JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 11, pp. 147-155, November 2022
https://doi.org/10.9708/jksci.2022.27.11.147

Performance Analysis of Trading Strategy using Gradient Boosting Machine Learning and Genetic Algorithm

Phil-Sik Jang*

*Professor, Dept. of Air Transportation and Logistics, Sehan University, Dangjin, Korea

[Abstract]

In this study, we developed a system to dynamically balance a daily stock portfolio and performed trading simulations using gradient boosting and genetic algorithms. We collected various stock market data from stocks listed on the KOSPI and KOSDAQ markets, including investor-specific transaction data. Subsequently, we indexed the data as a preprocessing step, and used feature engineering to modify and generate variables for training. First, we experimentally compared the performance of three popular gradient boosting algorithms in terms of accuracy, precision, recall, and F1-score, including XGBoost, LightGBM, and CatBoost. Based on the results, in a second experiment, we used a LightGBM model trained on the collected data along with genetic algorithms to predict and select stocks with a high daily probability of profit. We also conducted simulations of trading during the period of the testing data to analyze the performance of the proposed approach compared with the KOSPI and KOSDAQ indices in terms of the CAGR (Compound Annual Growth Rate), MDD (Maximum Draw Down), Sharpe ratio, and volatility. The results showed that the proposed strategies outperformed those employed by the Korean stock market in terms of all performance metrics. Moreover, our proposed LightGBM model with a genetic algorithm exhibited competitive performance in predicting stock price movements.

▸ **Key words**: Algorithmic trading, Stock market, Gradient boosting, Machine learning, Genetic algorithm

[요 약]

본 연구에서는 그래디언트 부스팅 기계학습과 유전 알고리즘을 이용하여 일별 주식 포트폴리오를 동적으로 구성하는 시스템을 구축하고 트레이딩 시뮬레이션을 통해 성능을 분석하였다. 이를 위해 유가증권시장과 코스닥시장에 상장된 종목들의 가격 데이터 및 투자자별 거래정보를 포함한 다양한 데이터를 수집하고, 전처리 과정과 변수가공을 통해 학습-예측에 이용될 변수들을 생성하였다. 첫 번째 실험에서는 예측정확도와 정밀도, 재현율 및 F1 점수 등 네 가지 지표를 활용하여 그래디언트 부스팅 기법들(XGBoost, LightGBM, CatBoost)의 성능을 비교 평가하였다. 두 번째 실험에서는 전 단계에서 선택된 LightGBM과 유전 알고리즘을 적용하여 상장 종목들의 일별 수익 여부를 학습-예측하였다. 그리고 예측된 수익 발생확률을 바탕으로 종목을 선별하여 트레이딩 시뮬레이션을 시행하고, CAGR, MDD, 사프지수 및 변동성 측면에서 코스피, 코스닥 지수와의 성능을 비교 평가하였다. 분석 결과, 제안된 전략들 모두 네 가지 성능평가 지표상에서 시장 평균을 넘어서는 것으로 나타났으며, 그래디언트 부스팅과 유전 알고리즘의 결합이 주식 가격 예측에 효과적으로 이용될 수 있음을 보여주었다.

▸ **주제어**: 알고리즘 트레이딩, 주식시장, 그래디언트 부스팅, 기계학습, 유전 알고리즘

• First Author: Phil-Sik Jang, Corresponding Author: Phil-Sik Jang
*Phil-Sik Jang (phil@sehan.ac.kr), Dept. of Air Transportation and Logistics, Sehan University
• Received: 2022. 10. 18, Revised: 2022. 10. 27, Accepted: 2022. 11. 08.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 기계학습은 일기예보, 자율주행, 음성인식, 인터넷 검색 등 다양한 분야에 활용되고 있으며, 컴퓨터 과학, 기술의 한계를 확장하는 중요한 원동력으로 인식되고 있다. 이러한 기계학습을 자본주의의 대표적 산물인 주식(증권) 시장에 적용하고자 하는 시도들도 활발히 진행되고 있는데, 이들 연구 결과들은 주식 및 암호화폐 시장에서 기계학습이 유용하게 활용될 수 있다는 가능성을 보여준다[1]. Kimoto and Asakawa[2]는 일본 주식시장에서 매수·매도 시점을 예측하기 위해 ANN(Artificial Neural Networks)을 이용하였으며, Kamijo and Tanigawa[3]는 ANN이 주가 패턴을 인식하는 데 효과적임을 보여주었다. 한 개 모델이 아닌 두 개 이상의 모델을 결합하는 하이브리드방식도 시도되고 있는데, 기존의 시계열 분석 방법인 ARIMA(Auto Regressive Integrated Moving Average)와 SVM(Support Vector Machine)을 결합하여 미국 주식시장에 적용하거나[4], ANN과 유전알고리즘을 결합하여 KOSPI 주가 예측에 적용하고자 하는 시도들[5-6]이 발표되었다. 또한 최근 십여 년 동안에는 심층학습(deep learning)이 광범위하게 주목받으면서 이를 활용하여 주가 및 암호화폐 가격 예측에 활용하고자 하는 노력들이 활발하게 진행되어 왔다. Siami-Namini 등의 연구[7]와 Selvin등의 연구[8]는 주가 변동을 예측하는 데 기존의 시계열 방법론인 ARIMA 보다 LSTM(Long Short Term Memory), RNN(Recurrent Neural Network) 및 CNN(Convolution Neural Network)의 성능이 우월함을 보여주었다. 하지만 심층학습의 경우 화상, 음성인식 등 비정형 데이터에서는 탁월한 성능을 보여주지만, 주가 변동을 포함한 정형데이터 대상으로는 의미 있는 성능향상을 보여주지 못하며[9], 알고리즘이 복잡하여 파라메터 설정이 쉽지 않다는 단점이 지적되고 있다[1].

