍番号：６０２１１２６
氏　　名：長澤　秀紀
指導教員：笛田　薫
提出年月日：２０２３年１月９日

目次

1 序論

1.1　研究の背景・目的

金融商品の価格予測は、実務的にも非常に興味深い研究である。なぜなら金融商品価格の予測可能性は、個人の利益だけでなく世界中の経済・社会活動に影響を及ぼし非常に重要な役割を担っているからである。こうした重要性により、この問題に対して数多くの実務者やアカデミアが取り組んできた。特に実務面では金融商品の価格予測に対して過去のヒストリカルデータを利用したテクニカル分析による予測が盛んに行われているが、このような予測手法に対して Fama (1965)は、「株価はランデムウォークであり連続した価格の変動は独立で同一に分布する確率変数のため、ヒストリカルデータの中に将来の動きに関する情報が含まれていると仮定し、過去の価格行動の『パターン』は将来も繰り返される傾向があるとするテクニカル分析は株式市場の投資家にとって何の価値もない」とした。その後、現代の資本市場理論の基礎となっている Efficient-market hypothesis (EMH) によって非ファンダメンタル分析であるテクニカル分析は根拠を奪われた。しかし、Menkhoff (2010)が 5 カ国 692 名のファンドマネージャを調査した結果、多くがテクニカル分析によって銘柄選定を行なっていた。特に数週間の予測期間と短期的な運用期間である場合はファンダメンタル分析よりもテクニカル分析がファンドマネージャにとって重要な分析形式であるとしている。多くの研究者から批判的な意見のあるテクニカル分析であるが、金融実務者の中ではテクニカル分析を根拠とした投資活動が行われており、テクニカル分析の有用性の観点では Park et al. (2007)が合計 95 の論文を調査し、その中で 56 テクニカルトレード戦略が有効であると示している。

なぜテクニカル分析が金融商品価格の予測に利用できるか理論的側面から考えると、次のようなことが考えられる。

- Grossman et al. (1980)によると資産価格モデルの主流である CAPM 等の均衡モデルは均衡に至る過程に興味はない。そのため価格形成を行う過程で均衡が変化する際にテクニカル分析の有効性が生まれるのではないかと考えられる。
- Bradford et al. (1990)によると金融市場にはノイズトレーダーと呼ばれる非合理的な参加者が取引に参加する。この非合理な投資家が合理的な価格と乖離する予測不可能なリスク・リターンを生み出し、裁定取引者のような合理的な投資家の投資行動を制限する。

このことから短期的にノイズトレーダーの投資行動がマーケットインパクトを持つ事で、金融商品の価格形成がランダムウォークでは無く、一定のパターンを持ち合理的価格水準から乖離する。それによりヒストリカルデータを用いた金融商品の価格予測が可能になると考えることできる。そのため、価格予測を行いやすいマーケットの条件としては「ノイズトレーダーが参入しやすい」「マーケット参加者に合理的的な参加者が少ない」「ボラティリティが高く合理的価格水準が判断しにくい」が適切である。Ito (1999)はボラティリティの

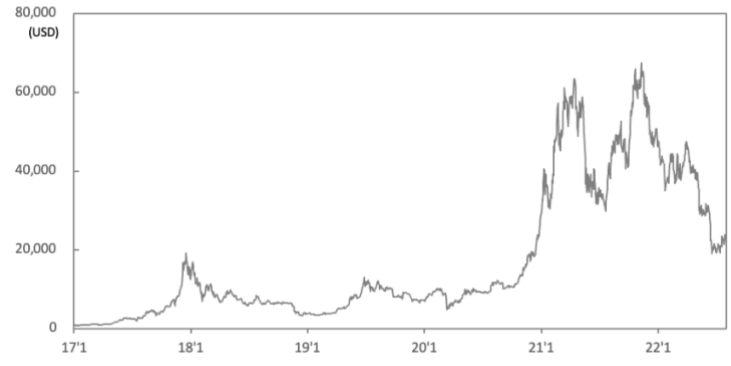
低い先進国よりもボラティリティが高い新興国の株式市場においてテクニカルトレーディングの有効性があり、利益率が高かったことを示している。

多くのノイズトレーダーが参入しボラティリティが高い商品は、近年規模を大幅に拡大し注目されている Bitcoin を代表とする暗号資産と考えられる。暗号資産の原初は、Nakamoto (2008)が提唱した Peer-to-Peer の電子マネーの仕組みで匿名性と安全性を担保したプルーフオブワークを使って記録された電子取引のシステムである。これは信用ではなく暗号化された証明に基づく電子取引システムである。これにより従来の電子取引では不完全だった二重支払防止の対策、電子取引を処理する第三者機関である金融機関を通さないフレームワークが作られ、取引に必要な仲介コストの減少や顧客と販売先の直接取引を可能とした。この発表後、Bitcoin-Client が作られ同時に Bitcoin の発行と、取引のための実際のプラットフォームが作られた。その後 2012 年には、BitCoin の標準化・保護・普及を活動する Bitcoin 財団の発足、2013 年に CoinBase が 1Bitcoin を 22 ドル強の単価で販売することが発表された。この頃には Bitcoin とは異なる暗号資産 Ethereum や Ripple 等が発行された。同時期に各国の法規制が整備され始め、タイでは外国為替政策局が Bitcoin を法的枠組みがなく違法としたが、米国の一部州内では Bitcoin が「通貨または貨幣の一形態」であり連邦法の定義に当てはまるという裁判所の判決が下された。ドイツでは暗号資産を法的・財務的な影響を及ぼすと金融商品として認め法整備を進めた。翌 2014 年には TeraExchage が米国商品先物取引委員会(CFTC)から Bitcoin の価格を原資産とする店頭スワップ商品の承認を受け、Bitcoin が原資産のデリバティブ商品が登場した。2015 年、暗号資産が G7 エルマウ・サミット首脳宣言や金融活動作業部会(FATFA)ガイダンスでの議題として取り上げられ、2016 年には日本で暗号資産の貨幣的価値が認められ暗号資産にかかる法改正が行われた。

このように暗号資産全体に法規制とマーケット規模拡大の流れを辿る中、暗号資産の新しいサービス・取引所・新通貨の発行・価格上昇により「Bitcoin Millionaires」と呼ばれる Bitcoin により多額の資産を形成した人々が多く登場した。このことから暗号資産への期待感が大きく膨れ上がり 2017 年 8 月に 4,000 ドル程度の価格だった Bitcoin はシカゴマーチャンダイル取引所(CME)での先物上場のニュースを追い風に多くの投機的取引を伴い同年 12 月には過去最高値の水準 19,000 ドルを超えて急上昇した。これはバブル的相場となりその後は反転、2017 年 12 月末日の価格は 14,000 ドルを下回って取引を終えた。2018 年には今まで拡大傾向であった仮想通貨に対して、日本最大の取引市場である Coincheck から 500 億円規模の資産がハッキングによって盗難。過大な広告が行われていた ICO に対して Facebook (現 Meta)・Google (現 Alphabet) 等の大手インターネット企業が広告の投稿の禁止、米国公的機関では証券取引委員会(SEC)や商品先物取引委員会(CFTC)による積極的な取り組み、暗号資産 AML/CFT といった匿名性に付随する問題点に対して米国司法省が積極的な規制を行ったことで大きな逆風を受け 2018 年末には Bitcoin 価格は前年末から 80%程度下落し 3,700 ドル程度となった。2019 年価格は上昇し 100%程度上昇の 7,500 ド

ル、2020 年には COVID-19 による各国の景気刺激策による金余り状況などが追い風となり、各金融資産の価格上昇の波を受け 280%程度上昇し 28,000 ドルを超える水準で取引を終えた。2021 年も価格上昇は継続し、一時 69,000 ドルを超えたがバブルとも取れるこの動きは崩壊し 2022 年 11 月末時点では最高値から 75%程度下落し 17,000 ドルとなった。2017 年以降の Bitcoin の価格推移を図 1.1 に示す。表 1.1 に「日経平均 225」「米国 S&P500 指数」「インド SENSEX30」「ドル円レート」の 2017 年から 2021 年までの年間騰落率を示すが、Bitcoin の価格変化率は他の資産と比較しても大きいことが分かる。

(図1.1) BitCoin価格推移



出所） Investing.com 2017/1/1～2022/7/31

(表1.1) 各アセット別年間騰落率

年間当落率	2017	2018	2019	2020	2021
BTC/USDT	1338%	-73%	94%	302%	60%
NKY225	18%	-13%	20%	18%	4%
SPX500	19%	-7%	30%	16%	27%
SENSEX30	28%	6%	14%	16%	22%
USD/JPY	-4%	-3%	-1%	-5%	11%

出所） Investing.com 2017/1/1～2021/12/31

Bitcoin は価格変化率が大きく、前述した合理的な投資家が参入しにくいマーケットである。そのため、過去のヒストリカルデータを活用した価格予測が有効な条件として考えられる「均衡価格の変化が大きい」「ノイズトレーダーによる不合理な価格決定が定常的に行われ、合理的な投資可能行動を制限する」を満たしている可能性が高い。過去のヒストリカルデータを利用した価格予測が他資産よりも行いやすい環境であると考え、本稿では Bitcoin のマーケットに対してヒストリカルデータのみを使い価格予測モデル作成の実証を行う。

本稿ではマーケットには種々のノイズトレーダーが参加しており、ノイズトレーダーによって短期的なマーケットの価格が形成されると考える。この非合理的投資家の投資判断基準は各々異なり、より多くの非合理的な投資家が投資判断の根拠としているマーケット

指標がマーケットの価格を形成すると仮定する。価格を形成する非合理的な投資家群は各マーケットタイミングに同質ではなく、マーケットセンチメントによって重要なマーケット指標が変化する。そのため価格の予測においては、マーケット状態を考慮し適切なモデルを選択して予測をする必要がある。具体的な予測ステップは各時点のマーケット環境をマルチクラスに分類し、分類結果をもとに適切なマーケット予測モデルを利用し予測を行う。

先行研究にはヒストリカルデータのみをインプットデータとした価格予測や、予測精度をあげるためにヒストリカルデータに加えてマーケット参加者の心理を考慮しSNSやニュースを説明変数に加えた価格予測の研究は多くあるが、ノイズトレーダーによる価格形成の観点から各マーケット環境を分類し、そのマーケット環境におけるモデルを利用し価格予測を行うケースは少ない。先行研究を踏まえた本稿の位置付けは、全てのマーケット環境を一律のモデルで表現するのではなく、異なるモデルで表現することでマーケットに携わる実務者にとって直感的であり受け入れやすいモデルの構築を可能とすることである。

1.2　本論文の構成

本稿の構成は、第2章において暗号資産の価格予測に関する先行研究を紹介する。その中でも従来型の機械学習と深層学習を用いた価格予測と価格予測に焦点を当てる。

第3章では使用するデータの説明と基礎分析を行う。利用するBitcoinのヒストリカルデータの基礎統計量を調査しデータの構造を示し、テスト期間における正解率ベースでの収益率パフォーマンスシミュレーションを行う。これにより当該資産の収益性の検証が可能となる。特にマーケットがランダムウォークの前提に立った場合の観点で収益性の整理を行う。そして、ヒストリカルデータよりモデルに利用する特徴量の算出手法を示す。

