

決定木とニューラルネットワークを用いたアンサンブル時系列予測 Ensemble Time Series Prediction using Decision Trees and Neural Networks

加藤 正峰[†] 六井 淳[†]
Masamine Kato Jun Rokui

1 はじめに

今日の社会は数多くの情報で溢れている。なかでも、時間経過に応じて変動する情報は時系列データと呼ばれる。このような時系列データに基づき、未来の時系列を予測することを時系列予測と呼ぶ。時系列予測を行うことができるモデルとして、Recurrent Neural Network (RNN) [1] や Long Short-Term Memory (LSTM) [2] といったニューラルネットワークに基づく機械学習モデルが利用されている [3]。これらの機械学習モデルは予測精度が優れる一方、大幅な学習時間を要するという課題がある。より単純なモデルを用いることで、計算時間を削減することができるが、表現力や予測精度が低下する。この問題に対する手法として、単純なモデルを複数用いることで予測精度を安定させる、アンサンブル学習 [4] という手法が存在する。Xin Hu らの研究 [5] では、顧客離れを予測するためのモデルとして、決定木 [6] とニューラルネットワークの予測を組み合わせる手法を提案している。この手法により、それぞれのモデルを単一で用いるより、予測精度が向上したことが報告されている。

本研究では、アンサンブル学習に基づき、学習が高速なモデルを複数用いることで、多くの計算を必要とせずに良い予測精度が見込める時系列予測手法を提案する。アンサンブル時系列予測に用いるモデルは、決定木とニューラルネットワークである。決定木は、木構造に基づく機械学習手法であり、学習速度や予測結果の可読性に長けている。今回は決定木のなかでも初期の手法に当たる Classification And Regression Tree (CART) [7][8] を用いる。ニューラルネットワークは、計算量が少ない Echo State Network (ESN) [9] を用いる。ESN は学習対象の重みを絞ることで高速な学習を実現するニューラルネットワークのモデルである。これらの高速な学習が可能なモデルを併用することで、学習時間と予測精度において良い結果を示す時系列予測手法を提案する。

2 関連技術

2.1 Classification And Regression Tree (CART)

Classification And Regression Tree (CART) [7][8] は、データ構造の一種である木構造を利用した決定木と呼ばれる機械学習手法である。特徴量の重要度や特徴量同士の関係を図示することができるため、分析結果を視覚的に捉えることができる手法として知られる。決定木の構造を図 1 に示す。

提案手法で用いる CART は、回帰問題を扱う回帰木である。回帰木は決定木のなかでも時系列予測との親和性が高く、ランダムフォレスト [10] の前例から、モデ

[†] 静岡県立大学大学院経営情報イノベーション研究科, Graduate School of Management and Information of Innovation, University of Shizuoka

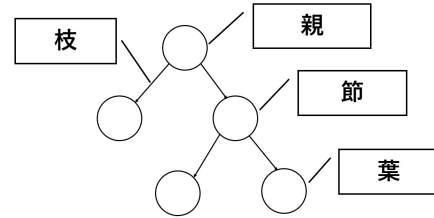


図 1 決定木の構造

ルを並列的に使用することが期待できる。CART が回帰問題において、節や葉にあたるノードを増やす際、分割基準として Mean Squared Error (MSE) が用いられる。MSE は実現値と予測値の差が小さいほど、小さな値をとる評価指標である。実現値 y に対する予測値を \hat{y} 、データの総数を n とすると、MSE は

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (1)$$

となる。MSE を用いて CART を構築する際のアルゴリズムを **Algorithm 1** に示す。

Algorithm 1 回帰木の構築

Input: 訓練データ X (特徴ベクトルの次元数 d)

- 1: **if** X がすべて同一のクラスに所属 **then**
- 2: 木の成長を終了
- 3: **end if**
- 4: MSE を基準として、 d 個の変数の中から X を最適に分割する変数と分割点を決定
- 5: 4 で決定した変数と分割点を用いて X を X_{left} と X_{right} の二つの部分集合に分割
- 6: X_{left} , X_{right} それぞれに対して再帰的に 1 を実行し、木を成長させる

2.2 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) は、ニューラルネットワークを用いた機械学習モデルのひとつである。モデルの特徴として、ニューラルネットワークの隠れ層が再帰的な構造を持つことにより、入力データの時系列構造を踏まえた予測を行うことができる点があげられる。RNN の構造を図 2 に示す。

