Vol.2022-HPC-185 No.15
2022/7/28

機械学習に基づくジョブスケジューリングのための GANによるデータ拡張

石井 翔[1,a)]　高橋 慧智[1,2,b)]　下村 陽一[2,c)]　滝沢 寛之[1,2,d)]

概要：近年，より効率的なジョブスケジューリングを実現するために将来の要求資源量を機械学習によって予測する研究が行われている．しかし，機械学習による要求資源予測で高い精度を達成するためには大量の訓練データが必要となる．そこで，本研究では表生成モデルである TabularGAN を用いてジョブデータを拡張し，訓練データ数を増加させる手法を提案する．提案手法により実データと比較してデータ数を増加させた拡張データを用いて，要求資源量を予測する LSTM モデルを訓練した結果，実データに比べて予測精度が向上することが示された．

1. 緒論

1.1 背景

多数のユーザによって共用される高性能計算システムは，同時に多くのジョブを実行している．あるジョブに対して必要以上の計算資源を割り当てた場合，計算資源が不足するなどの理由で他のジョブの開始実行が遅れてしまう恐れがある．その結果として，高性能計算システム全体の利用効率が低下する．

近年，高性能計算システムは大規模化・複雑化し，投入ジョブ数も増加している．ジョブスケジューリングは組合せ最適化問題であるため，ジョブ数と計算資源量の増加にともない，最適なスケジュールを厳密に求めるための計算量は急激に増加し，最適なスケジュールを実用的な時間内に求めることは困難となる．そこで，将来の要求資源量を予測する研究が注目されている．要求資源量とはジョブが実行される上で必要となる，CPU やメモリなどの計算資源量である．要求資源量を予測することにより，高性能計算システムのジョブスケジューリングや計算資源管理の効率化が期待される．そのため，機械学習を用いて将来のジョブ投入数や要求計算資源量を予測する研究が行われている [1].

Thang らの研究によると，機械学習による要求資源量の予測精度を向上させるためには，大量の訓練データが必要となる [2]．すなわち，機械学習に基づく要求資源量予測に関する研究を行う上で，訓練データの不足が課題となっている．これは投入ジョブの傾向が高性能計算システムごとに異なることや，要求資源量のパターンが多種多様であるためである．さらに収集したジョブデータに欠損がある可能性もあり，十分な量のジョブデータを収集することは困難である．そこで，ジョブスケジューリング研究のための，機械学習に基づく要求資源量予測における訓練データと，統計的特徴量が類似するジョブデータを拡張することで，機械学習における訓練データの不足を解消されることが期待されている．

1.2 目的

本研究では機械学習モデルによるジョブデータの拡張を目的とする．拡張したジョブデータを使用して，要求資源量を予測するモデルを訓練することで, その予測精度が向上することを示す．具体的には，将来のジョブ投入数を予測する機械学習モデルの予測精度向上を目指し，Generative Adversarial Network (GAN) [3] を用いて訓練データとなるジョブデータを拡張する．

1.3 本論文の構成

本論文は全 5 節からなる．1 節では，本論文の背景および目的を提示した．2 節では，提案手法および評価手法に関連する研究について説明する．3 節では，提案する TabularGAN を応用したジョブデータの拡張手法について説明する．4 節では，拡張データを評価した結果の説明および，その考察を行う．5 節では，本論文の結論と今後の課題について述べる．

[1] 東北大学大学院情報科学研究科
[2] 東北大学サイバーサイエンスセンター
[a)] sho.ishii@hpc.is.tohoku.ac.jp
[b)] keichi@tohoku.ac.jp
[c)] shimomura32@tohoku.ac.jp
[d)] takizawa@tohoku.ac.jp

ⓒ 2022 Information Processing Society of Japan

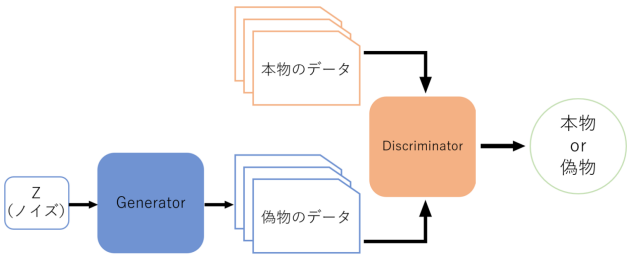

図 1 GAN の概要

2. 関連研究

本節では，まず，データ生成手法である GAN を概説する．その後，現行の機械学習に基づくジョブスケジューリングの課題点を示すため，既存の機械学習モデルを紹介する．