최근 몇 년간에는 전력수요[10] 및 초미세 먼지 예측[11], 부동산 수요예측[12]과 안전 운전자 예측[13] 등 정형데이터를 대상으로 그래디언트 부스팅(gradient boosting) 기반 트리 앙상블 모델을 활용하는 시도가 이어지고 있다. Arik 등[9]은 그래디언트 부스팅이 일반적으로 심층학습에 비해 정형데이터 모델을 좀 더 빠르게 개발할 수 있고, 예측성능이 우수하며, 높은 해석력을 가진다고 주장한다. 이러한 다양한 장점 때문에 코스피 200 주가지수 예측[1] 및 암호화폐 가격 동향 예측[14] 등에 그래디언트 부스팅을 활용하는 연구들이 최근 발표되고 있다. 본 연구에서는 그래디언트 부스팅 알고리즘 중 가장 효과적인 기법을 선별하여, 유전 알고리즘과 결합, KOSPI 및 KOSDAQ 상장 종목들을 대상으로 트레이딩 시뮬레이션을 시행하고 그 성능을 비교, 평가하였다.

II. Related Works

1. Machine learning models

1.1 Ensemble learning method

앙상블 학습법(ensemble learning method)은 한 개의 학습 알고리즘을 단독으로 사용하는 것보다 더 좋은 예측 성능을 얻기 위해 두 개 이상의 학습 알고리즘을 결합하는 방식이다[15-17]. 앙상블 학습의 핵심은 예측정확도가 낮은 모형(약한 학습자: weak classifier)들을 결합함으로써 예측정확도가 높은 모형(강한 학습자: strong classifier)을 만드는 것이다[18]. Siroky[19]는 앙상블 학습이 분류, 회귀, 순위 등에서 기존 방법론 들에 비해 성능향상을 보이며, 이상 징후 탐지, 데이터 스트림 학습과 같은 분석 과제를 처리하기 쉽다고 주장하였다.

앙상블 학습은 일반적으로 배깅(bagging)과 보팅(voting), 스태킹(stacking), 부스팅(boosting) 등으로 분류된다. 배깅은 Bootstrap Aggregation의 약자이며, 같은 모델을 이용하여 샘플을 여러 번 뽑아 Bootstrap을 생성하고, 각 모델을 학습시켜 결과물을 집계(aggregation)하는 방법이다[18]. 보팅은 배깅과 달리, 새로운 자료에 대해 여러 개의 예측 모델 결과들의 가중 투표(weighted vote)를 통해 최종 예측 결과를 결정한다. 스태킹은 여러 개의 상이한 모델들을 활용해 각각의 예측 결과를 도출하고, 그 예측 결과들을 결합해 최종 예측 결과를 만들어내는 방식이며, 부스팅은 연속적, 순차적으로 학습을 수행하는 형태인데, 이전 학습의 결과를 바탕으로 다음 분류기의 샘플 가중치를 조정하고 오차를 교정하는 과정을 반복함으로써 예측의 정확도 향상하는 기법이다[20]. 부스팅 기법 중 향상된 성능으로 최근 주목받고 있는 그래디언트 부스팅은 기울기 하강(gradient descent)을 사용하여 약한 학습자를 단계별로 적층하여 추가함으로써 모델의 손실을 최소화하는 방식이다. 그래디언트 부스팅은 정형데이터 분석 시 높은 예측 성능과 빠른 예측, 적은 메모리 사용 등의 장점 때문에 Kaggle 등 머신러닝 경연대회에 상위 입상한 팀들이 많이 사용하는 것으로 알려졌으며[21] 최근 다양한 분야와 주식 가격 변동 예측에도 활용되고 있다[11-14, 22].