RNN に入力された情報は、入力層から隠れ層へ伝播する。入力時系列数を n 、入力を x 、入力層のノード j から隠れ層のノード i への重みを w_{ij} 、隠れ層のノード数を m 、隠れ層のノード i から隠れ層のノード l への重みを w_{li} とすると、出力 u は、

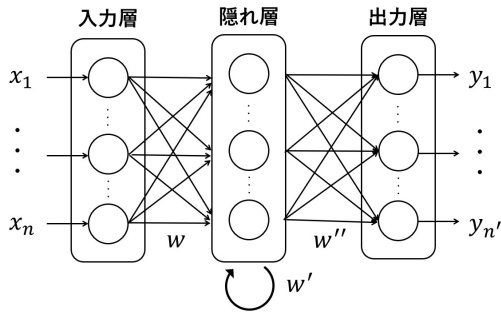


図 2 RNN の構造

$$u_i^t = \sum_{j=0}^n w_{ij}x_j^t + \sum_{l=1}^m w_{li}x_l^{t-1} \quad (2)$$

となる。次に、活性化関数を θ とすると、出力 z は、

$$z_i^t = \theta(u_i^t) \quad (3)$$

となる。そして、隠れ層のノード i から出力層のノード k への重みを w''_{ik} とすると、出力 v は、

$$v_k^t = \sum_{i=0}^m w''_{ik}z_i^t \quad (4)$$

となる。そして、活性化関数を f とすると、出力 y は、

$$y_k^t = f(v_k^t) \quad (5)$$

となる。層数が多い RNN では、出力層から遠い層ほど、重みを更新するための計算が煩雑になるため、誤差逆伝播法 [9] に基づいて重みが更新される。時系列予測を扱う RNN では、時間方向にも局所勾配を伝播させることを考慮した誤差逆伝播法である Backpropagation Through Time (BPTT) [9] という学習アルゴリズムがしばしば用いられる。これにより時系列構造を踏まえた予測が可能となるが、勾配消失問題 [11] が生じるため、RNN が扱うことのできるデータの長さには限度がある。この点を解消すべく提案された機械学習モデルが、Long Short-Term Memory (LSTM) である。

2.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) は、RNN の派生形である。RNN の勾配消失問題を解消し、長期的な記憶保持を実現した機械学習モデルである。LSTM の構造を図 3 に示す。

LSTM には長期的にデータを保持するための記憶セル C 、ユニット内で通過させる情報を選択するための忘却ゲート f 、入力ゲート i 、出力ゲート o を有している。

忘却ゲートは通過させる情報を取捨選択する役割を果たす。時刻 t における入力を x 、時刻 $t-1$ における隠れ層の状態を h_{t-1} 、忘却ゲートの重みを W_f 、忘却ゲートのバイアスペクトルを b_f とすると、忘却ゲートの出力は、

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (6)$$

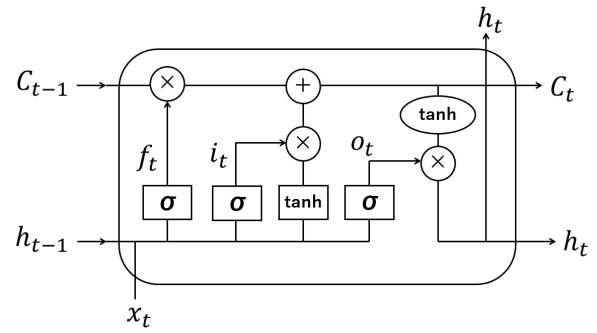


図 3 LSTM の構造

となる。入力ゲートは、記憶セルの状態を更新する役割を果たす。入力ゲートの重みを W_i 、入力ゲートのバイアスペクトルを b_i とすると、入力ゲートの出力は、

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (7)$$

となる。出力ゲートは、LSTM 全体における出力の調整の役割を果たす。出力ゲートの重みを W_o 、出力ゲートのバイアスペクトルを b_o とすると、出力ゲートの出力は、