2.1 Generative Adversarial Nerwork

GAN はデータ生成用の機械学習モデルである．GAN は Generator と Discriminator の 2 つのニューラルネットワークで構成される．図 1 に GAN の概要を示す．

データ x に対する，Generator の分布を P_g，入力ノイズに対する事前分布を $P_z(z)$ とし，入力ノイズからデータ空間へのマッピングを $G(z;\theta_g)$ とする．ここで θ_g は G のパラメータである．Discriminator の出力は 1 つのスカラ値であり，このモデルを $D(x;\theta_d)$ とする．ただし，$D(x)$ は x が実データである時の分類確率を示す．

Discriminator は，実データと Generator が生成したデータを正しく分類する確率が最大となるように訓練する．一方，Generator は $log(1 - D(G(z)))$ が最小となるように訓練する．すなわち，Generator が生成したデータを，Discriminator が実データであると分類する確率を最大化させるように Generator を訓練する．言い換えると，以下の価値関数 $V(D,G)$ が，Generator は最小，Discriminator は最大になるように訓練される．なお，p_{data} は実データの分布とする．

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

図 2 に訓練の概要を示す．実データの分布を黒の点線，生成データの分布を緑の実線，Discriminator による識別分布を青の点線で表す．2 本ある直線のうち，下の直線はノイズである z がサンプリングされる領域で，一様サンプルとしている．一方，上の直線は，x の領域の一部である．2 本の直線を繋ぐ矢印は，ランダムノイズから Generator によって生成されるデータの分布を表している．

図 2 (a) において，D は部分的には正確な分類器となっている．図 2 (b) は，D を訓練することにより， 実データと生成データを識別するようになり，$D(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$ に収束することを示している．図 2 (c) において，D の勾配を基に，G を更新することにより，G(z) が実データとして分類される可能性が高い領域に分布が近づく．図 2 (d) では，十分な学習の後，$p_g = p_{data}$ となり，Discriminator は 2 つの分布を区別することができなくなることを示している．このようにして GAN は，実在するデータに類似したデータを生成する．

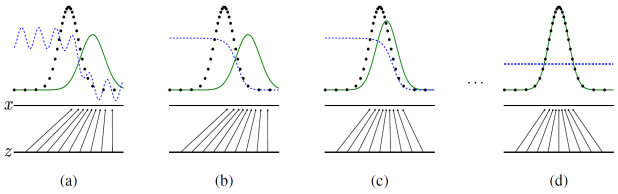

図 2 GAN の訓練過程 [3]

2.2 機械学習に基づくジョブスケジューリング

本節では，機械学習に基づくジョブスケジューリング研究を 2 例紹介する．その後，現在の機械学習における課題を示す．

2.2.1 Deep Reinforcement Agent for Scheduling

Deep Reinforcement Agent for Scheduling (DRAS) は Fan らによって提案された最適なスケジューリングポリシーを自動的に学習する，強化学習ベースのフレームワークである [4]．DRAS の目的は高性能計算システムのシステム管理者が特定のシステムやワークロードに対して最適なスケジューリングポリシーを構築することである．

DRAS は，図 3 に示すようにスケジューラをエージェントとして表現し，システム全体の利用効率やジョブの待ち時間などのスケジューリング性能を最適化することを目的とし，ジョブ割り当ての意思決定を行う．DRAS の特徴として，大規模ジョブの待ち時間減少のための計算資源予測と，計算資源がアイドル状態になることを防ぐためのバックフィリングがある．Fan らは文献 [4] の学術的貢献として以下の 3 つをあげている．1 つ目は深層強化学習を活用し，高性能計算システムのためのスケジューリングモデルの主要な特徴を組み込んだ新しいスケジューリングモデルである DRAS を設計したことである．2 つ目はスケジューリング環境 (システムおよびワークロード) を自動的に学習し，最適なスケジューリングポリシーに迅速に収束することが可能なことである．3 つ目は DRAS が，既存の人の手によるスケジューリング手法や，最適化手法と比較し，長期的なスケジューリング性能の向上および動的なワークロードの変化に対応したことである．