1.2 XGBoost

XGBoost(eXtreme Gradient Boosting)는 Chen and Guestrin[23]에 의해 2016년 개발되었으며, 기존 그래디언트 부스팅 기법의 단점인 과적합(over-fitting) 문제와 느린 수행 시간 등의 문제를 개선함으로써 빠른 실행속도, 높은 예측성능으로 최근 다양한 분야에 적용되고 있다[18]. XGBoost는 의사결정나무 계열의 알고리즘으로, 여러 개의 CART(Classification and Regression Tree)를 묶어 오차 값을 낮추는 부스팅 기법을 활용한다. 과적합을 방지하고 training loss를 최소화하기 위해 의사결정나무의 복잡도를 통제하여 가장 최적화된 모델을 생성하며, 주로 불순도 척도인 지니계수를 분류기준으로 사용한다[23]. Xgboost는 연속형, 범주형 데이터 모두 학습-예측이 가능하며, 모형의 최종 성능에 모든 잎(leaf)이 연관되어 있어 의사결정나무 모형 간 성능 우위를 쉽게 비교할 수 있다. 학습을 tree의 최저 깊이까지 진행한 후, 손실함수가 일정 수준까지 개선되지 않으면 불필요한 부분을 제거하는 가지치기를 역방향으로 진행하며, 이러한 과정을 통해 모형의 과적합을 제어한다[18][23]. 기존의 GBT에 비해 XGBoost는 다수의 CPU 코어들을 동시에 활용하며(parallel computing), 주 메모리를 넘어서는 데이터 처리가 가능(out-of-core computing)하고, 알고리즘과 데이터구조를 최적화(cache optimization)하는 시스템적 특징을 보여준다[23].

1.3 LightGBM

그래디언트 부스팅은 기존 모델들과 비교하면 예측성능이 뛰어나지만 여러 차원의 변수가 포함된 대규모 데이터의 경우 훈련 시간이 오래 걸리고, 메모리 활용이 비효율적이라는 단점이 있다[24]. LightGBM은 이러한 단점을 극복하기 위해 Microsoft에서 2017년 개발되었으며, GOSS(Gradient-based One-Side Sampling)를 활용하여 데이터 일부의 정보 이득을 먼저 빠르게 계산하고 EFB(Exclusive Feature Bundling)으로 특성 요인들을 감소시켜 계산속도를 높이는 방식을 사용한다[24]. 균형나무 분할(level-wise tree growth)방식의 XGBoost와는 달리 LightGBM은 잎 분할방식(leaf-wise tree growth)을 사용한다. 즉, 결정 나무의 균형을 유지하지 않고, 최대 손실 값(max delta loss)을 보이는 잎 노드(leaf node)를 계속하여 분할하면서 결정 나무의 깊이가 깊어지고 비대칭적인 규칙 나무가 생성된다. 이렇게 잎 노드를 지속해서 분할함으로써 만들어지는 규칙 나무를 통해 학습이 반복되며, 최종적으로 균형 나무 분할 방식보다 예측 오류손실을 낮출 수 있게 된다[24]. 이러한 방식을 통해 LightGBM은 XGBoost과 예측성능은 비슷하지만, 더 빠른 학습이 가능하고 더 작은 메모리를 사용하며, 범주형 자료 학습 시 원-핫 인코딩(one-hot encoding)을 사용하지 않고도 최적 변환 후 노드 분할이 가능하다는 장점이 있다[24].

1.4 CatBoost

CatBoost는 Category Boosting에서 유래하였다고 알려져 있으며, 2018년 러시아 기업인 Yandex에서 범주형 데이터 학습 시 과최적화 문제를 해결하고 빠른 학습 속도 구현을 위해 개발되었다[25]. CatBoost는 기존 부스팅 모델과는 달리, 모든 잔여 오차(residual error)를 차례대로 학습하지 않고, 일부 데이터의 잔여 오차를 계산하여 모델을 만들며, 이 모델을 이용하여 나머지 데이터들의 잔여 오차를 계산한다. 또한 ordered boosting 방식에 random permutation 기법을 더해 데이터 순서를 셔플링 해줌으로써 과최적화를 방지한다. 그리고 범주형 데이터 전처리를 위해 random permutation을 적용한 데이터셋에 같은 범주 변수들의 평균 표본 값을 계산, 활용하며, 같은 information gain을 가지는 변수들을 통합하는 feature combination을 이용하여 훈련 시간을 단축한다[25]. 또한 XGBoost와 마찬가지로 균형 나무 분할(level-wise tree growth)방식을 이용하여 규칙 나무를 형성하지만, 대칭적인 나무구조를 이용함으로써 훈련 속도를 높이게 된다. CatBoost는 XGBoost와 LigtGBM을 포함한 다른 그래디언트 부스팅 기법들과는 달리 초기 하이퍼파라미터(hyper-parameter) 설정이 최적화되어있어 하이퍼파라미터 튜닝이 일반적으로 필요치 않다는 장점이 있다[26].