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (8)$$

となる。また、時刻 t における記憶セル c_t の状態は、出力ゲートの重みを W_c 、出力ゲートのバイアスペクトルを b_c とすると、

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (9)$$

で表される。そして、時刻 t における隠れ層 h_t の状態は、

$$h_t = o_t * \tanh(c_t) \quad (10)$$

で表される。なお、学習パラメータの更新は、RNN と同じく BPTT がしばしば用いられる。

2.4 Echo State Network (ESN)

Echo State Network (ESN) は Recurrent Neural Network (RNN) の派生形である。RNN はニューラルネットワークを用いた機械学習モデルであり、再帰的な構造によって時系列構造を踏まえた予測を行うことができる。ESN の基本的な構造は RNN と共通しているが、隠れ層の重みが固定されたニューラルネットワークとなり、これをリザーブと呼ぶ。リザーブを用いることで、過去に投入された時系列情報が残る状態 (エコー状態) を生み出す。学習の対象となる重みは、リザーブから出力層にかけての重みのみである。このように、学習の対象とするパラメータを絞ることで、高速な学習を実現している。ESN の構造を図 4 に示す。

時刻 $t+1$ におけるリザーブの状態 $x(t+1)$ は、時刻 $t+1$ における入力を $u(t+1)$ 、時刻 t におけるリザーブの状態を $x(t)$ 、入力層のノードからリザーブのノードへの重みを W^{in} 、リザーブ内の重みを W とすると、

$$x(t+1) = f(W^{in}u(t+1) + Wx(t)) \quad (11)$$

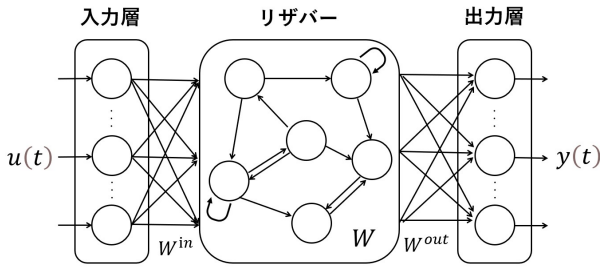


図 4 ESN の構造

と表される。 W^{in} は、正の実数を扱うパラメータを a_{in} とすると、区間 $[-a_{in}, a_{in}]$ の一様分布に従う乱数として生成される。 W は、一様分布、正規分布、2 項分布などの確率分布からランダムに初期設定されたのちに、リザーバーが満たすべき性質である Echo State Property (ESP) [9] が保障されるようスケールが施される。 ESP とは、パラメータの初期設定の影響が時間経過とともに消失する性質である。

時刻 $t + 1$ における出力 $y(t + 1)$ は、リザーバーのノードから出力層のノードへの重みを W^{out} とすると、

$$y(t + 1) = W^{out}x(t + 1) \quad (12)$$

で表される。学習パラメータである W^{out} は、線形回帰もしくはリッジ回帰 [9] によって更新される。線形回帰の場合、リザーバーの大きさを N_x とし、時刻 $n = 1, \dots, T$ のリザーバーの状態 $x(t)$ を連結した行列を $X = [x(1), \dots, x(T)] \in \mathbb{R}^{N_x \times T}$ とし、目標となる出力 $d(t)$ を連結した行列を $D = [d(1), \dots, d(T)] \in \mathbb{R}^{N_y \times T}$ とすると、

$$\hat{W}^{out} = DX^T(XX^T)^{-1} \quad (13)$$

により更新される。リッジ回帰の場合、正則化項の大きさを調整するパラメータを β 、単位行列を $I \in \mathbb{R}^{N_x \times N_x}$ とすると、

$$\hat{W}^{out} = DX^T(XX^T + \beta I)^{-1} \quad (14)$$

により更新される。

2.5 アンサンブル学習

アンサンブル学習 [4] は、あるアルゴリズムに基づいて複数のモデルを融合させ、精度向上を図る手法である。アンサンブル学習のひとつであるバギング法 [4] は、アンサンブル学習のなかでも初期の手法である。回帰問題におけるバギング法は、複数のモデルを独立に学習させたのちに、それぞれのモデルの予測値の単純平均値を最終的な予測値として算出する。一般的なバギング法のアルゴリズムを Algorithm 2 に示す。

3 提案手法

先行研究 [5] では、決定木とニューラルネットワークを組み合わせた予測モデルを用いることで、それぞれのモデルを単体で用いるよりも良い精度が得られることが報告されている。本研究のアンサンブル時系列予測モデル