次に第4章では本稿におけるBitcoinマーケットの価格予測手法と、強化学習モデルによる学習方法とクラスタリング手法について示す。本稿ではBitcoinのヒストリカルデータを状態として捉え、ヒストリカルデータより作成した各特徴量を強化学習エージェントの学習環境に変換し、タイムステップの都度エージェントによる投資判断を行うモデルを作成する。初めに強化学習を用いて作成した暗号資産の価格予測を行うエージェントがランダムウォークと比較して有効な予測精度が出せるかの検証を行う。ランダムウォーク対比での実証を行う理由として、先行研究の多くが従来の機械学習による価格予測・深層学習でもLSTM 等を利用したモデルが多く、強化学習を適応させた事例が少ない。そのため強化学習を利用しエージェントによる取引判断が可能か実証する必要があると考える。本稿ではモデルによる正解率の検証・バイアンドホールド戦略と比較した場合の収益性の比較を行う。

第 5 章ではマーケットの状態をマルチクラスに分類し各クラスに適切なエージェントを配置することで、各状態にあったエージェントが投資判断を行うモデルについて検証を行う。理想的なエージェントは全てのマーケット環境の状態に対して、どのような値であっても最適投資判断を行えるエージェントであるが、前述したようにマーケットには種々のノ

イズトレーダーがおり投資判断の基準は各々異なる。つまり、あるタイムステップにおいて短期的なマーケットの価格はその時点の最も優勢なノイズトレーダーが重視した状態に基づき形成されると仮定される。そのため単独のエージェントで全てのマーケットの投資判断を行うと表現できる範囲は限られてしまい、全期間を通じて有利なノイズトレーダーを判断するエージェントになる可能性が考えられる。本稿ではこの仮定を考慮し各マーケット環境にとって適切な強化学習エージェントを配置して、複数のエージェントによってマーケット予測をさせるマルチエージェントモデルを実証する。

2 先行研究

2.1　暗号資産に関する先行研究

本章では暗号資産の価格予測に関する先行研究について説明する。はじめに暗号資産のヒストリカルデータより価格予測性があるか、すなわち暗号資産のマーケットは効率的市場仮説の観点から見て効率性が無いことを示す。Urquhart (2016)は Fama (1965)が提唱したファイナンスの基礎である効率的市場仮説(EMH)の観点でビットコイン市場が弱度（ウィークフォーム）の効率性を持つかを Ljung (1978)の「Ljung-Box 検定」、Wald et al. (1940)の「ランズ検定」、Kim (2009)の「ワイルドブートストラップ AVR 検定」等を用い検定を行なった。その結果ビットコインの非効率性は非常に強く EMH における弱度の効率性は低いことを示唆した。暗号資産市場に対して効率性が示されない理由として高いボラティリティが考えられる。ボラティリティが高いことで、合理的な投資家は参入しにくくノイズトレーダーが多く参入することで効率性が低くなる。ボラティリティが高くなる理由として Yermack (2015)は「ビットコインは所有者に対して高いリスクを齎し、一般的な通貨や金との相関は無相関であり、従来の通貨の定義である『交換手段』『価値の貯蔵』の基準を満たしておらず投機的な資産として位置付けしているように見える」という考察を出しており、ボラティリティバブルが起こりやすいことが背景にあると考えられる。

このように BitCoin はノイズトレーダーが価格形成に強い影響を与えるマーケットであるため、価格予測を行える可能性が高く関連する研究が多い。Fang et al. (2022)の暗号資産に対するサーベイでは 2013 年～2021 年に発表された 146 の暗号資産取引に関する論文のうち統計的手法・機械学習のカテゴリに関する論文は全体の中で約 70%と関心の高い項目となっている。本章では機械学習・深層学習の価格予測に絞り第 2 節で従来の機械学習を利用した暗号資産価格予測、第 3 節で深層学習を利用した暗号資産の価格予測について述べる。

2.2　機械学習を利用した暗号資産の価格予測

暗号資産に対する機械学習を利用した価格予測は数多く発表されている。従来の機械学習手法では Sebastião et al. (2021)は線形モデル、SVM, Random Forest とアンサンブル学習を利用して暗号資産の主要な三通貨 Bitcoin,Litecoin,Ethereum に対して取引戦略の収益性を検証した。Sebastião はタイムステップの頻度を日次とし、次のステップの価格を予測するモデルと、バイナリの売買シグナルを生成する分類モデルでの予測を行った。取引所の情報 31 個、ブロックチェーンに関する情報 12 個、曜日情報 7 個の計 50 個の変数を利用して予測をおこなっている。テスト期間におけるバックテストは複数のモデルを組み合わせて予測を決定するアンサンブル学習を行い、Litecoin と Ehtereum に適用した場合のパフォーマンスは手数料控除後でそれぞれ 5.730%と 9.622%であった。一方で Bitcoin の年率

リターンは−23.66%と他通貨に比べてパフォーマンスが下がった。

Madan et al. (2015) は、SVM, Random Forest, Binomial GLM を用い、Blockchain のデータを利用して Bitcoin 価格を予測した。その結果 98.7%の精度であったが、クロスバリデーションなどは行なっておらず、作成されたモデルの汎用性があるかなど一般化に対する問題は残っている。

Branwal et al. (2019) は複数のアルゴリズムをスタッキングしモデルの比較を行った。利用した特徴量は、外部データを用いずトレンド、モメンタム、ボリューム、ボラティリティ等のテクニカル指標で、精度 54%のパフォーマンスを出している。

その他にも機械学習モデルを利用した暗号資産の価格予測は多く研究されている、先行研究の中には Madan et al. (2015)のように正解率が非常に高いモデルも見られるが汎用性の観点では一般化されていないケースが見られる。

2.3　深層学習を利用した暗号資産の価格予測

従来の機械学習手法からより高い予測精度が期待できる DeepNeuralNetwork (DNN)を利用した暗号資産の価格予測も盛んに研究されている。

Nakano et al. (2018) は Bitcoin の 15 分毎時系列取引データとテクニカル指標の MACD, RSI, OVA 等のテクニカル指標を学習データとしてロング／ニュートラル／ショートのマルチクラスの分類を DNN で行い、売買のシグナルを出した。この中で複数の ANN モデルやデータセットを用意し、何が性能向上の鍵となるか分析している。特に非定常の金融時系列データの分類において、様々なテクニカル指標を用いることでオーバーフィッティングを防ぎ、トレーディングパフォーマンスを向上させる可能性があることを数値的に示している。

MacNally et al. (2018) はリカレントニューラルネットワーク（RNN）と長短期記憶（LSTM）ネットワークの実装を行い LSTM は 52%の分類精度と RMSE は 8%を達成した。深層学習モデルと時系列予測の ARIMA モデルと比較し非線形ディープラーニング手法が、ARIMA 予測を上回る結果を示した。

Shahbazi et al. (2021) は Litecoin と Monero の価格予測のために、ブロックチェーンフレームワークと統合された強化学習を予測アプローチに活用した。

それ以外にも深層学習による暗号資産の予測は畳み込みニューラルネットワーク(CNN)や深層残差ネットワーク(ResNet)等の組み合わせによる予測モデルの作成など多岐に渡る手法で試みられている。一方で、Ji (2019)が行った深層学習手法による暗号資産価格回帰・分類の予測モデルを比較したテストでは明らかに高いパフォーマンスを示すものはなく、予測精度向上のためには外部データからの特徴量エンジニアリングなどの必要性を示している。

3 データについて

3.1　利用するデータ

今回の研究で利用するデータは実際にトレードが行われる取引所から提供されているデータを利用する。データ取得元は 2022 年 3 月時点で世界最大の取引所 Binance である。Binance は暗号資産を 400 近く取り扱っており、取引高は 24 時間で邦貨換算約 2.7 兆円程度、Bitcoin の流動性も他取引所よりも高い。利用するヒストリカルデータの期間は 2018 年 1 月 1 日～ 2022 年 3 月 31 日とし Binance より提供されている api から取得した。取得したヒストリカルデータは以下の期間に分割し「学習・検証・テスト」期間として利用する。

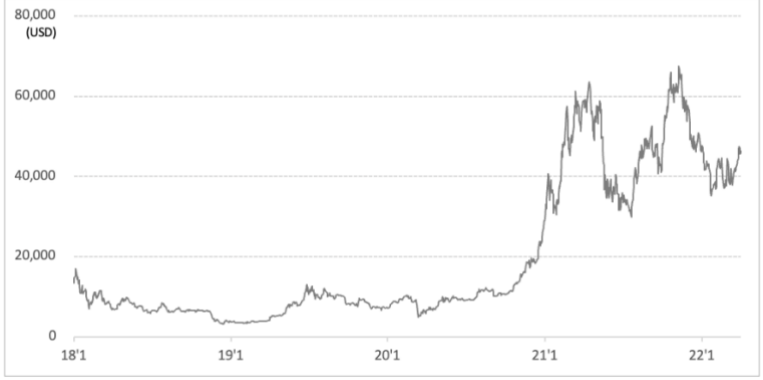
・学習期間:2018 年 1 月 1 日～2021 年 8 月 31 日

・検証期間:2021 年 9 月 1 日～2022 年 1 月 14 日

・テスト期間:2022 年 1 月 15 日～2022 年 3 月 31 日

取引所から取得したデータは一般的な candlestick で各時間における始値・高値・安値・終値・売買高となる。時間足の頻度は 1 分、5 分、15 分、30 分、60 分、日次である。取得データのうち、売買高については同一暗号資産であっても取引所によって分布が異なる可能性があるため利用を行わず、ヒストリカルデータによるマーケット予測を行うため外部データを利用せずに api から取得したデータのみを利用する。この期間における日次 Bitcoin の終値価格推移は図 3.1 となる。

(図3.1)BitCoin価格推移



出所）Binance 2018/1/1~2022/3/31

3.2　データ基礎調査

取得したヒストリカルデータの各足終値から価格の変化率を取得する。基礎調査時の変化率は対数を用いて以下で定義する。

$$r_t = ln(\frac{P_t}{P_{t-1}}) \tag{3.1}$$

該当期間における各足の変化率に関する統計量を算出し、現実的なトレーディングの観点で予測に利用する時間足データを決定する。各データに対して「価格変化率の平均」「標準偏差」「価格変化率の絶対値の平均」「歪度」「尖度」「ヒストリカルボラティリティ」を算出した。算出する値を以下で定義する。

<u>価格変化率の絶対値の平均</u>

$$E = \frac{1}{N}\sum_{t=1}^{N}|r_t| \tag{3.2}$$

<u>歪度</u>

$$skewness = \frac{1}{\sigma^3 N}\sum_{t=1}^{N}(r_t - \bar{r})^3 \tag{3.3}$$

<u>尖度</u>

$$kurtosis = \left(\frac{1}{\sigma^4 N}\sum_{t=1}^{N}(r_t - \bar{r})^4\right) \tag{3.4}$$

<u>ヒストリカルボラティリティ</u>

$$HV = \sqrt{365 \times 24 \times \frac{60}{M} \times \frac{1}{N}\sum_{t=1}^{N}(r_t - \bar{r})^2} \tag{3.5}$$

Nは期間内の各足の合計本数。Mは各足の取得期間(分)を示す。固定値の 365 は一般的にヒストリカルボラティリティは指定期間の年率換算で算出し、暗号資産の取引所は基本的に休場がないので 365 営業日換算とした。今回基礎調査を行った 1 分、5 分、15 分、30 分、60 分、日足の基礎調査の結果を表 3.1 に示す。