2.2.2 エージェントに基づく動的スケジューリング

Aydin らは，システム全体の実行時間を短縮させること

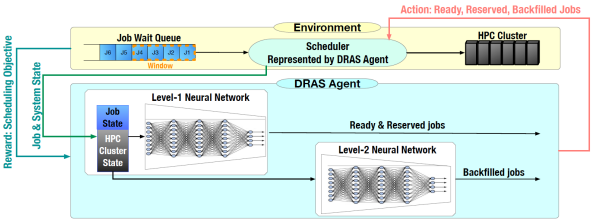

図 3 DRAS の概要 [4]

を目的として，強化学習によって訓練される知的エージェントの新たな学習アルゴリズムを提案している [5]．Q 学習では，過去に各行動にどのような信用を付加したか管理することができない．そこで，Aydin らは Q 学習に対し，Hard c-Means という距離関数を導入することで，過去の経験を反映することを可能にした．

システムは，エージェントとシミュレーション環境によって構成されている．エージェントはシミュレーション環境から現在の状況についての情報を受け取る単純な観測モジュール，一連の行動ルールと意思決定者からなる認知モジュール，メッセージを生成し，シミュレーション環境に送り返す行動モジュールから構成されている．シミュレーション環境は，マシンに割り当てるジョブを選択する必要がある際に，エージェントと通信することで，スケジューリングプロセスを動的にシミュレーションする役割を果たす．

何らかのイベントが発生するたびに，シミュレーション環境はエージェントと通信し，適切な行動ルールを特定するよう依頼する．エージェントはシミュレーション環境から現在の状況を認識し，利用可能な情報に基づいて行動ルールを選択する．シミュレーション環境は選択された行動ルールに基づき，待ち行列の中にある待機中のジョブを 1 つ選択する．そして，そのジョブを作業可能なマシンに割り当て，そのマシンでの終了時間を計測する．

2.2.3 現行の機械学習モデルにおける課題

以上 2 つの機械学習モデルから，より効率的に高性能計算システムを運用するための要件として，要求計算資源量の分析と予測を行うことがあげられる．要求計算資源量を予測することにより，利用可能な計算資源の動的な再配分が容易になり，エネルギーとコストを節約しながら，利用効率を向上させることが可能である．

しかし，一般的に要求計算資源量を予測することは困難である．そこで，Recurrent Neural Network (RNN) に基づいた機械学習モデルが採用され，広範な評価で有望な結果が得られている．さらに，Long Short Term Memory (LSTM) [6] を用いることによって，RNN に比べて平均的に精度の高い予測を行うことが可能である．しかし，LSTM には非常に大量の訓練データと膨大な計算時間を必要とするという課題がある [2]．

3. 提案手法

3.1 概要

2 節で述べたように，効率的に高性能計算システムを運用するために，計算資源量の分析と予測が有用である．しかし，十分な予測精度を達成するために，非常に大量の訓練データが必要であるという課題がある．計算システム毎に要求資源量が様々であることや，同一システム内であっても要求資源量の周期性が一定でないことがあるため，計算資源量を予測する上で有用な訓練データを大量に収集することは多くの場合，困難である．

そこで，本研究では表データの生成モデルである TabularGAN を用いてジョブデータを拡張する手法を提案する．また，TabularGAN で拡張されたデータの類似度は様々であり，そのジョブデータによる精度向上に確率的な揺らぎが生じることが考えられる．そこで，まず TabularGAN によって N 組のジョブデータを拡張する．その中から K 組のジョブデータを使用して機械学習モデルの学習を行う．

3.2 TabularGAN

TabularGAN は Ashrapov らによって開発された表生成用の機械学習モデルである [7]．まず，TabularGAN がデータを拡張する方法を説明する．なお本研究では，連続値のみを扱うため，離散値の拡張方法の説明は割愛する．TabularGAN では表の各列を変分混合ガウス分布 (Variational Gaussian Mixture，以下 VGM とする) によってモデル化する．VGM はモード数を推定した後に，混合ガウスモデルを使用することでサンプルをクラスタリングする．推定するモード数は最大 10 モードである．クラスタリングを終えた後に，前処理として各モード内で正規化を行う．図 4 に例を示す．ここで，c は元のサンプル，η は各モードの平均値，ϕ は各モードの分散値，α は元のサンプルをモード内で正規化した値，β はサンプルが属しているモードを示す one-hot ベクトルを表している．また，青い点線はサンプルの実際の分布，黒い線は VGN によってクラスタリングされた分布を示す．2 つのベクトル α と β の両方が Generator の出力および，Discriminator の入力となる．