Algorithm 2 バギング法

Input: 訓練データの説明変数 X_{train} (特徴ベクトルの次元数 d)、訓練データ目的変数 y_{train} 、テストデータ X_{test} (特徴ベクトルの次元数 d)

Output: バギング法による予測値 \hat{y}

- 1: 訓練データ X_{train} , y_{train} からモデルの数に応じてサンプリングを行う
- 2: 各サンプリングデータを各モデルに学習させる
- 3: テストデータ X_{test} を用いて、各モデルから得られる予測値から平均値を算出し、バギング法の予測値 \hat{y} とする

ルにおいては、良い予測精度に加えて、学習に多くの時間を要さないモデルの実現を目指す。アンサンブル時系列予測に用いるモデルは、CART と ESN である。それぞれ単一で学習・予測を行い、それぞれの予測値を結合させる。CART と ESN の予測値を結合させるための式を結合ルールと称する。アンサンブル時系列予測の構造を図 5 に示す。

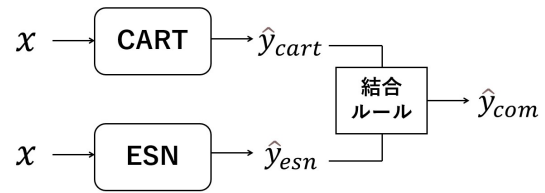


図 5 アンサンブル時系列予測モデルの構造

3.1 結合ルール

時刻 t における CART の予測値を \hat{y}_{cart}^t 、時刻 t における ESN の予測値を \hat{y}_{esn}^t 、CART に対する重みを W_{cart} 、ESN に対する重みを W_{esn} とすると、時刻 t におけるアンサンブル時系列予測モデルの予測値 \hat{y}_{com}^t は、

$$\hat{y}_{com}^t = \frac{1}{\frac{1}{W_{cart}} + \frac{1}{W_{esn}}} \left(\frac{1}{W_{cart}} \hat{y}_{cart}^t + \frac{1}{W_{esn}} \hat{y}_{esn}^t \right) \quad (15)$$

で表される。結合ルールの重み W_{cart} と W_{esn} は、モデルの評価指標として用いられる MSE と Kullback Leibler divergence (KLD) [12] を適用する。KLD は二つの確率分布間の距離を示す指標であり、連続確率分布を P, Q , P, Q の確率密度関数を $p(x), q(x)$ とすると、

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \quad (16)$$

で表される。MSE と KLD は、値が小さいほど、モデルの予測値が実現値に近いことを示す指標である。このため、式 (15) における W_{cart} と W_{esn} は逆数とすることで、モデルの予測精度を反映させることができる。本研究では、MSE を重みとしたアンサンブル時系列予測モデル

を MSE アンサンブル, KLD を重みとしたアンサンブル時系列予測モデルを KLD アンサンブルと称する. 各アンサンブル時系列予測は **Algorithm3** の手順で実行する.

Algorithm 3 アンサンブル時系列予測

Input: 訓練データの説明変数 X_{train} (特徴ベクトルの次元数 d), 訓練データの目的変数 y_{train} , テストデータ X_{test} (特徴ベクトルの次元数 d)

Output: 予測値 \hat{y}_{com}

- 1: X_{train}, y_{train} を用いて CART の学習を行う
- 2: X_{test} を用いて学習済みの CART から予測値 \hat{y}_{cart} を得る
- 3: \hat{y}_{cart} を用いて重み W_{cart} を計算する
- 4: X_{train}, y_{train} を用いて ESN の学習を行う
- 5: X_{test} を用いて学習済みの ESN から予測値 \hat{y}_{esn} を得る
- 6: \hat{y}_{esn} を用いて重み W_{esn} を計算する
- 7: 式 (15) より, 予測値 \hat{y}_{com} を計算する

4 検証実験

検証実験では, 時系列予測における従来手法 (CART, RNN, LSTM, ESN) と提案手法 (MSE アンサンブル, KLD アンサンブル) を用いて検証実験を行う. 実験に用いるデータセットは, 2001 年 12 月 7 日から 2021 年 12 月 3 日までの 150 銘柄の株価の終値である. 株価の終値は, 収益率に変換する正規化を施す. 時刻 t における収益率を r_t , 株価の終値を x_t とすると, 収益率 r_t は,

$$r_t = \frac{x_{t+1} - x_t}{x_t} \quad (17)$$

で表される. 次に, 収益率に変換したデータを, 窓長に応じて束ねる. 今回の検証では窓長を 5 に設定しているため, 5 日分のデータを用いて, その翌日の収益率を予測する形式となる. そして, 訓練データと検証データに分割する. 今回の検証では, 訓練データを 9 割, 検証データを 1 割とする.