2. Genetic algorithm

유전 알고리즘은 멘델(G. J. Mendel)의 유전법칙과 다윈(C. Darwin)이 주창한 자연선택의 개념을 이용한 최적화 기법으로, 생물의 진화를 모방하여 최적해 또는 유사최적해를 찾아내는 최적화 기법이다[27]. 유전 알고리즘은 해결하고자 하는 문제에 대한 가능한 해(solution)들을 특정한 형태의 데이터구조로 나타낸 다음, 이들을 단계적으로 변형함으로써 더 좋은 해들을 생성한다. 여기에서 해들을 나타내는 데이터구조는 유전자로 표현되며, 더 좋은 해를 만들어내는 과정을 진화로 표현된다[27]. 기존의 최적화 기법들과는 달리 유전 알고리즘은 한 개체 탐색이 아닌 개체군을 대상으로 하는 병렬 탐색이 이루어지며, 탐색 영역과 방향이 초깃값에 과도하게 좌우되지 않고, 진화하는

세대에 따라 확률적으로 변화하기 때문에 전역 최적화(global optimization)를 가능하게 한다[28].

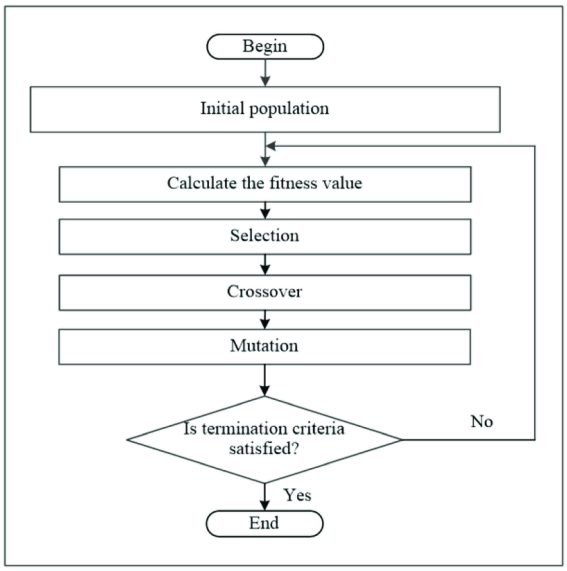

Fig. 1. Flowchart of the Standard Genetic Algorithm

유전 알고리즘의 일반적인 흐름을 나타내면 Fig. 1.과 같다[29]. 본 연구에서는 그래디언트 부스팅을 이용한 학습, 예측에 사용될 변수들 중 성능을 극대화할 수 있는 변수 조합을 찾기 위해 유전 알고리즘을 이용하였다.

III. Data and Experiment

1. Data Set

데이터는 증권사(크레온, 이베스트 투자증권)에서 제공하는 API(Application Programming Interface) 및 한국거래소 웹사이트(www.krx.co.kr)를 활용하여 취합하였다. 취합된 데이터는 2013년 7월 1일부터 2022년 6월30일까지 유가증권(KOSPI)시장과 코스닥(KOSDAQ)시장에 상장되어 거래된 종목(시기에 따라 1,733~2,664개)의 일별 데이터와 분별데이터이다. 분별 취합된 데이터는 분봉(시고저종가), (누적) 거래량, (누적) 거래금액이며, 일별 데이터는 일봉(시고저종가), 거래량, 거래금액과 수급 정보(개인, 기관, 외국인, 증권사, 보험, 투신 등 10개 투자자별 순매수량, 평균단가) 등이다. 또한 일별로 종목별 시가총액 및 외국인 투자 한도, 감리, 관리종목 여부, 유·무상증자 발표 등 종목 관련 정보를 수집하였다. 수집된 데이터들은 데이터 정제(결측, 이상, 중복데이터 처리) 과정을 거치고, feature engineering을 통해 학습-예측에 사용될 변수들로 가공, 생성되었다. 변수가공 및 생성에 활용한 기술적 지표들과 산출 방법은 다음 Table 1과 같으며, 위에서 언급한 일봉, 분봉 및 거래량, 투자자별 거래정보 등과 Table 1 지표들을 포함하여 총 118개의 변수를 생성하였다.