(表3.1) 各足 対数変化率統計値

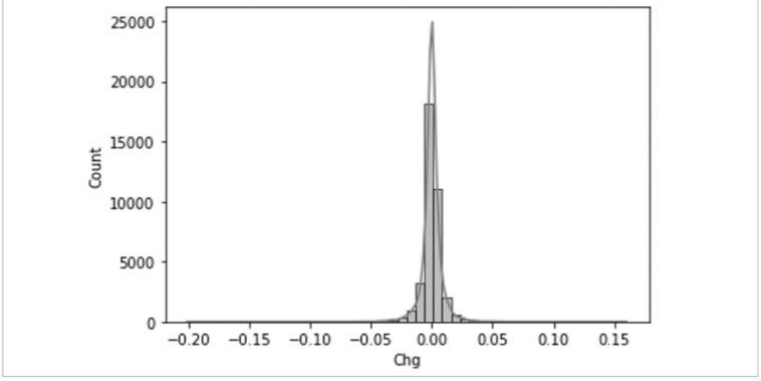
	mean	std	min	max	Historical Volatility	Skew	Kurt	Change (Abs)
min	5.39E-07	0.001	-0.075	0.072	0.872	0.127	103.772	0.0006
min5	2.72E-06	0.002	-0.107	0.169	0.859	0.203	109.857	0.0015
min15	8.18E-06	0.004	-0.141	0.203	0.838	0.172	75.796	0.0026
min30	1.64E-05	0.006	-0.181	0.159	0.825	-0.289	46.778	0.0036
min60	3.27E-05	0.008	-0.201	0.160	0.811	-0.524	34.826	0.0051
day	8.00E-04	0.040	-0.502	0.178	0.781	-1.288	16.579	0.0273

出所）Binance 2018/1/1~2022/3/31

取引所を Binance の BTC/USDT 先物とした場合 Bid/Ask のテイカー手数料は片道 0~0.018%であるため、手数料率を 0.018%とし往復時の手数料によるリターン減衰は

0.036%となる。タイムステップ毎に売買を行うことを想定した場合、価格変化率の絶対値の平均が手数料率を超える必要がある。30 分以下のタイムステップでは価格予測を常に正解したとしても、手数料率がリターンを超えるため利益を産み出すことができない。そのため、対象となるタイムステップについてトレーディングによる収益とデータレコード数の観点で 60 分足を対象とした。60 分足の価格変化率の分布は図 3.2 となる。尖度は 34.826 となり、強い凸型分布となり変化率 0 近辺での変化が多く予測モデルを作成しても手数料率で損失が出る可能性が高い。そのため手数料率を含めたトレーディングでのシミュレーションを次節で行い、ランダムウォークによりモデルがマーケットの予測を行えなかった場合・マーケットが予測できた場合の損益状況を検証する。

(図3.2)60分足　対数変化率ヒストグラム

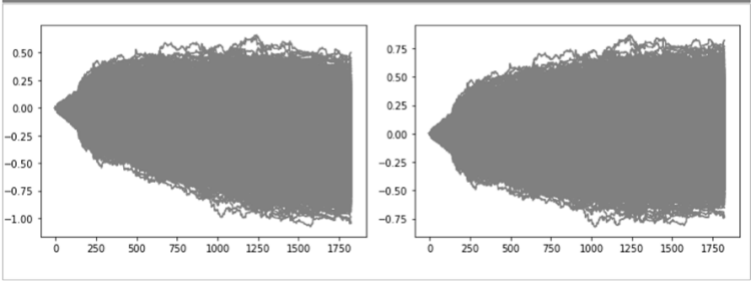


出所）Binance 2018/1/1～2022/3/31

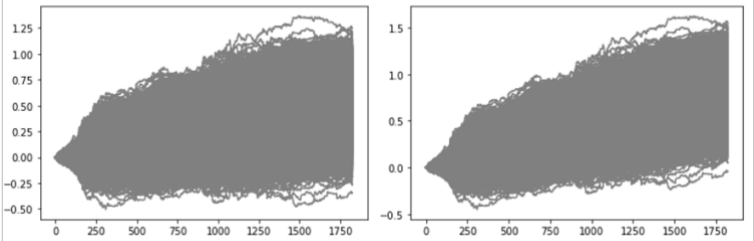
3.3　収益率シミュレーション

次に対象資産をテスト期間でトレーディングした際の収益率のシミュレーションを行う。このシミュレーションを用い、モデル運用時に利益を上げるために必要なモデル精度を把握する。シミュレーション期間のステップ毎に売買を行い、最終ステップ時点で正解率(50%, 55%, 57%, 60%)の際に各正解率における収益率の信頼区間を算出する。対象となるシミュレーション期間は本稿のテスト期間である 2022 年 1 月 15 日 9 時～2022 年 3 月 31 日 8 時の 1824 ステップとし指定の正解率になるよう時系列に沿ってランダムな売買を行う。収益率は各ステップでの総収益額をシミュレーション開始期間の始値で除した値である。手数料率は Binance のテイカー先物取引手数料の片道 0.018%とする。売買の手数料は連続したタイムステップで売買取引判断が異なる場合に計上する。シミュレーション回数は正解率毎に 10,000 回行った。シミュレーション結果の各正解率による手数料を含めた収益率・手数料を含めない収益率を図 3.3-6 に示し、当該期間における収益率の 90%～99%信頼区間を表 3.2 に示す。マーケットがランダムウォークであり、ヒストリカルデータを用いた予測可能性がない場合は図 3.3 の正解率 50%が該当する。

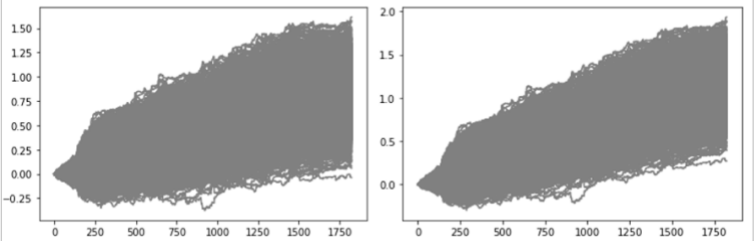
（図3.3）正答率50%（手数料含む/無手数料）収益率推移



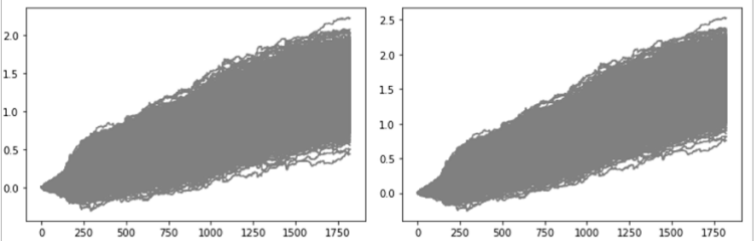
（図3.4）正答率55%（手数料含む/無手数料）収益率推移



（図3.5）正答率57%（手数料含む/無手数料）収益率推移



（図3.6）正答率60%（手数料含む/無手数料）収益率推移



（表3.2）正答率によるテスト期間収益率信頼区間（手数料含む）

Accuracy	Mean	Std	CI90	CI95	CI99
50%	-0.307983	0.214522	-0.661~0.045	-0.728~0.112	-0.86~0.244
55%	0.511563	0.214661	0.158~0.865	0.091~0.932	-0.041~1.064
57%	0.847292	0.209909	0.502~1.193	0.436~1.259	0.307~1.388
60%	1.334557	0.208622	0.991~1.678	0.926~1.743	0.797~1.872

マーケットがランダムウォークでヒストリカルデータを利用した機械学習では予測不可能とした場合(正解率 50%)収益率の 95%信頼区間は−72.8%~11.2%、正解率の信頼区間は47.7%~52.3%となる。次章で作成するモデルのシミュレーション結果と同等のテスト期間でバックテストを行い、上記のランダムウォークによる正解率・収益率の信頼区間を超えるパフォーマンスを出すことができるかを検証する。

3.4　特徴量について

本モデルに用いた特徴量は、Bitcoin の価格ヒストリカルデータより算出した値のみとする。算出した特徴量は以下の通りである。

直前タイムステップの価格変化率

$Openprice$はタイムステップの始値、$Closeprice$はタイムステップの終値を示す。前回のタイムステップにおける価格の変化率を表す。

$$PriceChange = \frac{ClosePrice_{t-1}}{OpenPrice_{t-1}} - 1 \tag{3.6}$$

直前タイムステップの分足標準偏差

mは直前タイムステップの１分足のヒストリカルデータを示す。前回のタイムステップにおける価格の変化率の分布を表す。

$$Sigma = \sqrt{\frac{1}{60}\sum_{m=1}^{60}(r_m - \bar{r})^2} \tag{3.7}$$

直前タイムステップの１分毎価格上昇回数

直前タイムステップのリターン上昇回数をカウント。１時間足内での１分足リターンが上昇している回数の総和を取る。

$$UpCount = \sum_{m=1}^{60} I\{r_m > 0\} \tag{3.8}$$

IはインジゲーターファンクションでIが成立する場合は 1、不成立の場合は 0 を取る。

直前タイムステップの１分毎価格下落回数

直前タイムステップのリターン下落回数をカウント。1 時間足内での 1 分足リターンが下落している回数の総和を取る。

$$DownCount = \sum_{m=1}^{60} I\{r_m < 0\} \tag{3.9}$$

Simple Moving Average (SMA)

SMAは終値の単純移動平均で重みを持たずに計算を行う。$SMA(n)$ は以下と定義する。

$$SMA_t = \frac{1}{n}\sum_{i=1}^{n} P_{t-i} \tag{3.10}$$

本稿では、現在価格とSMAの乖離率を利用する。

Moving Average Convergence Divergence (MACD)

$MACD$は短期と長期の指数移動平均(EMA)から計算される。時刻tにおけるEMAは以下と定義する。

$$EMA_t = \left[\frac{2}{n} \times (P_t - EMA_{t-1})\right] + EMA_{t-1} \tag{3.11}$$

Pは原資産価格、nはEMAの期間数である。スタートのEMAは期間nの単純移動平均とする。$MACD$に利用される短期EMA期間は 12、長期EMA期間は 26 とし、短期EMAから長期EMAを減算して算出し、売買タイミングに用いられる$MACD$の指数は算出した$MACD$から$MACD$のEMAを取得する。一般的に利用されるシグナルは 9 日間の$MACDEMA$となる。

$$MACD_t = ShortEMA_t - LongEMA_t \tag{3.12}$$

$$MACD\ Signal_t = \left[\frac{2}{n} \times (MACD_t - MACDEMA_{t-1})\right] + MACDEMA_{t-1} \tag{3.13}$$