Generator は 2 つの段階で連続値を生成する．まず，値を示す α を生成した後に，tanh を活性化関数として用いて，属しているモードを示す β を生成する．Generator にはバッチ正規化 [8] と ReLU[9] を，Discriminator には LeakyReLU[10] とバッチ正規化を用いた多層パーセプトロンを使用している．

TabularGAN の機構を図 5 に示す．まず拡張するデータセットを T_train と T_test に分割する．次に，T_train を使用して，モデルを訓練し，T_train と類似する T_synth を生成する．その後，T_train と T_synth を合成したデータと

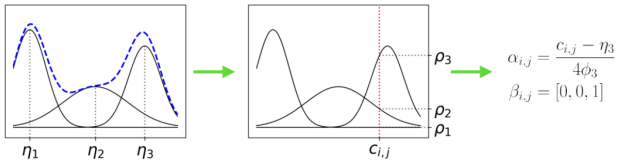

図 4 TabularGAN における連続値の前処理 [7]

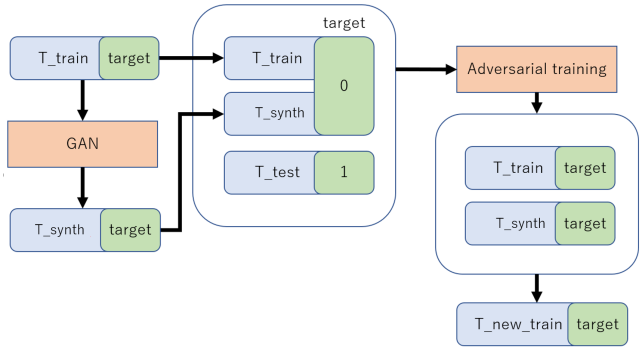

図 5 TabularGAN の概要 ([7] を参考に作成)

T_test を敵対的に学習させる．そして，T_train と T_synth から生成した T_test に類似するデータを，新たなデータセットとする．

4. 評価

本節では，まず評価条件について説明する．次に，提案手法によって拡張したデータが機械学習モデルの訓練データとして有用であるか確認するため，LSTM 予測モデルを用いて評価を行う．その際，5 つの条件を設け，提案手法の有用性を確認する．最後に，LSTM 予測モデルでは確認できないジョブ属性を，統計量によって評価する．

4.1 TabularGAN によるデータ拡張

4.1.1 データ拡張方法

TabularGAN の訓練データとして，スーパーコンピュータ AOBA のサブシステム AOBA-A[11] に投入されたジョブデータを用いる．AOBA とは東北大学サイバーサイエンスセンターに設置されているスーパーコンピュータであり，AOBA-A および AOBA-B の 2 つのサブシステムから構成される．AOBA-A は SX-Aurora TSUBASA (日本電気株式会社製) である．AOBA-A は演算処理を行うベクトルエンジン (VE) 部と，主に OS 処理を行うベクトルホスト (VH) 部によって構成されている．

2021 年 4 月 6 日から 5 月 31 日に AOBA-A の共有キューに投入され，かつ最後まで中止されなかった，ジョブトレースを実データとして使用する．使用したジョブの属性はジョブ投入時刻，並列数，実行時間，使用 CPU 数，使用 VE 数の 5 項目である．

TabularGAN への入力として，実データ全体を 10 分割した後にそれぞれ区域の前から 70%のジョブを抽出したデータを使用した．その 70%のデータの中から，さらにランダムに 70%抽出したデータを訓練データ，残りの 30%をテストデータとして使用した．なお，実データを 10 分割し，前から 70%のジョブを抽出する理由は 4.2.1 節で説明する．

表 1 実験におけるパラメータ

パラメータ名	パラメータ値
バッチサイズ	500
最大エポック数	1,000
patience	25
データ拡張倍率	1.7

TabularGAN モデル訓練時のパラメータを表 1 に示す．バッチサイズは 500，最大エポック数は 1,000，早期打ち切りにおける patience は 25 とした．訓練データ数の 1.7 倍程度のデータを 1 組として拡張し，合計 40 組分のジョブデータを拡張した．