2. Experiments and Method

본 연구에서는 두 단계에 걸쳐 실험을 진행하였다. 첫 단계 실험에서는 그래디언트 부스팅 기법들 중 가장 많이

Table 1. Technical Indicators Used for the Analysis

Indicator	Formula
Range Position	$RP=(\frac{C-L}{H-L})\times 100$ where: C = Closing price, L = Low price, H = High price
Price Rate Of Change	$ROC=(\frac{C_p-C_{p-n}}{C_{p-n}})\times 100$ where: C_{p-n}= Closing price n periods before most recent period
Average True Range	$TR=Max[(H-L), \mid (H-C_p) \mid , \mid (L-C_p) \mid], ATR=(\frac{1}{n})\sum_{i=1}^{n} TR_i$ where: TR_i= A particular true range, n= The time period employed
Simple Moving Average	$SMA=\frac{A_1+A_2+...+A_n}{n}$ where: A= Average in period n, n= Number of time periods
Disparity Index	$DI=\frac{(C_p-nPMAV)}{nPMAV\times 100}$ where: $nPMAV$ = n-Period moving average value
On-Balance Volume	$OBV=OBV_{prev}+\begin{cases} Vol, & \text{if } C>C_{prev} \\ 0, & \text{if } C=C_{prev} \\ -Vol, & \text{if } C<C_{prev} \end{cases}$ where: OBV = Current OBV level, OBV_{prev} = Previous OBV level, Vol = Latest trading volume amount
Moving Average Convergence Divergence	$MACD=\sum_{i=p-9}^{p}(EMA(12)_p-EMA(26)_p)$ where: EMA = Exponential moving average value

활용되는 세 가지 기법(XGBoost, LightGBM, CatBoost)들의 성능 비교와 평가를 통해 본 연구에 적합한 그래디언트 부스팅 기법을 선택하였다. 두 번째 단계 실험에서는 선택된 그래디언트 부스팅 기법을 바탕으로, 유전 알고리즘을 적용하였을 때의 성능향상 여부를 실험하였다.

2.1 Experiments 1

첫 번째 실험에서는 그래디언트 부스팅을 기반으로 한 예측모형의 성능 비교와 평가를 위해 정확도(accuracy)와 정밀도(precision), 재현율(recall) 및 F1 점수(F1- score) 등 네 가지 지표를 사용하였다. 이 지표들은 일반적인 기계학습 이진 분류(binary classification) 모형의 성능 지표로 많이 이용되고 있으며, Table 2와 같이 예측값과 실제 값(positive:참, negative:거짓)을 비교하여 작성되는 혼동행렬을 바탕으로 산출된다[14].

정확도는 전체 데이터 건수 중 예측 결과와 실제가 같은 건수의 비율을 나타내며, 정밀도는 참으로 예측한 건수 대비 실제 참인 건수 비율을 나타낸다. 재현율은 실제 참인 대상 건수 중 예측과 실제 값이 참으로 일치한 건수 비율로 민감도(sensitivity) 또는 TPR(True Positive Rate)라고도 한다. 분류 모델에서 정확도만 가지고 예측성능을 판단할 경우, 모델의 신뢰도가 떨어질 수 있어 정밀도와 재현율을 사용하는 것이 바람직한 것으로 알려져 있다. 하지만 정밀도와 재현율은 서로 반비례하는 성향이 있어, 이 두 지표의 성능을 동시에 고려하는 F1 점수로 예측모형 성능을 전체적으로 평가할 수 있다[14].

Table 2. Confusion Matrix and Four Measures

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection

$$\text{Accuracy} = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-score} = \frac{2TP}{2TP+FP+FN}$$

그래디언트 부스팅 모델 간 예측성능 비교평가를 위해 전체 데이터 중 2013년 7월~2018년 6월까지 60개월 데이터를 학습데이터로, 2018년 7월~2020년 6월까지 24개월 데이터를 테스트용으로 사용하였다. 하이퍼파라미터 튜닝에는 R ver 4.21의 mlr package와 mlr3tuning package를 이용하여 모형별 최적의 하이퍼파라미터를 도출하였다. 알고리즘별로 튜닝에 이용된 하이퍼파라미터 범위는 Table 3과 같다.