モデルの学習から評価までは, 次の手順で行う.

1. 訓練データを用いてモデルの学習を行う.
2. 検証データを用いて, 学習済みのモデルからの予測値を得る.
3. 評価指標を用いて, 検証データと予測値を比較する.

なお, CART は分割点を決めるために必要なサンプルの数を 2, 葉に最小限必要なサンプルの数を 1 と設定しており, この条件を満たさない状態になる前に学習を終了する. ESN は訓練データをすべて学習に使用した時点で学習を終了する. RNN と LSTM のハイパーパラメータの設定は表 1 の通りである. RNN と LSTM にはアーリーストップを設定しており, 20 エポックの間モデルに改善が見られない場合, その時点で学習を終了する.

MSE アンサンブルと KLD アンサンブルに組み込まれている CART と ESN の学習終了条件は, それぞれのモ

表 1 RNN と LSTM のハイパーパラメータ

パラメータ	設定した値
エポック数	2000
学習率	0.0001
中間層数	200
バッチサイズ	16

デルを単体で用いるときの学習終了条件と同様である. 各モデルの学習速度に関しては, 学習開始から学習終了までにかかった時間を評価指標とする. 各モデルの予測精度の評価には KLD を用いる. 150 銘柄分の予測精度をまとめて評価するため, 各銘柄にて最も良い精度 (トップ) を示したモデルと最も悪い精度 (ワースト) を示したモデルを数え上げ, その結果をまとめる.

4.1 学習時間の評価

各モデルの学習にかかった時間を表 2 に示す.

表 2 各モデルの学習時間

モデル	学習時間 (秒)
CART	7.95
RNN	3119.53
LSTM	6403.09
ESN	92.22
MSE アンサンブル	99.99
KLD アンサンブル	100.02

学習時間が最も高速なのは, 6 つのなかで唯一ニューラルネットワークを含まない CART である. 次に高速なのは, ESN である. RNN や LSTM は他のモデルと比べ, 学習に多くの時間を必要とすることがわかる. 各アンサンブル時系列予測モデルの学習時間は, CART と ESN の学習時間の総和に近い. これは, 結合ルールが学習時間にほとんど影響を及ぼさないことを意味していると考えられる.

4.2 予測精度の評価

150 銘柄の検証データを用いた予測結果に, KLD に基づいて評価した結果を表 3 に示す.

表 3 KLD 基準のモデル評価

モデル	トップ銘柄数	ワースト銘柄数
CART	76	0
RNN	0	12
LSTM	0	137
ESN	15	1
MSE アンサンブル	3	0
KLD アンサンブル	56	0

トップの銘柄数が最多だったモデルは CART であり, 次点が KLD アンサンブルである. RNN と LSTM はトップの予測結果だった銘柄がなく, 多くの銘柄でワーストを示している. CART と各アンサンブル時系列予測モデルはワーストの銘柄数が 0 とあるが, これは RNN と LSTM がほとんどの銘柄でワーストを示していることに起因している. RNN と LSTM の予測結果を省いた

結果を表 4 に示す。また、各モデルの予測値を確認するため、ANA の収益率予測結果を図 6 に示す。

表 4 KLD 基準のモデル評価 (RNN, LSTM なし)

モデル	トップ銘柄数	ワースト銘柄数
CART	76	4
ESN	15	110
MSE アンサンブル	3	33
KLD アンサンブル	56	3

最も良い予測精度を示したのは CART である。CART の予測結果を確認すると、予測の変動が激しいことが分かる。このため、変動の激しい日次データに対する予測に適していると考えられる。

ESN の予測精度はワーストが多いものの、予測結果を確認すると、実現値が大きく変動する部分において予測が大きく変動する傾向にある。このため、大局的な予測に長けているといえる。