4.2 ジョブ投入数の予測精度に関する評価

4.2.1 評価方法

本評価で用いる LSTM 予測モデルは，1 時間当たりの投入ジョブ数を 24 時間分入力し，その後 24 時間分の 1 時間当たりの投入ジョブ数を予測するモデルである．入力データおよび出力データはともに 1 時間当たりのジョブ数と投入時刻である．LSTM 予測モデルでは周期性を学習させるために，投入時刻を以下の 4 項目に変換して入力する．

$$Daysin = \sin(timestamp \times 2\pi/day) \quad (2)$$

$$Daycos = \cos(timestamp \times 2\pi/day) \quad (3)$$

$$Yearsin = \sin(timestamp \times 2\pi/year) \quad (4)$$

$$Yearcos = \cos(timestamp \times 2\pi/year) \quad (5)$$

ここで，day と year はそれぞれ 1 日および 1 年の秒数である．timestamp はそれぞれのジョブの投入時刻を表す．その後，5 項目それぞれの平均値および分散値を用いて正規化を行う．表 2 に前処理後の実データの例を示す．

使用する実データの周期性を正確に学習させるため，訓練データとテストデータに分割する前に，全体を j 分割する．図 6 に分割方法の概要を示す．訓練データとは，モデルの学習に使用するデータである．テストデータとは，モデルの予測精度を評価する際に使用するデータである．j を決定するため，$j = 2, 3, \ldots, 15$ のそれぞれで，実データで LSTM 予測モデルを訓練し予測を行った．j 分割した後に，それぞれ分割したデータの前から 70%を訓練データ，30%をテストデータとしている．

訓練は最大エポック数 100，早期打ち切りの patience を 10 として行った．結果を図 7 に示す．なお，図中の loss はそれぞれ最終エポック時の訓練データの平均二乗誤差を表しており，test_loss は訓練終了後のテストデータの予測精

表 2　前処理後の実データ

	投入ジョブ数	Daysin	Daycos	Yearsin	Yearcos
0	1	-8.05e-13	1.00	0.996	-0.091
1	3	0.258	0.996	0.996	-0.092
...	...	...	...	...	...
1318	83	-0.500	0.866	0.509	-0.860
1319	46	-0.259	0.966	0.509	-0.861

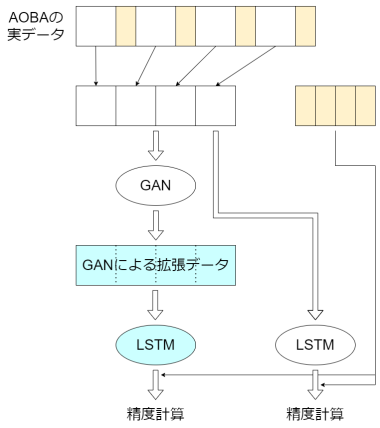

図 6　入力データの分割方法の概要

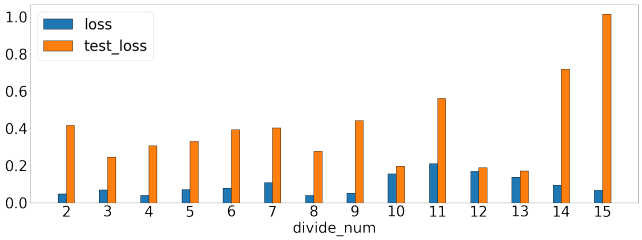

図 7　入力データの分割数と予測精度の関係

度である平均二乗誤差を表している．

図 7 より，検証データとテストデータ，10 分割のときに予測精度が高くなることが分かる．このため，以降の評価では 10 分割を採用し，提案手法の有用性を議論する．

投入ジョブ数予測モデルの概要を説明する．まず入力データとして，24 時間 ×5 項目を LSTM に入力する．その後，LSTM が出力した結果から 120 出力となるように全結合を行い，24 時間 ×5 項目にする．なお，LSTM の隠れ層の次元数は 32 に設定している．

以下 5 つの条件について LSTM 予測モデルで評価する．

条件 1　実際に AOBA-A に投入された実データ

条件 2　TabularGAN で拡張したデータの中からランダムに 1 組選んだデータ

条件 3　実際に AOBA-A に投入された実データの一部を除いたデータ

条件 4　TabularGAN で拡張したデータからランダムに選んだ 20 組のデータ

条件 5　TabularGAN で拡張した全 40 組のデータ

条件 1 は，全体を 10 分割した後に，それぞれ前から 70%のジョブを抽出したデータを訓練データとして使用した．条件 2 では，3 節で説明した TabularGAN で拡張した 40 組のデータの中から 1 組のデータをランダムに選び，それを訓練データとして用いる．条件 3 では，AOBA に投入された実データから一部のデータを取り除いたデータを訓練データとして使用する．具体的には，全体を 10 分割した後に，それぞれ前から 60%を訓練データ，後ろから 30%をテストデータとして使用する．なお，訓練データとテストデータの間にある 10%のデータは条件 3 では使用していない．条件 4 では，TabularGAN で拡張した 40 組のデータの中からランダムに 20 組選び，それぞれを訓練データと検証データに分割した後に，20 組分合成する．条件 5 は，TabularGAN で拡張した 40 組のデータ全てを使用し，訓練データとする．それぞれの条件における LSTM 予測モデルへの入力データの大きさを表 3 に示す．