本稿で利用する特徴量としての$MACD$は、$MACD$から$MACD\ Signal$を減算した以下の値である。

$$X_t = MACD_t - MACD\ Signal_t \tag{3.14}$$

4　分析

4.1　　モデル概要

本研究における主な目的は

1.過去のヒストリカルデータを用いた暗号資産価格の予測可能性の検討

2.運用により利益をあげるモデルの構築

の２点となる。そのため、

・予測正答率がランダムウォークを上回るモデルの構築

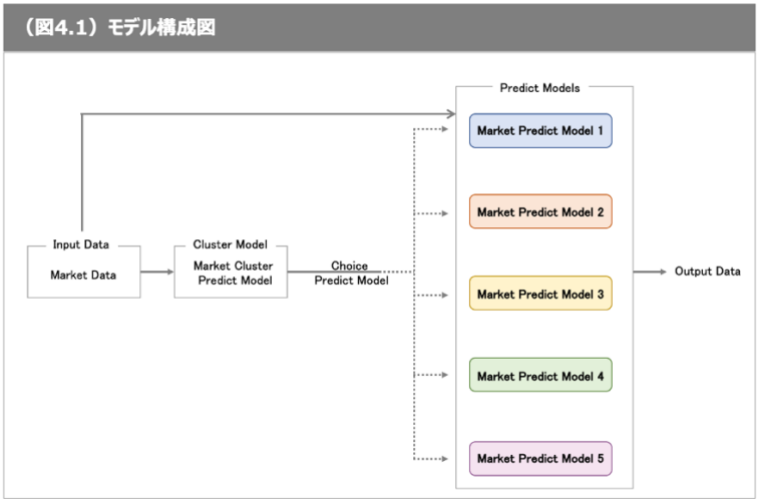
・バイアンドホールド戦略を上回るリターンを得るモデルの構築

を前提とした暗号資産トレーディングモデルを作成する。暗号資産の予測可能性を示すためにモデルの出力内容を$t-1$の終値からtの終値が上昇するか下落するかのバイナリの出力とし、正解率(Accuracy)を計測する。$Accuracy$はクラス分類に利用される一般的な評価指標であり、計測方法は以下で定義される。

$$Accuracy = \frac{TP + TN}{(TP + TN + FN + FP)} \tag{4.1}$$

マーケットをランダムウォークと仮定した場合の$Accuracy$は 0.50 となり過去のヒストリカルデータからマーケットを予測することが出来ない事となる。マーケット予測モデルは強化学習を利用する。各時刻tのヒストリカルデータから算出した特徴量を状態とし、エージェントは、与えられた状態に対して「購入」「売却」のいずれかを行動する。またエージェントに与えられる状態はエージェント毎に異なり、作成した特徴量を 3 項目以上含む組み合わせの状態を与える。与えられた状態においてランダムウォークの正解率・バイアンドホールド戦略を超えるパフォーマンスを出すモデルが作成できるか実証する。エージェントに与える状態についてエージェント毎に異なる理由は今回構築するマーケット予測モデルの構成に起因する。構築するマーケット予測モデルの構成図は図 4.1 の通りである。

（図4.1）モデル構成図



本稿のモデルではタイムステップの都度マーケットの状態を識別する。その結果に基づき識別したマーケットの状態において最適な投資判断ができる強化学習エージェントが予測を行う。具体的には、マーケットの全ての特徴量を利用してマルチクラス分類のモデルを作成し、作成したクラス毎に強化学習エージェントの推論結果を評価する。各クラスの検証期間で正解率の高いエージェントをクラス毎に投資判断を行うエージェントとして利用する。

理想的な強化学習モデルのエージェントは全てのマーケット環境に対して正確なマーケットを予測し投資行動を行う単独のエージェントであるが、単独のエージェントで表現することができないと考える理由が大きく分けて２点ある。それは、

1.全ての投資家が利用するマーケット環境の状態を用意することが不可能である。

2.マーケット状態毎に価格形成を担うノイズトレーダーの判断基準が異なる。

という点である。1,2 点目ともにマーケット参加者の多様性に関する点だが、1 点目については理想的なエージェントを作成するのに非常に高いハードルとなる。マーケットの価格形成には、ヒストリカルデータだけではなく対象資産外のさまざまな要因が寄与している。これはいうまでもなく、別アセットの価格動向・暗号資産の認知度の増減・世界全体の景気動向等、数えることが出来ないほど多岐に渡り、月の満ち欠けを考慮するようなアノマリーを重視する投資家も存在する。有効性としては大小存在するが、全てのマーケット環境を表現する状態を作成することは現実的に不可能である。そのため単独のエージェントが、全ての期間に対して適切な予測を行うことは非常に難しい課題であると考えている。2 点目についてはノイズトレーダーが画一的な判断基準を持たないことから発生する課題である。短期的な資産価格の形成はノイズトレーダーの行動によって決定する。一方でノイズトレーダーの投資決定の判断基準に画一的な判断基準はなく、資産価格の形成を担うノイズトレーダーが重視するマーケットの状態も異なると考えられる。そのため単一のエージェントで投資判断を行う場合、全ての期間を通して優勢と考えられるノイズトレーダーに適応してしまう可能性が高い。

そのため本稿では、一般的なノイズトレーダーの判断基準になるヒストリカルデータのみを用い、マーケットのマルチクラス分類を行うことで暗にその時に優勢なノイズトレーダーを分類し、そのノイズトレーダーにあった適切な投資判断を行えることを期待して複数のエージェントによる価格の予測を行っていく。

4.2　強化学習について

本稿でのマーケットの予測モデルは現在のタイムステップから次のタイムステップに遷移した際に対象資産の価格が上昇しているか下降しているかのバイナリの予測を行う。一般的にバイナリーの予測を行う際は、分類アルゴリズムを利用し先行研究の多くは SVM や RandomForest、勾配ブースティング、深層学習を利用しているが、今回の研究では強化学

習を利用し、各タイムステップにおいて「購入・売却」の行動を選択するモデルを作成する。教師あり学習ではなく強化学習により投資判断を行う理由は、主に特徴量空間を削減した際の学習効率である。本稿の目的は「価格形成を行うノイズトレーダーが重視するマーケットの状態に対して適切なモデルを選定する」である。そのため、あるマーケットではノイズトレーダーが重視する指標が MACD、ボラティリティ、前回の変化率といった少ない特徴量空間の場合、教師あり学習では学習を効果的に行うことが出来ない。今回用いる強化学習であればエージェントは行動価値関数の更新を行なっていくため、状態の表現が少なくても売買の行動基準を状態から更新できると考えた。

強化学習では環境とその環境内でアクションを行うエージェント（学習を行う主体）において、ある「状態」にいるエージェントが、「行動」（環境への作用等）を選択すると、別の「状態」に移り、結果としてそれが良かった悪かったという情報（報酬）が得られるとする。この時報酬を最大化するような行動の指針である「方策」を見つけるために色々な行動をした後に、その情報（報酬）をもとに、どういう行動をすればよいかの「方策」を自分で学習する。(情報処理推進機構 2019)

強化学習の問題ではエージェントの内部構造を設計することはできるが、環境は原則として所与であるとして考える。この環境は一般的にマルコフ決定過程(Markov Decision Process, MDP)で規定され、マルコフ決定過程で状態、行動、遷移関数、報酬関数の４つの要素から有限マルコフ過程として定式化されることが多くみられる。(加納 2020)

状態有限集合

$$S = \{s_1, s_2, \ldots . s_m\} \tag{4.2}$$

行動有限集合

$$A = \{a_1, a_2, \ldots . a_m\} \tag{4.3}$$

遷移関数：$s \in S$において$a \in A$選択時の$s' \in S$への遷移確率

$$T(s'|s,a) = Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \tag{4.4}$$

報酬関数：$s \in S$において$a \in A$を選択し$s' \in S$遷移時のエージェントに与えられる報酬の期待値

$$r(s,a,s') = \mathbb{E}[r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'] \tag{4.5}$$

離散的なタイムステップtの環境における状態s_tは、エージェントが置かれている状況を表しエージェントはその状態に対する行動a_tを方策$\pi(a \mid s)$に基づき決定する。その行動の結果、状態遷移確率$T(s'|s,a)$に基づき報酬$r(s_t, a_t, s_{t+1})$を受け取り状態s_tから状態s_{t+1}に移行する。最終的には各タイムステップで受け取る報酬$(r_{t+1}, r_{t+2}, \ldots, r_{t+m})$の累計報酬$R$は割引率$\gamma \in [0,1]$を用いて以下のように定式化される。

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^{m-1} r_{t+m} \tag{4.6}$$

方策πの算出方法には最も累積報酬の期待値が高くなる状態sになるよう行動aを方策πに従って決定する状態価値関数$V^{\pi}(s)$を算出する手法と、方策πに従って状態sにおいて最も累積報酬の期待値が高くなるよう行動aを選択する行動価値関数$Q^{\pi}(s,a)$を算出する手法があ

る。逐次的な価値関数の更新のためには、ベルマン方程式を利用する。状態価値関数・行動価値関数のベルマン方程式は以下となる。

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \sum_{s' \in S} T(s'|s,a)\big(r(s,a,s') + \gamma V^{\pi}(s')\big) \tag{4.7}$$

$$Q^{\pi}(s) = \sum_{s' \in S} T(s'|s,a)\left(r(s,a,s') + \sum_{a' \in A(s')} \gamma\pi(a'|s')Q^{\pi}(s',a')\right) \tag{4.8}$$

状態遷移確率が予め陽に分かれば、複雑な問題に対してもベルマン方程式を立てることが可能であるが、状態遷移確率は未知であるため、価値関数を求める必要がある。その一つの Q-Learning は累積報酬が最大化させる最適方策π^*の下での行動価値関数$Q^*(s,a)$を価値反復法によって算出する。

$$Q^*(s,a) = \mathbb{E}_{\pi*}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s, a_t = a\right] \tag{4.9}$$

これによりベルマン最適方程式を満たす関数は以下となる。

$$Q^*(s,a) = \sum_{s' \in S} T(s'|s,a)\left(r(s,a,s') + \gamma \max_{a' \in A} Q^*(s',a')\right) \tag{4.10}$$

Q-Learning では上記のベルマン最適方程式を満たすよう行動価値関数$Q(s,a)$を逐次的に次のように更新する。

$$Q(S_t,A_t) \leftarrow (1-\alpha)Q(S_t,A_t) + \alpha\left(R_{t+1} + \gamma \max_{\alpha' \in A(S_{t+1})} Q(S_{t+1},\alpha')\right) \tag{4.11}$$

αは学習率である。これにより状態遷移確率が未知であっても状態遷移確率を式に含めないため計算することが可能となる。また、常にQが最大となる行動を学習し続けた場合は最適な値が初期値に大きく依存し局所的な解に陥る場合がある。それを回避するために ε-greedy 法によって確率 ε でQ値が最大ではない行動をランダムで選択することで、最適な行動価値関数を探索する。(牧野 2016)

ただ、この場合の反復アルゴリズムは一般的に i→∞として最適行動価値関数Q^* に収束するがこれは各状態に対して個別に推定され、マルコフ決定過程における状態と行動の組みを全て保持する必要がある。そのため関数近似を用いて行動価値関数$Q(s,a,\theta) \approx Q^*(s,a)$の推定を行う必要がある。Mnih et al. (2015)は行動価値関数の推定に深層学習を適用した Deep Q Netework (DQN)を提唱した。DQN は重みをもつニューラルネットワークの関数近似を Q-network と呼び。各iの反復における次の損失関数$L_i(\theta_i)$を以下のように最小化する。

$$L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot)}\left[\big(y_i - Q(s,a;\theta_i)\big)^2\right] \tag{4.12}$$

$\rho(s,a)$は状態sと行動aに関する確率分布で、ここでは行動分布と呼ばれる。y_iは以下で定義される。

$$y_i = \mathbb{E}_{s\prime\sim\varepsilon}\left[r + \gamma \max_{a\prime} Q(s',a';Q_{i-1}) \mid s,a\right] \tag{4.13}$$