各アンサンブル予測モデルの結果について、KLD アンサンブルは CART に次いで良い予測精度を示している。図 6 を確認すると、実現値が大きく変動する箇所は ESN、その他は CART による予測を反映していた。このため、KLD アンサンブルの場合、結合ルールにより、CART と ESN の予測傾向を組み合わせることができていると考える。

MSE アンサンブルは ESN に次いで悪い予測精度を示しており、同じアンサンブル時系列予測において、KLD と MSE によって精度に差が生じている。このため、MSE アンサンブルの予測精度の問題は、評価指標である MSE にあると考える。今回は KLD 基準で各モデルの予測精度の評価を行っているが、MSE に基づいて 150 銘柄の検証データの予測精度を評価すると、表 5 のようになる。

表 5 MSE 基準のモデル評価

モデル	トップ銘柄数	ワースト銘柄数
CART	0	133
RNN	59	0
LSTM	70	0
ESN	20	16
MSE アンサンブル	1	0
KLD アンサンブル	0	1

KLD 基準の予測精度評価とは異なり、RNN と LSTM の予測精度が高く評価されている。ただし、オリンパスの収益率予測の結果 (図 7) を確認すると、ほぼ横ばいの予測になっていることがわかる。このため、今回の検証実験において、MSE を基準としてモデルを評価することは不適切である。この結果が MSE アンサンブルの予測精度に影響していると考えられる。

5 おわりに

本研究では、高速な学習を可能とするモデルを複数用いるアンサンブル時系列予測という手法を提案した。検証により、CART と ESN の予測の傾向を示し、それぞれの予測を組み合わせることで、学習時間を大幅に増加させることなく、予測精度の改善を図ることができる可