表 3　実験条件における入力データの大きさ

条件	入力データの大きさ
条件 1	923 ジョブ × 5 項目
条件 2	923 ジョブ × 5 項目
条件 3	792 ジョブ × 5 項目
条件 4	18,460 ジョブ × 5 項目
条件 5	36,290 ジョブ × 5 項目

確率的な揺らぎを考慮し，条件 1 と条件 3 と条件 5 では，それぞれ訓練と予測を 50 回繰り返し，予測精度の平均値を比較する．また，条件 2 と条件 4 ではデータを選ぶ際の確率的な揺らぎを考慮するために，他の条件で行った評価を 10 回繰り返し，その平均値で評価する．また，エポック数は訓練時のデータ数に対するパラメータの更新頻度を考慮し，条件 1 と条件 2 は 25 エポック．条件 3 と条件 4 は 3 エポック，条件 5 は 1 エポックとした．なお，損失関数は平均二乗誤差，最適化アルゴリズムは Adam[12] を用いる．

テストデータは全ての条件で同一であり，実データ全体を 10 分割した後に，それぞれの区間で，後ろから 30%抽出したデータをテストデータとして使用した．

4.2.2　評価結果

表 4 に各条件下で LSTM 予測モデルの訓練した後の，テストデータの予測値の平均二乗誤差を示す．実データを用いた条件 1 に比べて，拡張データから 1 組のデータを使用した条件 2 は誤差が 4.8%増加している．これは TabularGAN によって拡張したデータは実データの統計的特徴を捉えることは出来ているが，実データに比べ，LSTM 予測モデルに適した統計的特徴は捉えられていないためと考えられる．実データの一部を除いた条件 3 は 5 つの条件の中で最も精度が低くなっている．これは，条件 1 や条件 2 と比較して，条件 3 の訓練データ数が少ないためと考えられる．提案手法で拡張したデータを 20 組用いた条件 4 および 40 組用いた条件 5 は，条件 3 に比べて精度が向上している．また，条件 5 に関しては，実データで訓練を行った条件 1 と比べても，誤差が 1%減少していることが確認できる．これらの結果より，TabularGAN で拡張したデータを用いて，訓

ⓒ 2022 Information Processing Society of Japan

表 4 LSTM 予測モデルによる投入ジョブ数の予測精度

条件	平均二乗誤差
条件 1	0.186
条件 2	0.195
条件 3	0.336
条件 4	0.190
条件 5	0.184

表 5 条件 2 における予測精度

平均二乗誤差
0.1967
0.1996
0.1922
0.1923
0.1880
0.1923
0.1894
0.2008
0.1920
0.2034

表 6 条件 4 における予測精度

平均二乗誤差
0.1969
0.1861
0.1911
0.1922
0.1917
0.1863
0.1850
0.1947
0.1915
0.1849

練データ数を増加させることにより，LSTM 予測モデルによる予測精度の向上が確認された．

表 5 に条件 2，表 6 に条件 4 の 10 回分の予測精度を示す．表 5 より，1 組のデータを使用する際，使用したデータによっては少なくとも誤差が 0.1880 まで改善することがわかる．一方，表 6 より，データ数を 20 組に増やしたとしても，試行によっては 0.1969 まで悪化していることがわかる．これらより，たとえデータ数を増加させたとしても，統計的なばらつきにより，精度が悪化してしまう可能性があることが確認できる．

表 7 実データの統計量

	平均値	最大値	最小値	標準偏差	中央値
使用 CPU 数	16.4	2048	1	121.7	2
使用 VE 数	7.90	256	0	12.67	8
実行時間	2443.6	1694922	1	21428.6	402
並列数	1.24	32	1	1.83	1

表 8 拡張データの統計量

	平均値	最大値	最小値	標準偏差	中央値
使用 CPU 数	4.88	64	2	5.53	2
使用 VE 数	13.1	256	1	19.0	8
実行時間	8582.5	262212	5	20366.0	408
並列数	2.40	32	1	2.72	1