損失関数L_iを最適化する場合、前ステップのθ_{i-1}からパラメータが固定される。損失関数の重みを微分し以下の勾配を得る。

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(\cdot);s\prime\sim\varepsilon}\left[\left(r + \gamma \max_{a\prime} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\ \nabla_{\theta_i} Q(s,a;\theta_i)\right)\right] \tag{4.14}$$

また期待値で計算するのではなく、確率的勾配法でも最適化することもできる。

一般に DQN では誤差L_iを最小化して学習していく。DQN の問題点としてはQ^*の値が過大評価される傾向があることが指摘されている。Hasselt et al. (2016) は$Q(s,a)$にランダムノイズが乗ると、過大評価が起こりやすいとし解消するために DDQN (Double DQN) を提案した。DDQN では Q-network として Q-network と Target Q-network (Q_t)の二つを用意し、行動決定を Q-network から、$Q(s,a)$の評価をQ_tから行い、Q^*の過大評価を低減する。

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \eta\left(R_{t+1} + \gamma Q_t(s_{t+1},a_t) - Q(s_t,a_t)\right) \tag{4.15}$$

DDQN ではこの更新をより安定させる方法として更新式に以下を利用する。

$$a_m = \arg\max_a Q(s_{t+1},a)$$
$$Q_m(s_t,a_t) \leftarrow Q_m(s_t,a_t) + \eta\left(R_{t+1} + \gamma Q_t(s_{t+1},a_m) - Q_m(s_t,a_t)\right) \tag{4.16}$$

次の状態s_{t+1}でのQ値が最大となる行動a_mを Q-network から求め、その行動a_mでのQ値は target Q-network から求める。また、モデルの作成にあたっては学習の安定化を図るため、経験再生(Experience Replay)・報酬 Cliping を入れ、損失関数は Huber 関数、最適化手法は Adam を利用した。

経験再生

時間ステップの近いデータ間で強い相関を持つ場合、価値関数を近似すると直近の入力に引きずられたパラメータの修正が行われる。この場合、収束性が悪く過去の行動価値関数の推定が悪化し、収束が遅くなる。この問題を解消するために Lin (1992)は経験を状態xにおいて行動aを行うと新しい状態yと報酬rが得られると定義し、これを再利用することで学習エージェントは以前に同様の経験をしたこととしネットワークが通常よりも収束が早くなる、経験再生を提言した。なお、経験再生が有効である条件を環境が急激に変化しないことが重要であるとしている。本研究での経験再生は一様乱数によってランダムに経験を選び出し繰り返し利用する。

報酬クリッピング

金融商品のマーケットに対して、得られた利益rを連続値として扱う場合、マーケットのボラティリティが高まるタイミングでは、急激に報酬が増減するタイミングが頻繁に訪れる。報酬を連続値として扱うと行動価値関数Q値の急激な増大が発生し、学習が安定しない。そのため報酬クリッピングを導入した。報酬クリッピングは Mnih et al. (2015)による DQN を利用する際の手法で「正の報酬 : 1」「負の報酬 : -1」「報酬なし : 0」とする手法である。本研究では、収益率rの大きさを考慮せず「$r > 0: 1$」「$r < 0: -1$」「$r = 0: 0$」として報酬を設定した。報酬クリッピングを導入した場合、利益率の大小を区別できなくなりマーケットクラッシュによる大きな損失や反対に大きな利益を区別できなくなる一方で利益率を考えないことで、問題設定を単純化することが出来る。モデルはマーケットが次の時間ステップ時点で上昇していると判断するときは購入、下落していると判断するときは売却を行う。

4.3　強化学習によるマーケット予測

本稿での強化学習により行動価値関数を更新したエージェントはタイムステップ毎に状態として与えられた各特徴量から行動価値関数に基づき 2 通りの行動 $a_t \in \{Long, Short\}$ を選択する。

環境の始点と終点であるエピソードは、1 日単位を 1 エピソードとした。暗号資産取引所のオープン時間は 24 時間 365 日であり、取引所のメンテナンスでクローズしているタイミング以外は取引が可能であるため、1 エピソードに含まれるタイムステップは 24 タイムステップとなる。1 エピソード内に取引所システムのメンテナンス等によりクローズしているタイミングがある場合は、該当エピソードを学習・検証・テストデータから除外する。これはメンテナンス中に他取引所で行われている取引によって対象取引所の価格がメンテナンス終了時に大きく変化しやすいためである。エピソードに利用する学習・検証・テスト期間は以下の通りである。

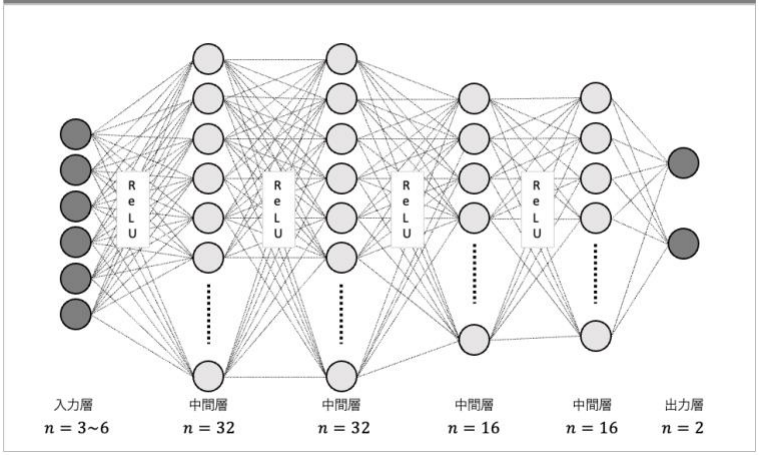
学習期間:2018 年 1 月 1 日～2021 年 8 月 31 日
検証期間:2021 年 9 月 1 日～2022 年 1 月 14 日
テスト期間:2022 年 1 月 15 日～2022 年 3 月 31 日

テスト期間は信用取引が行える先物取引市場でのバックテストを行う。
本稿ではノイズトレーダーによるトレードは各タイミングにおいて各々の判断に基づき投資が行われ、各タイミングで優勢なノイズトレーダーの判断基準は異なると仮定し、全ての特徴量を用いて状態を構築し同時に価値反復法を行うのではなく、各特徴量を組み合わせた状態をエージェントに与え行動価値関数の更新を行なった。

学習期間においては経験学習を採用し期間内の 1 エピソードをランダムサンプリングし、合計 80,000 エピソード、各状態の組み合わせにおいて 2 回の学習機会を設けた。モデルは 20,000 エピソード毎に保存を行い、検証期間において正解率の高いモデルをテスト期間でバックテストする。

まず、強化学習によるマーケット学習の可能性を検討するにあたり、単一モデルでテストデータに対して行動を決定した場合の正解率を確認する。本稿で利用した DDQN のネットワーク構成は図 4.2 の通りである。入力層は各特徴量の組み合わせで 3〜6 の入力となり、中間層 4 層、出力層は 2 値の出力となる。このモデルによる正解率上位 10 モデルは表 4.1 の通りである。

（図4.2）強化学習モデルニューラルネットワーク構成図



(表4.1)強化学習モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
		○	○	○		○	2	v0	0.551	0.563
	○	○		○	○		2	v1	0.549	0.524
○	○		○	○	○		8	v0	0.549	0.545
○	○		○	○	○		4	v0	0.549	0.553
○	○		○	○			4	v0	0.548	0.559
○		○	○	○			2	v1	0.548	0.550
		○	○	○		○	2	v1	0.547	0.548
○	○	○	○		○		4	v0	0.547	0.545
○	○	○	○				4	v1	0.547	0.550
○	○	○	○				2	v1	0.546	0.557

テスト期間におけるタイムステップは 1824 ステップであり、ランダムウォークに対してテスト期間のモデルが有意であるか母比率に対する仮説検定を行う。帰無仮説・対立仮説は以下の通りである。