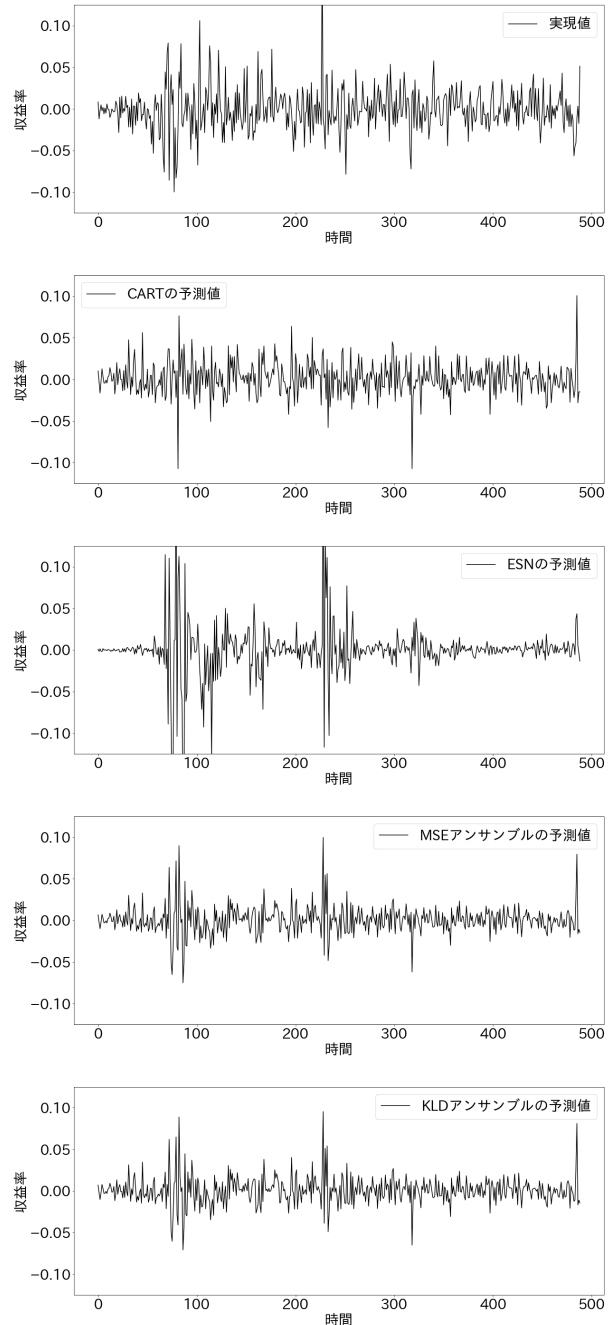


図 6 ANA の収益率予測

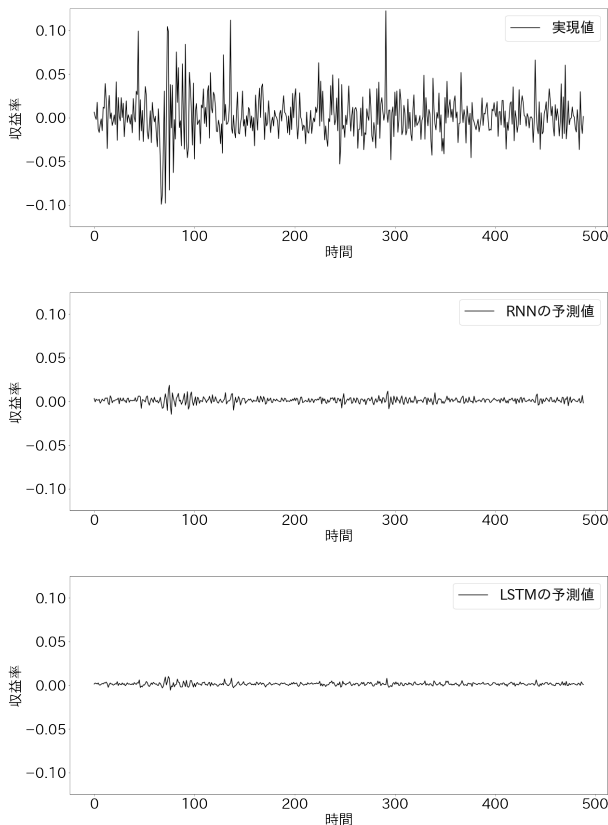


図7 オリンパスの収益率予測

能性を示した。今後の課題は、結合ルールの改善と適切な評価基準の検討である。各モデルの予測の傾向を結合ルールに反映させることにより、アンサンブル時系列予測の改善を図ることを目指す。また、今回の検証により、MSEによるモデル評価が不適切な可能性が示唆された。MSEはCARTの学習に使われている指標であるため、CARTの分割基準を検討することで、アンサンブル時系列予測モデルの精度向上に繋げることを目指す。

参考文献

- [1] 麻生川稔. 時系列信号を認識するリカレント・ニューラルネットワーク. 全国大会講演論文集, 人工知能および認知

科学, pp. 410–411, 1989.

- [2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [3] Pascal A. Schirmer, Iosif Mporas, and Ilyas Potamitis. Evaluation of regression algorithms in residential energy consumption prediction. In *2019 3rd European Conference on Electrical Engineering and Computer Science (EECS)*, pp. 22–25, 2019.
- [4] 上田修功. アンサンブル学習. 情報処理学会論文誌コンピュータビジョンとイメージメディア (CVIM), Vol. 46, No. SIG15 (CVIM12), pp. 11–20, 2005.
- [5] Xin Hu, Yanfei Yang, Lanhua Chen, and Siru Zhu. Research on a customer churn combination prediction model based on decision tree and neural network. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp. 129–132, 2020.
- [6] J. Ross Quinlan. Induction of decision trees. *Machine learning*, Vol. 1, pp. 81–106, 1986.
- [7] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, Vol. 1, No. 1, pp. 14–23, 2011.
- [8] Qiwei Yin, Ruixun Zhang, Yiwei Liu, and XiuLi Shao. Forecasting of stock price trend based on cart and similar stock. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, pp. 1503–1508, 2017.
- [9] 田中剛平, 中根了昌, 廣瀬明. リザーコンピューティング-時系列パターン認識のための高速機械学習の理論とハードウェア-. 森北出版, 2021.
- [10] 波部齊. ランダムフォレスト. 研究報告コンピュータビジョンとイメージメディア (CVIM), Vol. 2012, No. 31, pp. 1–8, 2012.
- [11] 岡谷貴之. 深層学習について. 日本ロボット学会誌, Vol. 33, No. 2, pp. 92–96, 2015.
- [12] 中村燎太, 松宮遼, 高橋一志, 大山恵弘ほか. Kullback-leibler情報量を用いた亜種マルウェアの同定. コンピュータセキュリティシンポジウム 2013 論文集, Vol. 2013, No. 4, pp. 877–884, 2013.