4.3 ジョブ投入数以外のジョブ属性の評価

本研究では TabularGAN を用いて，投入時刻，使用 CPU 数，使用 VE 数，実行時間，並列数の 5 属性を拡張した．LSTM 予測モデルでは，AOBA への 1 時間当たりのジョブ投入数を評価した．1 時間当たりのジョブ投入数は，各ジョブの 5 つのジョブ属性のうち投入時刻にのみ影響を受けている．そのため，残りの 4 つのジョブ属性は評価されていない．そこで，残りの 4 つのジョブ属性は統計量を用いて実データとの類似性を評価する．

表 7 と表 8 に，実データと拡張データを比較した結果を示す．表 7 と表 8 より，使用 CPU 数は，実データに対して拡張データの平均値が 4 分の 1，最大値が 32 分の 1，標準偏差が約 24 分の 1 となっている．最小値および中央値は，実データと類似する値を示している．使用 VE 数は，実データに対して拡張データの平均値が約 1.6 倍，標準偏差は約 1.5 倍，最大値と最小値および，中央値は近い値を示している．使用 CPU 数に比べると，使用 VE 数は，特徴を捉えて拡張されていると確認できる．実行時間は，実データに対して拡張データの平均値が平均値が約 4 倍，最大値は約 10 分の 1，標準偏差は約 2 分の 1 倍となっており，最小値と中央値は近い値を示していることがわかる．並列数は平均値が 2 倍であるが，他の項目は近い値を示している．他の 3 つのジョブ属性に比べて，1 番特徴を捉えられていると考えられる．

図 8，図 9，図 10，および図 11 に各ジョブの使用 CPU 数，使用 VE 数，実行時間，並列数とジョブ投入時刻の関係を示す．図 8~11 から，実データは数値の小さい値に多く存在していることがわかる．特に，使用 CPU 数と実行時間ではその傾向が大きい．このような特徴を捉えた結果，TabularGAN で拡張したデータは小さい値に集まっており，値の大きなデータを生成しきれていないと考えられる．これらの理由として，中央値から離れていて，かつ近傍にデータ数が少ない場合は，混合ガウスモデルにおいて，小さい値と同じモードに属していると認識されているということが考えられる．その結果として，値の大きいデータの生成が困難となっている．

5. 結論

近年，高性能計システムは複雑化し，投入ジョブ数も増加している．ジョブスケジューリングは組合せ最適化問題であるため，厳密なスケジュールを求めることが困難となっている．そこで，機械学習に基づくジョブスケジュー