帰無仮説

$$H_0 : p = 0.5$$

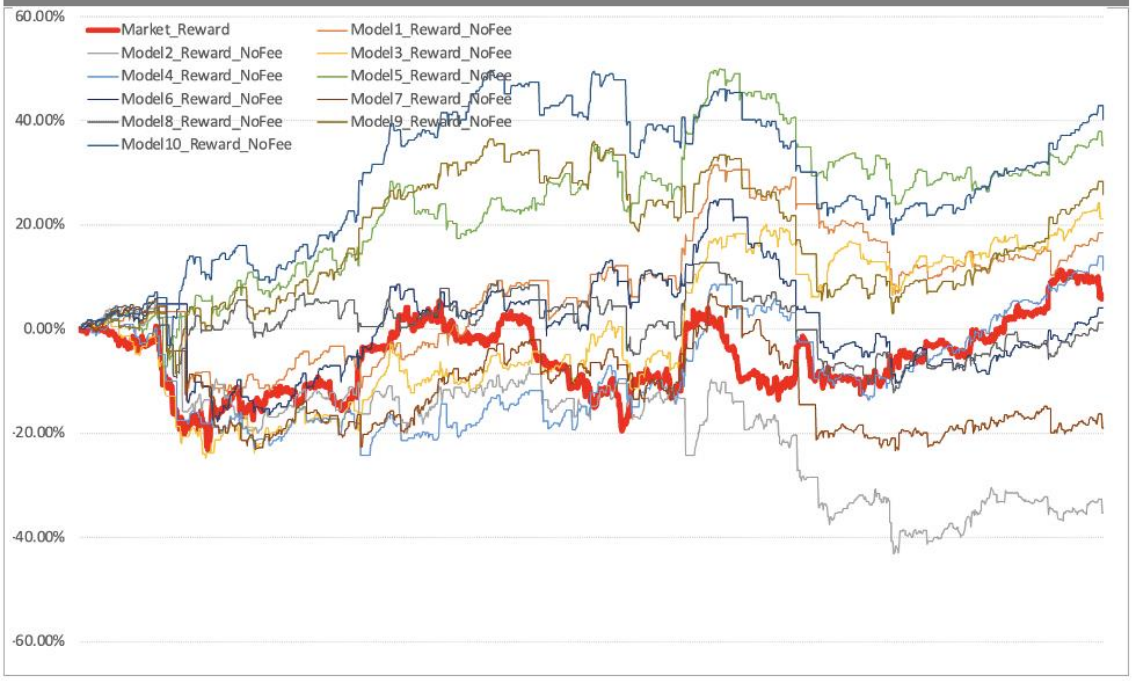
対立仮説

$$H_1 : p > 0.5$$

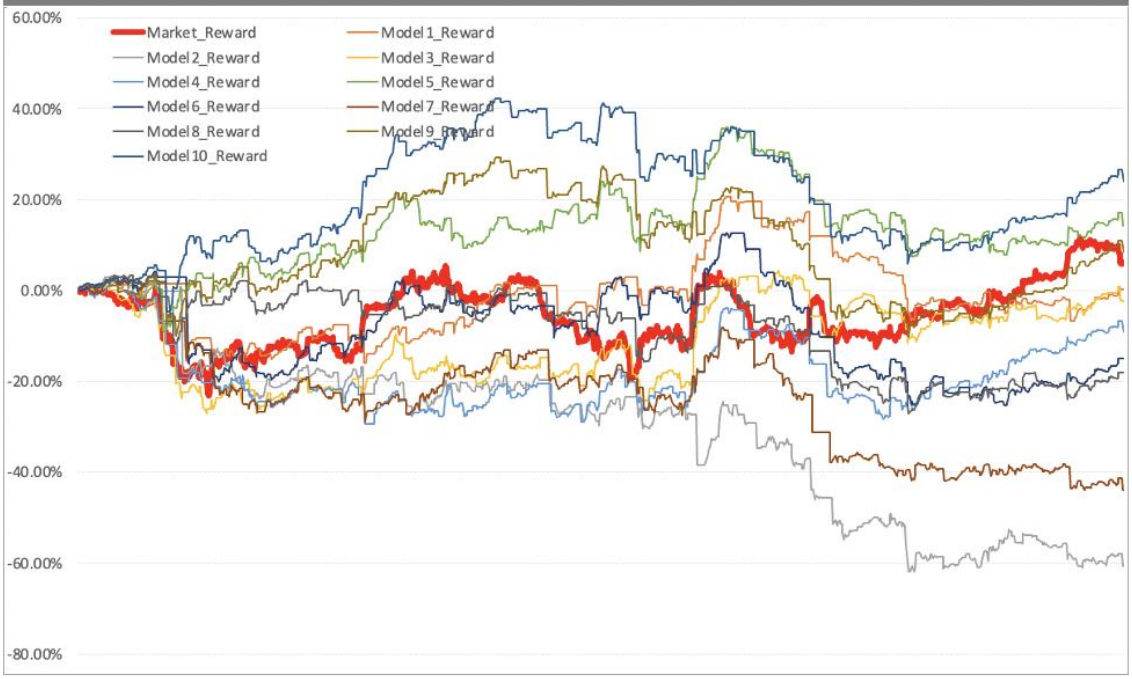
上位 10 モデルのうちテスト期間における検定統計量は$2.647 < z < 5.552$となり、正解率の最も低いモデルにおいても帰無仮説は有意水準 1%で棄却される。上位 10 モデルを運用した場合のテスト期間における手数料除く収益率を図 4.3 に示す。図 4.4 には手数料を含む

モデルの運用シミュレーションを示す。モデルは購入・売却の判断時には常に 1 BTC 単位での取引を行い、次のステップで前のステップと同等の判断を行った際は反対売買をせずに、手数料は反対売買を行ったタイミングで計上する。

（図4.3）バイアンドホールド/モデル別収益率比較(手数料除く)



（図4.4）バイアンドホールド/モデル別収益率比較(手数料含む)



単一モデルでの収益率で手数料を考慮しない場合は、検証期間における正解率上位 10 モデルのうち 6 モデルがバイアンドホールドと比較して高い収益率となっている。手数料を考慮した場合はバイアンドホールドを上回ったパフォーマンスを上げているモデル数は 3 となった、タイムステップの浅い箇所では、8 モデルがバイアンドホールドと比較して収益率が高く推移している。一方 3 章で行ったシミュレーションの結果と収益率の乖離が見られる。テスト期間における収益率の信頼区間 99%点は−4%~106%の範囲となるが、正解率 55%のモデルによる収益率は、低いもので−15%程度の損失が確認できる。この原因について調査したところ、大きく変動するマーケット環境での正解率の低さが見られた。テスト期間において 1%以上の変動幅となるタイムステップは 211 ステップあったが、表 4.2 に示すように該当モデルのその区間における正解率は 43%程度であった。同タイムステップ数での正解率 55%の母比率の 99%信頼区間は下限で 46.16%となる。作成したモデルは大きな価格変動時に対する適応ができておらず、その結果収益率がシミュレーションと乖離したと考えられる。

(表4.2)高変動時正解率

Ver	Accuracy Test	1%以上変動 正解率
Model_1	0.563	0.436
Model_2	0.524	0.436
Model_3	0.545	0.469
Model_4	0.553	0.436
Model_5	0.559	0.450
Model_6	0.55	0.431
Model_7	0.548	0.436
Model_8	0.545	0.422
Model_9	0.55	0.479
Model_10	0.557	0.483

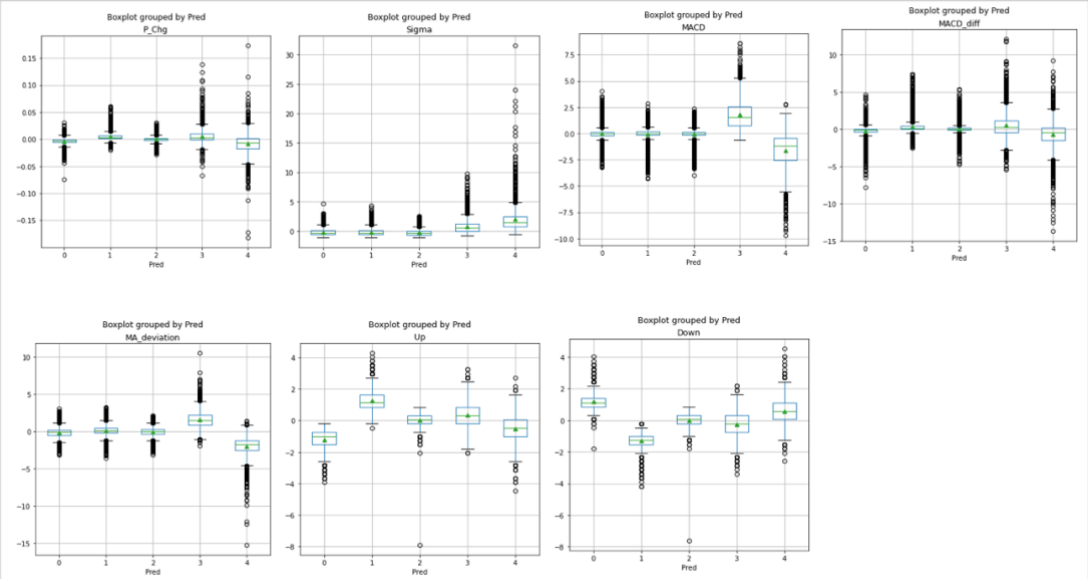
4.4　クラスタリングモデルについて

前節で作成したモデルをマーケット環境に応じて適切に配置し推論を行うために、マーケット環境を分類するクラスタリングモデルを作成する。分類するマーケット環境は、強化学習に利用した特徴量である「直前タイムステップの価格変化率」「直前タイムステップの分足標準偏差」「直前タイムステップの 1 分毎価格上昇・下落回数」「SMA・移動平均乖離率」「MACD」「MACD の差分」である。クラスタリングは教師なし学習の一つであり、各グループのデータ点が規則性を示すようにグループ化する。 (Jianliang et al. 2009)

Fang et al. (2022) の調査によると、暗号資産の先行研究におけるクラスタリングで最も使用されているアルゴリズムは K-Means である。K-Means はN次元のM個のポイントをK

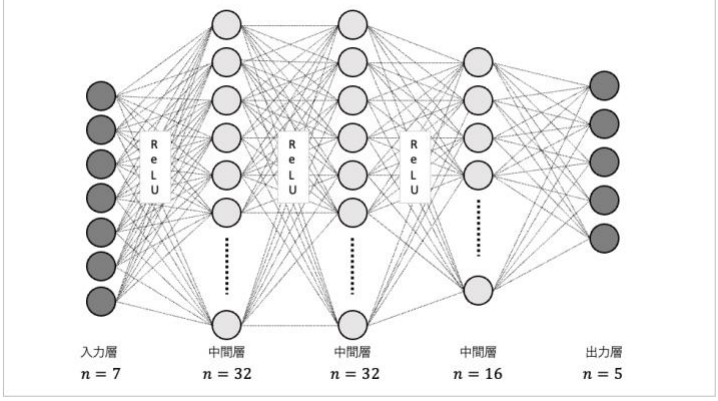
個のクラスタに分割して、クラスタ内の平方和を最小にする。$K = 2$の場合を除き、解がすべての分割に対して最小の平方和を持つことを要求するのは現実的ではないため、クラスタから別のクラスタへの点の移動がクラスタ内の平方和を減少させないような解である局所最適解を求める。これは類似性のグループ化、非線形予測、多変量分布の近似、独立性のノンパラメトリック検定に応用が可能である。 (MacQueen 1967, Hartigan et al.1979) K-Means アルゴリズムは初期値に任意のKを決定する必要があり、本研究でのクラスタ数$K = 5$とした。クラスタリングした結果の特徴量を四分位でまとめ、クラス毎の各特徴量の分布を図 4.5 に示す。

（図4.5）クラスタ別特徴量箱ひげ図



クラスタ 3,4 の分布に大きな違いが見られ、各マーケットの状態を分類できている可能性が高いと考えられる。K-Means でマーケット状態を 5 クラスに分類し、今後のデータで利用するために K-Means の分類結果を分類問題と捉え、マルチクラス分類のモデルを作成した。マーケット分類予測モデルは DNN を利用し、図 4.6 にモデル構成図を記載する。

（図4.6）マーケット分類予測モデル構成図



5　マルチエージェントモデルについて

非合理的なノイズトレーダーの投資行動は、ある時刻のマーケットの状態に対して全てのトレーダーが画一的な見方をすることはない。なぜなら、合理的な投資家は「現時点の価格は全ての情報を加味した上での適切な価格」という判断になり自身のポジションから発生した損益に影響を受け投資行動が変化することはないが、非合理的な投資家は自身のポジション状況やその損益によって、マーケットの変化に対して投資行動が変化する。非合理的な投資家の行動は行動経済学のプロスペクト理論に基づくと、下落相場の際にポジションを保有している非合理的な投資家は認知バイアスや損失回避性から、マーケットに対する不安感は大きくなり悲観的な感情を齎し適切な投資行動を選択する可能性ができなくなる。一方で既に利益を得ておりポジションを保有していない投資家にとっては投資判断が楽観的になり、ロングポジションを新たに入れるタイミングを模索する場面となる。

今回のクラス分類後に強化学習モデルにより適切なモデルを選択する方法論は、マーケットの状態に着目し各マーケットに存在するノイズトレーダーの行動に対してその状態に最も適切なモデルを選択することで、マーケットセンチメントをモデルに反映し各タイムステップで優勢なノイズトレーダーの行動を反映して予測を行えることが期待される。

本モデルの検証のためにまず検証期間で各タイムステップに対してクラス分類を行い各モデルの各クラスタにおける正解率を確認する。クラス分類後の正解率上位モデルの検証期間とテスト期間での正解率を表 5.1-5 に示し、各クラスタの精度上位 10 銘柄が利用した特徴量の集計結果を表 5.6 に示す。クラスタにおける精度のばらつきはクラス毎のデータセット数により変化が大きくなっているが、検証期間において上位の精度を出しているモデルの利用している特徴量をクラスタ間で比較すると、利用している特徴量に差が現れている。移動平均の乖離率については各クラスタ間で使われている頻度に大きな差はないが、MACD や前ステップの価格変化率などにクラスタ間の差が見られる。

(表5.1)マーケットクラスタ0 モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
○	○		○	○	○		8	v0	0.606	0.538
○	○		○	○	○		6	v1	0.604	0.603
○	○		○	○			4	v0	0.604	0.603
○	○		○	○	○		6	v0	0.601	0.558
○	○		○	○			6	v0	0.601	0.598
○		○	○		○		6	v1	0.599	0.548
○	○		○	○			8	v0	0.599	0.613
○	○			○		○	4	v0	0.597	0.583
○		○	○	○			2	v1	0.597	0.588
○	○			○	○		6	v0	0.597	0.608