Table 3. Range of Values for Hyperparameter Tuning

Model	Range of Values
XGBoost	Booster: [gbtree, dart] Number of Trees: [50, 100, 150, 200] Depth of Tree: [3, 4, 5, 6, 7, 8] Control Depth of tree: [1, 3, 5] Regularization parameter: [0, 0.2, 0.4] Learning Rate: [0.05, 0.1, 0.15, 0.2]
LightGBM	Boosting type: [gbdt, dart] Number of Leaves: [31, 200, 500, 800] Max Depth of Tree: [20, 50, 80, 110] Control Depth of tree: [10, 20, 30] Feature Fraction: [0.5, 1, 1.5] Regularization parameter: [0, 0.2, 0.4] Learning Rate: [0.01, 0.05, 0.1, 0.15]
CatBoost	Number of Trees: [100, 500, 1000] Depth of Tree: [6, 8, 10] Regularization parameter: [1, 3, 10, 100] Learning Rate: Automatic

본 실험에서는 당일 종가 매수 후 다음 날 시초가에 매도하는 전략과 당일 종가 매수 후 다음 날 종가에 매도하는 두 가지 전략을 대상으로 성능 비교평가를 시행하였다. 60개월 동안 매일 상장 종목들을 당일 종가 매수 후 다음 날 시초가 매도 시(전략 1)와 종가 매도 시(전략 2) 각각의 수익 여부를 학습하고, 24개월간 매일 매수 가능 종목의 다음 날 시초가/종가 매도 시 수익 발생 여부를 예측하였다. 그리고 가장 확률이 높은 10개 종목을 매수, 다음 날 시초가/종가에 매도한다고 가정하였을 때, 실제 수익 여부의 정확도, F1 점수 등 지표를 측정하였다.

2.2 Experiments 2

실험 2는 유전 알고리즘을 적용하여 변수를 선별하는 경우, 모든 변수를 이용하였을 때와 비교하여 성능이 향상되는지를 검증하는 것이 목적이다. 즉, 검증 데이터(validation set)를 대상으로 118개의 변수 모두를 이용하고 하이퍼파라미터 튜닝 후 모델을 생성한 경우와 유전 알고리즘을 이용하여 변수를 선택하고, 파라메터 튜닝 후 모델을 생성한 경우를 비교하였다. 유전 알고리즘은 엘리트

보존 방식(elitism)을 이용하고, 모집단의 크기는 50개체, 교차(crossover) 확률과 돌연변이(mutation) 확률은 0.5, 0.03을 사용하였으며, 40세대 동안 연속하여 개선되지 않거나 최대 70세대에 도달하면 종료하도록 하였다.

적용된 기본 전략은 실험 1에서 성능 우위를 보인 당일 종가 매수 후 다음 날 시초가 매도 전략이며, LightGBM을 이용하여 다음 날 수익 발생 여부를 학습, 예측하였다. 그리고 예측된 수익 가능성이 가장 큰 10개 또는 20개 종목을 매일 매수, 다음 날 매도하는 트레이딩 시뮬레이션을 시행하고, 그 결과를 KOSPI 및 KOSDAQ 지수의 수익률과 비교하였다. 과도한 슬리피지(slippage) 발생을 방지하기 위해 일 거래금액 40억 미만 종목들과 관리종목을 제외한 당일 유가증권시장과 코스닥시장의 거래 종목들 데이터를 학습, 예측하고 트레이딩 시뮬레이션에 이용하였다. 트레이딩 시뮬레이션 시에는 증권거래세, 위탁수수료, 유관기관 수수료 및 슬리피지를 고려하여, 매수-매도마다 0.5% 비용이 발생하는 것으로 가정하였다.

최종 성능 비교 및 평가를 위한 지표로는 연평균 복리수익률(CAGR), 최대 자본 인하액(MDD), 샤프지수(Sharpe ratio) 및 변동성(volatility)을 활용하였으며, 샤프지수 산출 시 무위험 수익률은 국고채 금리 3년물의 일별 데이터를 이용하였다. 또한 트레이딩 알고리즘의 검증과 성과분석을 위해 다음 Fig. 2와 같이 시계열 교차검증(time series cross-validation)을 시행하였다. 즉, 학습 시작 시점은 2013년 7월 1일로 고정하였으며, 최근 데이터 2년을 2개월씩 분할하고 그 이전 기간을 학습 후 학습된 모델을 이용하여 순차적으로 테스트하였다.