ⓒ 2022 Information Processing Society of Japan

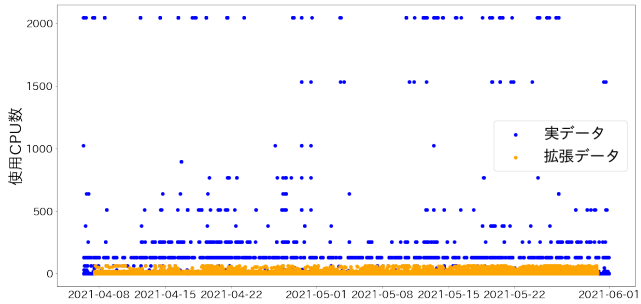

図 8 使用 CPU 数の比較

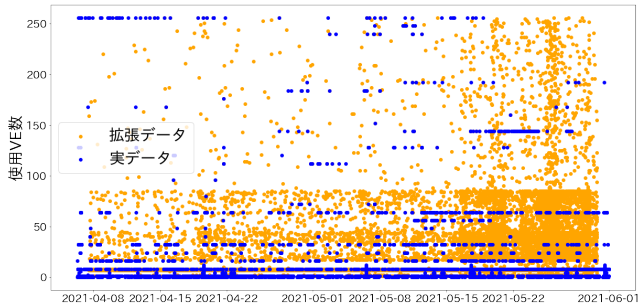

図 9 使用 VE 数の比較

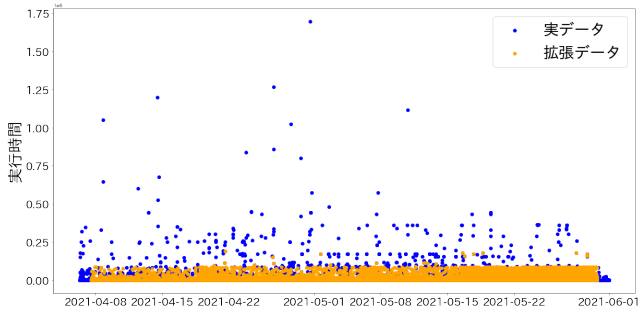

図 10 実行時間の比較

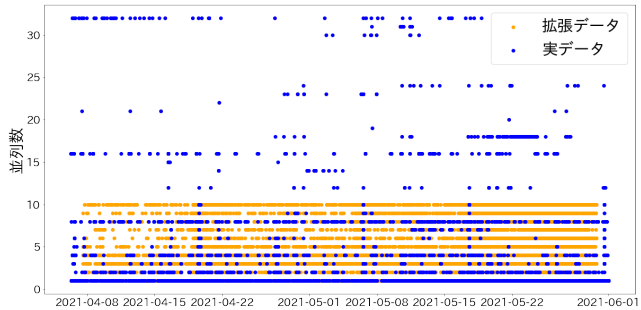

図 11 並列数の比較

リング手法により，スケジューリングの効率化が期待されている．しかし，機械学習において予測精度を向上させるためには大量の訓練データが必要となるため，訓練データの不足が課題となっている．

そこで，本研究ではジョブスケジューリング研究における訓練データの不足に対する解決手法として，実測データに類似するデータを，TabularGAN によって拡張することを提案した．

提案手法の妥当性を評価するため，LSTM に基づくジョブ投入数予測モデルによる評価を行った．評価結果より，拡張データを増加させることにより，実データに比べて，予測精度が向上することを確認できた．最後に，LSTM 予測モデルで評価していないジョブ属性について，統計量を用いて評価した．評価結果として，一部の拡張データは，実データの特徴を捉え切れていないことが確認された．

以上より，提案手法によって，LSTM 予測モデルの予測精度向上が確認された．これにより，提案手法を用いることで，機械学習に基づくジョブスケジューラの訓練データとして，拡張したデータを用いることが可能であることが明らかとなった．

今後の課題は 3 点ある．1 点目は，LSTM 予測モデルにおける精度向上に関わる条件の調査である．4 節で，訓練データ数を増加させることにより，予測精度を向上させることが可能であることを示した．しかし，表 5 と表 6 から予測精度の向上に関与する他の条件が存在することが確認された．この条件を調査し解明することにより，より予測精度の向上するデータを拡張することが期待される．2 点目は，より類似度の高いジョブ属性の拡張である．TabularGAN のアルゴリズムの変更や，ハイパーパラメータの調整，ジョブデータの前処理方法の変更などを行うことにより，より類似度の高いジョブデータの拡張が期待できる．3 点目は，複数のジョブ属性を必要とする機械学習モデルによる評価である．本研究で用いた，評価方法である LSTM 予測モデルは投入ジョブ数のみを予測するモデルであり，実用性には未だ多くの課題がある．より実用的な機械学習モデルによって評価を行うことにより，様々な種類の機械学習モデルの訓練データ収集に対する効率化が期待できる．

謝辞 本研究は，文部科学省「次世代領域研究開発」量子アニーリングアシスト型次世代スーパーコンピューティング基盤の開発，および JSPS 科研費 21H03449 の支援による．

参考文献

[1] Sisheng Liang, Zhou Yang, Fang Jin, and Yong Chen. Data centers job scheduling with deep reinforcement learning. In *Advances in Knowledge Discovery and Data Mining*, pp. 906–917, 2020.

[2] Thang Le Duc, Rafael Garcia Leiva, Polo Casari, and Per-Olov Ostberg. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. In *ACM Computing Surveys*, Vol. 52, pp. 1–39, 2020.

[3] Ian Goodfellow. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 27, 2014.

[4] Yuping Fan and Zhiling Lan. DRAS-CQSim:A reinforcement learning based framework for HPC cluster scheduling. *Software Impacts*, Vol. 8, , 2021.

[5] M.Emin Aydin and Ercan Oztemel. Dynamic job-shop

ⓒ 2022 Information Processing Society of Japan

scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, Vol. 33, No. 2, pp. 169–178, 2000.

[6] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation 9*, Vol. 8, pp. 1735–1780, 1997.

[7] Insaf Ashrapov. Tabular GANs for uneven distribution, 2020. arXiv:2010.00638.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[10] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015. arXiv:1505.00853.

[11] サブシステム AOBA-A. `https://www.ss.cc.tohoku.ac.jp/sx-aurora/`.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. arXiv:1412.6980.

ⓒ 2022 Information Processing Society of Japan