(表5.2)マーケットクラスタ1 モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
○	○	○	○			○	6	v0	0.551	0.542
	○	○		○	○		2	v1	0.549	0.508
	○	○		○	○		6	v1	0.545	0.539
○	○	○	○			○	8	v0	0.543	0.533
○	○	○		○	○		4	v0	0.543	0.529
		○	○	○	○	○	8	v0	0.542	0.527
○	○	○	○		○		8	v1	0.541	0.521
		○	○	○	○		4	v1	0.541	0.518
		○	○	○		○	4	v1	0.54	0.544
○		○	○	○			2	v1	0.54	0.512

(表5.3)マーケットクラスタ2 モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
○	○		○			○	4	v0	0.598	0.569
○			○	○		○	6	v0	0.593	0.554
○	○		○	○	○		8	v1	0.585	0.538
○	○			○		○	8	v0	0.581	0.554
○		○		○	○	○	2	v0	0.581	0.531
○			○	○		○	8	v0	0.581	0.531
○	○		○			○	6	v0	0.581	0.554
○				○	○	○	6	v0	0.577	0.577
○	○		○		○		4	v1	0.577	0.554
○	○			○		○	4	v0	0.577	0.569

(表5.4)マーケットクラスタ3 モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
○	○	○			○	○	8	v0	0.565	0.578
○	○	○	○	○			6	v1	0.564	0.57
○	○	○			○		4	v1	0.56	0.57
		○	○	○			4	v0	0.56	0.594
○	○	○			○		6	v1	0.558	0.565
○	○		○	○			4	v0	0.557	0.567
○	○	○		○			4	v1	0.556	0.554
	○		○	○	○		2	v0	0.556	0.583
○	○	○	○	○			8	v1	0.556	0.573
○	○	○			○	○	6	v0	0.554	0.573

(表5.5)マーケットクラスタ4 モデル推論結果

Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down	Episode	Ver	Accuracy Valid	Accuracy Test
		○	○	○	○	○	2	v1	0.572	0.574
		○	○	○		○	4	v0	0.572	0.548
○		○	○		○		6	v1	0.571	0.569
○		○	○	○	○		2	v0	0.571	0.514
○		○	○	○	○		4	v1	0.569	0.567
		○	○	○	○	○	4	v1	0.568	0.559
○		○	○		○		4	v1	0.566	0.564
	○	○		○	○		8	v1	0.564	0.546
○		○	○		○		8	v1	0.564	0.567
		○	○	○	○	○	6	v1	0.564	0.54

(表5.6)精度上位10モデルクラスタ別利用特徴量

	Previous Change Rate	Previous Sigma	MACD	MACD Diff	MA Deviation	Up	Down
Cluster_0	10	8	2	8	9	5	1
Cluster_1	5	6	10	7	7	6	4
Cluster_2	10	6	1	6	7	4	8
Cluster_3	8	9	8	5	6	5	2
Cluster_4	5	1	10	9	7	9	4

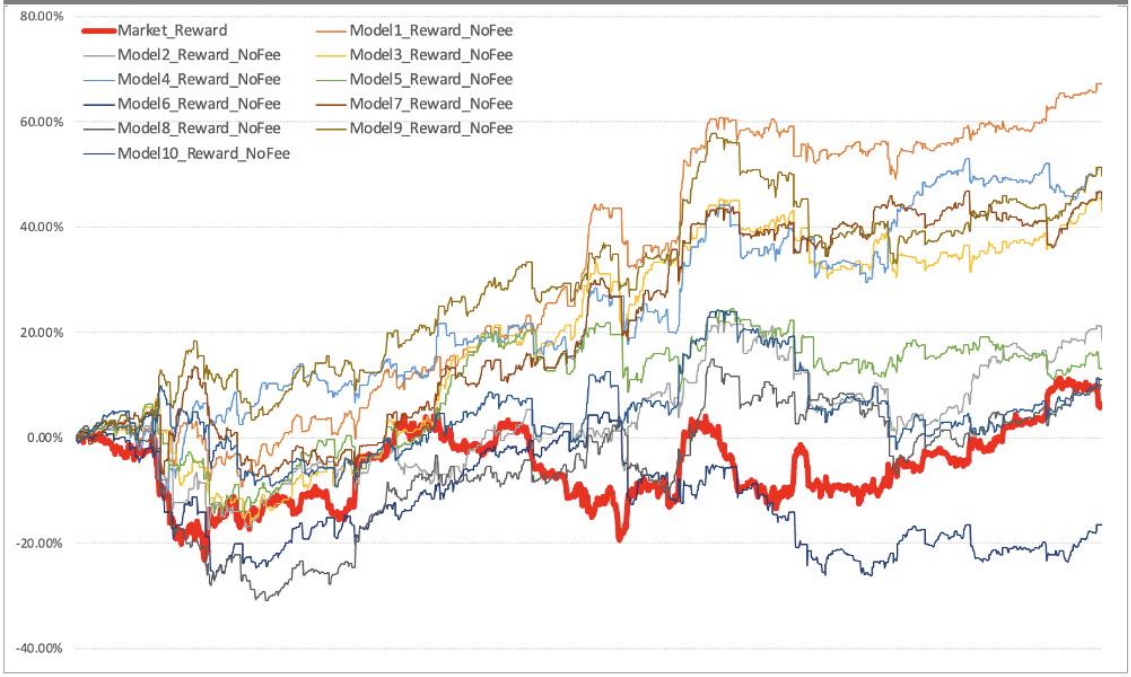
このことから各クラスタにおいて正解率が高いエージェントが投資判断を行うために必要としている状態が異なることがわかる。次に各クラスタに対して適切なモデルを組み合わせ、正解率・収益率の精度向上が見られるか実証する。クラスタリング後に選択するモデルの組み合わせは、各クラスタの検証期間における正解率の順位順に組み合わせを行う。テスト期間においては、作成したクラスタリング結果を予測するモデルで該当するクラスタを予測し、その予測結果に基づき各モデルの組み合わせが適切なエージェントを使いマーケットの予測を行った。テスト期間における正解率と単独モデルで課題のあった、変動幅の大きいタイムステップにおける正解率を表 5.7 に示す。

手数料含めないケースの収益率を図 5.1 に、手数料を含めたケースを図 5.2 に示す。

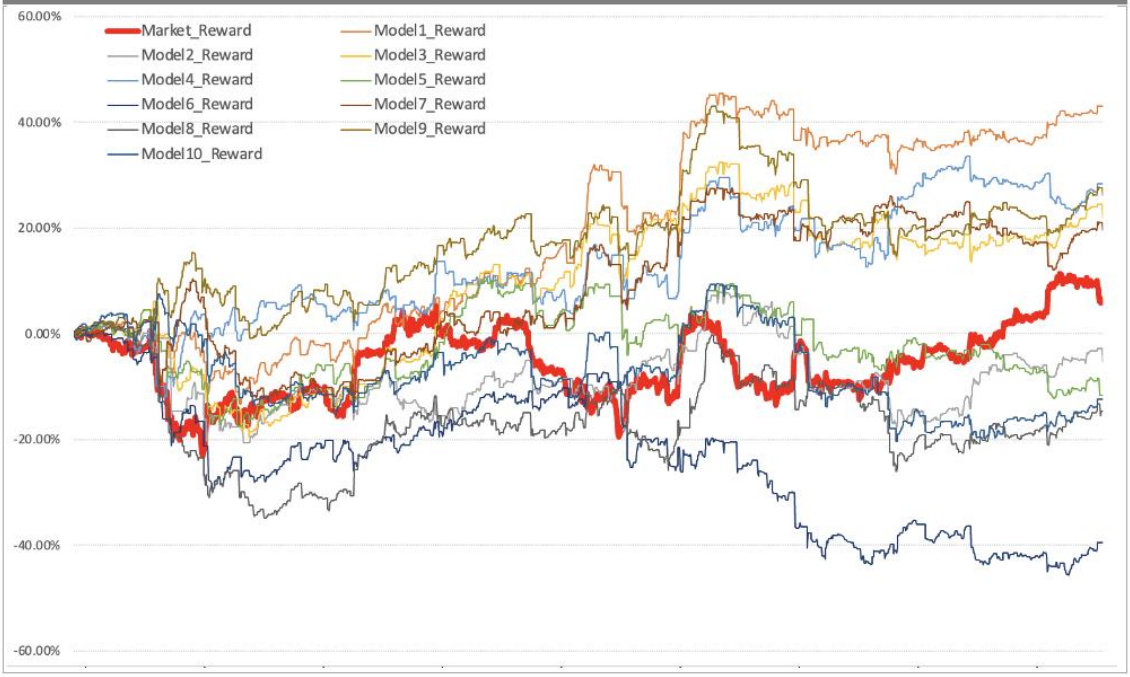
(表5.7)モデル組合せ別正解率

組合順位	Accuracy Test	1%以上変動 正解率
Model_1	0.558	0.502
Model_2	0.542	0.474
Model_3	0.558	0.479
Model_4	0.550	0.502
Model_5	0.548	0.441
Model_6	0.539	0.389
Model_7	0.549	0.488
Model_8	0.549	0.431
Model_9	0.560	0.464
Model_10	0.543	0.431

(図5.1)バイアンドホールド/モデル別収益率比較(手数料除く)



(図5.2)バイアンドホールド/モデル別収益率比較(手数料含む)



正解率の観点ではマーケット環境に応じた組み合わせによって精度の向上は見られなかったが、単独モデルで課題のあった変動幅が大きいタイムステップでの精度悪化については改善が見られた。収益率については、単独のエージェントでは検証期間において最も精度の高かったエージェントがテスト期間において最も収益を上げるエージェントとはならなかったが、今回の手法では検証期間において最も精度の高い組み合わせのエージェントが最も収益率が高いモデルという結果になった。また、表 5.8 に手数料を含めた単独モデルと今回の手法を利用した複合モデルの検証期間正解率上位 10 モデルの収益率比較を示す。ここからもわかるように、単独モデルと比較して収益率について安定化が見られ、最もパフォーマンスの悪かったモデルの収益率については 20%近くの向上が見られ、平均収益率はプラスとなった。また、手数料を含めてバイアンドホールド戦略を上回るパフォーマンスが出たモデル数も、単独モデルでは 3 モデルだったのに対して複合モデルでは 5 モデルがバイアンドホールド戦略を上回った。

(表5.8)単独モデル･複合モデル収益率比較

	最大 収益率	最小 収益率	平均 収益率	収益率 標準偏差
単独モデル	23.92%	-60.71%	-10.25%	0.24
複合モデル	43%	-39.54%	5.54%	0.24