IV. Results

1. Experiment 1

그래디언트 부스팅 알고리즘 별로 당일 종가에 매수하고, 다음 날 시초가에 매도하는 전략(전략 1)과 종가에 매도하는 전략(전략 2)을 대상으로 한 네 가지 예측 결과 평가지표는 Table 4와 같다. 평가 결과, 정확도는 전략 1의 경우, LightGBM이, 전략 2의 경우는 CatBoost와 LightGBM이 우수한 것으로 나타났다. 정밀도는 전략 1의 경우, CatBoost가, 전략 2의 경우는 XGBoost가 우수한 것으로 나타났으며, 재현율과 F1 점수는 LightGBM이 높은 것으로 나타났다. 근소한 차이지만, 두 전략 모두 LightGBM이 전반적으로 우수한 것으로 평가되었으며, 이에 따라 LightGBM을 최종 모형으로 선정하였다. 그리고 당일 종가 매수 후 다음 날 시초가 매도 전략(전략 1)과 종가 매도 전략(전략 2)의 경우, 전략 1이 모든 지표상에서 성능 우위를 보여주었다.

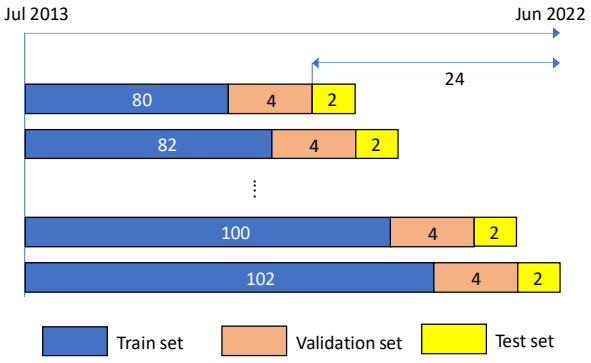

Fig. 2. Cross-validation Time Series Split (Unit: month)

Table 4. Evaluation of Gradient Boosting Classification Algorithms

(a) Strategy 1 (sell on the next day at opening price)

Model	XGBoost	LightGBM	CatBoost
Accuracy	0.751	**0.762***	0.753
Precision	0.776	0.764	**0.779***
Recall	0.946	**0.996***	0.943
F1-score	0.853	**0.864***	0.853

(b) Strategy 2 (sell on the next day at closing price)

Model	XGBoost	LightGBM	CatBoost
Accuracy	0.677	**0.686***	**0.686***
Precision	**0.697***	0.690	0.691
Recall	0.941	**0.989***	0.982
F1-score	0.800	**0.813***	0.811

* Highest value

2. Experiment 2

유전 알고리즘을 이용한 변수 선택 여부 × 일별 매수 종목 개수에 따른 네 개 전략들의 트레이딩 시뮬레이션 결과는 Table 5, Fig 3과 같다. 전략 5(유전 알고리즘 적용, 10개 종목 트레이딩)의 CAGR(27.32%)이 가장 높은 것으로평가 되었으며, 전략 3(모든 변수, 10개 종목 트레이딩)이 두 번째 높은 CAGR(15.06%)을 보였다. 네 개 전략의 CAGR은 모두 10% 이상이며, KOSPI 지수 및 KOSDAQ 지수의 CAGR(5.23%, 1.22%)을 상회하고 있다. 샤프지수 또한 전략 5가 가장 높은(2.22) 것으로 나타났으며, 전략 6(유전 알고리즘 적용, 20종목 트레이딩)이 그다음 높은(1.64) 것으로 평가되었다. KOSPI 및 KOSDAQ의 샤프지수가 각각 0.35, 0.14인 것에 비해 네 개 전략 모두 샤프지수는 1 이상인 것으로 평가되었다.

Table 5. Performance Analysis

Strategy	Variable Selection	Number of Stocks	CAGR (%)	MDD (%)	Sharpe Ratio	Volatility
3	All	10	16.06	-20.85	1.52	14.05
4	All	20	10.47	-20.31	1.25	11.31
5	Genetic Algorithm	10	27.32	-19.18	2.22	14.96
6	Genetic Algorithm	20	15.00	-20.44	1.64	11.94
KOSPI	-	-	5.23	-29.98	0.35	17.46
KOSDAQ	-	-	1.22	-32.61	0.14	22.06