6　結論・今後の課題

Fama (1965)は「テクニカル分析は株式市場の投資家にとって何の価値もない」とし過去のヒストリカルデータから将来の価格を予想することを否定したが、過去の文献などからも分かるようにアカデミック・ビジネス実務の両観点から、金融資産の価格予測は研究され続けている。本研究でも示唆を得たように、過去のヒストリカルデータのみから作成したモデルを用いてランダムウォークを上回る正解率を導出するモデルの作成は可能である。また強化学習のフレームワーク内でマーケットの予測を行うという試みは、先行研究自体が少ないものの外部データに依存せずヒストリカルデータのみの環境においてもランダムウォークを上回る正解率のエージェントを作成することが可能であることが実証された。また実務の観点では正解率ではなく収益率が重要視されるが、作成したモデルを用いてベンチマークであるバイアンドホールド戦略をアウトパフォームした。単独の強化学習モデルで全てのマーケットを予測するという切り口では、正解率ではランダムウォークを超えることが出来たが、収益率の観点で見ると手数料を含まないケースでは 10 モデル中 6 モデル、手数料を含むケースでは 10 モデル中 3 モデルにとどまり収益率に課題が残った。一方で、単独のモデルでのマーケット予測に対して、マーケット環境をクラスタリングし暗にその予測ステップにおける重要な状態を考慮し、予測ステップで適切なモデルを複合的に組み合わせる複合型の運用モデルでは、単独モデルから正解率の観点では精度の向上は見られなかったが、収益率の観点では手数料を含まないケースでは 10 モデル中 9 モデル、手数料を含むケースでは 10 モデル中 5 モデルがバイアンドホールド戦略を上回るパフォーマンスとなった。

本稿での実証により強化学習を用いたマーケットの価格予測の可能性と強化学習を利用した際にマーケット環境に対する適切なエージェントの選定による複合モデルの効果がある可能性を示唆した点が本稿の貢献である。今後の研究課題は以下の 4 点が挙げられる。

1.モデルの再学習のタイミングについて適切なタイミングの検証が必要である。今回の収益率推移で多くのモデルがテスト期間の 200 時間程度の浅いタイムステップではバイアンドホールド戦略の収益率を上回っていたが、時間経過とともに収益率の悪化が見られた。実運用に向けたモデル再学習のタイミングを検討する必要がある。

2.状態空間特徴量の選定が課題である。今回利用したテクニカル指標はマーケット参加者にとって利用頻度が多いMACDと移動平均乖離率であり他はマーケットの特徴を表す特徴であった。マーケット判断に活用されるテクニカル指標は多数あり、ノイズトレーダーが重視するテクニカル指標を組み込むことで精度向上が期待できると考える。

3.複合モデル利用モデルの選択基準について検証が必要である。本稿では簡易的に検証期間における単独モデルの正解率降順の組み合わせで複合モデルを構築した。一方で複合モデルの選択を正解率順に絞らず収益率など考慮した場合、テスト期間の収益率が向上するケ

ースが見られた。今後モデル選択基準を他手法で明示的に選ぶことができればパフォーマンスの向上が期待できる。

4.複合モデルにした際に改善は見られたが単独モデルで見られたボラティリティが増加しているマーケットの精度悪化が挙げられる。本稿で採用した報酬クリッピングによるリターンの調整で収益率の悪化が見られた可能性が考えられ、適切な報酬設定など今後の研究課題である。

謝辞

本論文の作成にあたり、終始適切な助言を賜り、ご指導いただきました主査の笛田薫教授に感謝します。金融データに対する適切なアプローチ方法・分析方法・考察など多くの面でご助言いただき誠にありがとうございました。副査の田中琢磨教授に感謝いたします。内容の構成や表現方法、私では気づかなかった問題点など適切なアドバイスを多くいただき、今後自分で研究を進める際の課題点など多くのことに気づくことができました。副査の姫野哲人准教授に感謝いたします。修士論文を書く際の重要な事項や内容の整理など多くのアドバイスをとても丁寧にいただき、アドバイスを頂かなければ適切な論文の作成が出来ませんでした。最後に社会人派遣学生として派遣していただいた SMFG・SMBC 日興証券の皆様に感謝いたします。

参考文献

[1] Barnwal, A. et al. (2019) "Stacking with neural network for cryptocurrency investment." arXiv preprint arXiv: 1902.07855.

[2] Bradford, J. et al. (1990) "Noise Trader Risk in Financial Markets." Journal of Political Economy, Vol.98, No.4, pp.703-738.

[3] Fang, F. et al. (2022) "Cryptocurrency trading: a comprehensive survey." Financial Innovation, Vol.8, No.13.

[4] Fama, E. (1965) "The Behavior of Stock-Market Prices." Journal of Business, Vol.38, No.1, pp.34-105.

[5] Grossman, S. J. et al. (1980) "On the Impossibility of Informationally Efficient Markets." American Economic Review, Vol.70, No.3, pp.393-408.

[6] Hartigan, J. et al. (1979) "A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol.28, No.1, pp.100-108.

[7] Hasselt, H. et al. (2016) "Deep reinforcement learning with double Q-learning", in Thirtieth AAAI Conference on Artificial Intelligence, Vol.30, No.1, pp.2094-2100.

[8] Ito, A. (1999) "Profits on technical trading rules and time-varying expected returns: evidence from Pacific-Basin equity markets." Pacific-Basin Finance Journal, Vol.7, No.3, pp.283-330.

[9] Ji, S. et al. (2019) "A comparative study of bitcoin price prediction using deep learning." Mathematics, Vol.7, No.10, 898.

[10] Jianliang, M. et al. (2009) "The application on intrusion detection based on k-means cluster algorithm." 2009 International Forum on Information Technology and Applications, Vol.1, pp.150-152.

[11] 情報処理推進機構編　(2019)「AI 白書 2019」　角川アスキー総合研究所

[12] 加納　由希夫 (2020)「好奇心に基づく内部報酬を用いた強化学習によるローグライクゲームの学習」, 東京大学大学院情報理工学系研究科修士論文 https://repository.dl.itc.u-tokyo.ac.jp/records/54236

[13] Kim, J. (2009) "Automatic variance ratio test under conditional heteroskedasticity." Finance Research Letters, Vol.6, No.3, pp.179-185.

[14] Ljung, G. (1978) "On a measure of lack of fit in time series models." Biometrika, Vol.65, No.2, pp.297-303.

[15] Lin, L. (1992) "Self-improving reactive agents based on reinforcement learning, planning and teaching", Machine Learning, Vol.8, pp.293-321.

[16] Madan, I. et al. (2015) "Automated bitcoin trading via machine learning algorithms."
https://cs229.stanford.edu/proj2014/Isaac%20Madan,%20Shaurya%20Saluja,%20Aojia%20Zhao,Automated%20Bitcoin%20Trading%20via%20Machine%20Learning%20Algorithms.pdf

[17] 牧野 貴樹 他 (2016) 「これからの強化学習」 森北出版

[18] McNally, S. et al. (2018) "Predicting the price of bitcoin using machine learning." 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp.339-343.

[19] MacQUEEN, J. (1967) "Some methods for classification and analysis of multivariate observations." In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp.281-297.

[20] Menkhoff, L. (2010) "The use of technical analysis by fund manager: International evidence." Journal of Banking & Finance, Vol.34, No.11, pp.2573-2586.

[21] Mnih, V. et al. (2015) "Human-level control through deep reinforcement learning", Nature, Vol.518, No.7540, pp.529-533.

[22] Nakano, M. et al. (2018) "Bitcoin technical trading with artificial neural network." Physica A: Statistical Mechanics and its Applications, Vol.510, No.15, pp. 587-609.

[23] Nakamoto, S. (2008) "Bitcoin: A Peer-to-Peer Electronic Cash System."
https://bitcoin.org/bitcoin.pdf

[24] Park, C. et al. (2007) "What do we know about the profitability of technical analysis? " Journal of Economic Surveys, Vol.21, No.4, pp.786-826.

[25] Sebastião, H. et al. (2021) "Forecasting and Trading Cryptocurrencies with Machine Learning under Changing Market Conditions." Financial Innovation, Vol.7, No.3.

[26] Shahbazi, Z. et al. (2021) "Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning." in IEEE Access, Vol.9, pp. 162651-162659.

[27] Urquhart, A. (2016) "The inefficiency of bitcoin." Economics Letters, Vol.148, pp.80-82.

[28] Wald, A. et al. (1940) "On a Test Whether Two Samples are from the Same Population." Annals of Mathematical Statistics, Vol.11, No.2, pp.147-162.

[29] Yermack, D. (2015) "Is bitcoin a real currency? An economical appraisal." Handbook of Digital Currency. Academic Press, pp.31-43.

付録

強化学習ネットワーク構成

```
class Network:
    def __init__(self,input_size,output_size):
        self.lr = 0.0001
        self.model = nn.Sequential()
        self.model.add_module('fc1',nn.Linear(input_size,32))
        self.model.add_module('relu1',nn.ReLU())
        self.model.add_module('fc2',nn.Linear(32,32))
        self.model.add_module('relu2',nn.ReLU())
        self.model.add_module('fc3',nn.Linear(32,16))
        self.model.add_module('relu3',nn.ReLU())
        self.model.add_module('fc4',nn.Linear(16,16))
        self.model.add_module('relu4',nn.ReLU())
        self.model.add_module('fc5',nn.Linear(16,output_size))
        self.optimizer = optim.Adam(self.model.parameters(),lr = self.lr)
```

DDQN 構成

```
class DDQN_network:
    def __init__(self):
        self.x ='DDQN'
    def q_update(self):
        if len(brain.Memory.memory) < batch_size:return
        self.tmp_memory = np.array(random.sample(brain.Memory.memory,batch_size))
        self.state_b,self.action_b,self.next_state_b,self.reward_b =
                                                brain.Memory.replay_memory(self.tmp_memory)
        brain.main_q_network.model.eval()
        brain.target_q_network.model.eval()
        Q = brain.main_q_network.model(self.state_b).gather(1,self.action_b)
        Q_main_index = brain.main_q_network.model(self.next_state_b).max(1)[1].view(-1,1)
        Q_target_value =brain.target_q_network.model(self.next_state_b)
                                                    .gather(1,Q_main_index).detach().squeeze()
        Q_expe = self.reward_b + 0.99 * Q_target_value
        brain.main_q_network.model.train()
        loss = F.smooth_l1_loss(Q,Q_expe.unsqueeze(1))
        brain.main_q_network.optimizer.zero_grad()
        loss.backward()
        brain.main_q_network.optimizer.step()

    def target_q_update(self):
        brain.target_q_network.model.load_state_dict(brain.main_q_network.model.state_dict())
```

ExperienceReplay 構成

```
class Memory:
    def __init__(self,memory_size,state_size):
        self.memory_size = memory_size
        self.memory=[]
        self.index =0
        self.s = state_size
        self.ns = state_size +1
        self.r = self.ns + self.s
    def save_memory(self,state,action,next_state,reward):
        tmp =[state,[action],next_state,[reward]]
        tmp = list(itertools.chain.from_iterable(tmp))
        if len(self.memory) < self.memory_size:
            self.memory.append(None)
        self.memory_number = self.index % self.memory_size
        self.memory[self.memory_number] = tmp
        self.index +=1
    def replay_memory(self,tmp_memory):
        self.state = brain.tensor_to(tmp_memory[:,:self.s])
        self.action=brain.tensor_to(tmp_memory[:,self.s])
        self.action=self.action.to(torch.long)
        self.action=torch.reshape(self.action,(batch_size,1))
        self.next_state= brain.tensor_to(tmp_memory[:,self.ns:self.r])
        self.reward = brain.tensor_to(tmp_memory[:,self.r])
        return self.state,self.action,self.next_state,self.reward
```