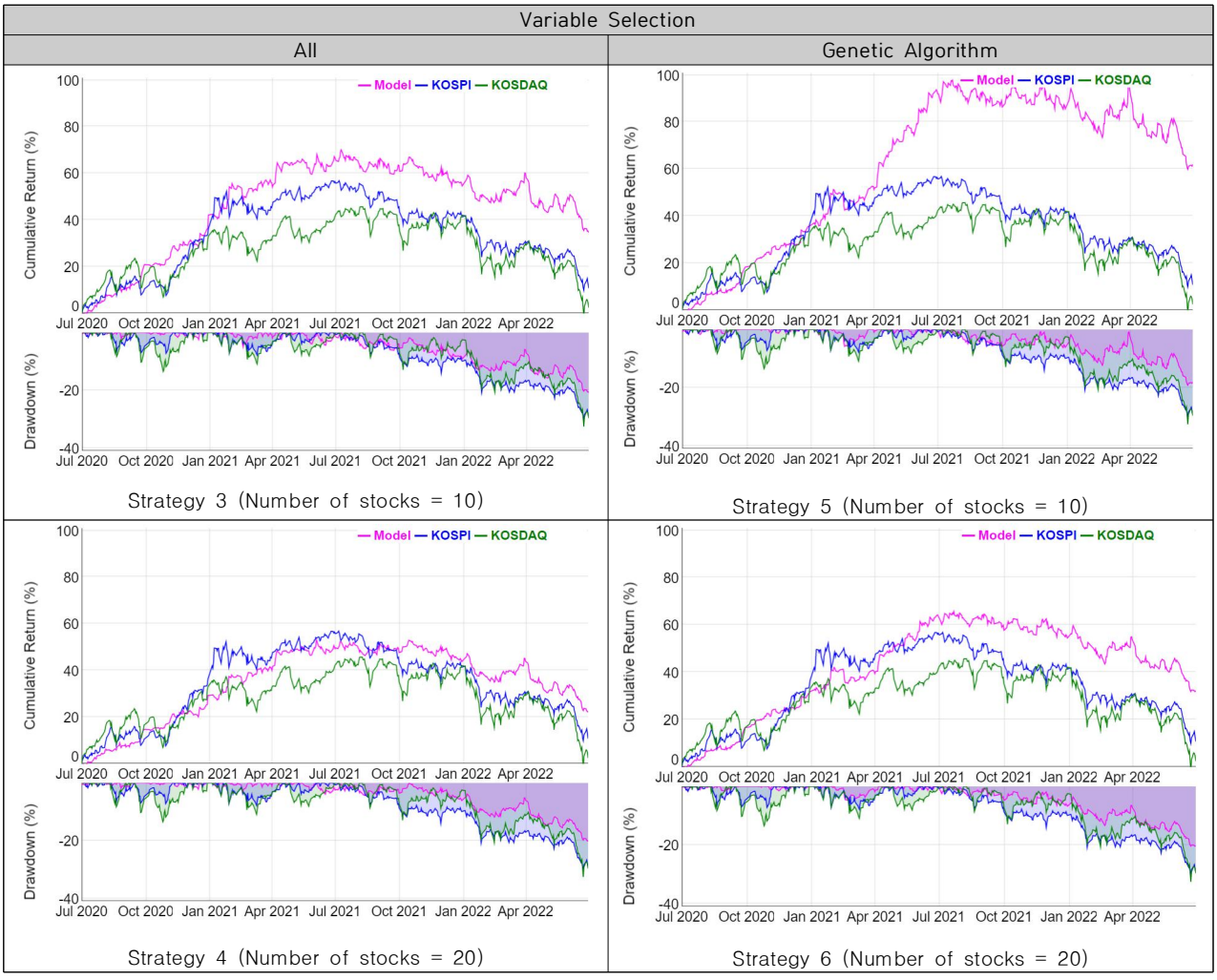

Fig. 3. Cumulative Return and Drawdown Chart of Strategies

MDD의 경우, 전략 5(-19.18%), 전략 4(-20.31%)의 순으로 우수한 것으로 나타났으며, 네 개 전략 모두 KOSPI 지수 및 KOSDAQ 지수의 MDD(-29.98%, -32.61%)보다 양호한 것으로 분석되었다. 변동성은 전략 4(11.31)가 가장 낮으며, 전략 5(14.96)가 네 개 전략 중 가장 높은 것으로 나타났다. KOSPI 지수 및 KOSDAQ 지수의 변동성은 각각 17.46, 22.06으로, 네 개 전략 모두 KOSPI 지수 및 KOSDAQ 지수에 비해 양호한 것으로 분석되었다.

V. Conclusions

본 연구는 앙상블 기계학습 기법 중 그래디언트 부스팅과 유전 알고리즘을 결합, 학습을 통해 KOSPI 및 KOSDAQ 종목들을 선별, 트레이딩 시뮬레이션을 시행하고, 그 성능을 비교 평가하였다. 실험을 위해 증권사 API 등을 활용하여 2013년 7월에서 2022년 6월까지 KOSPI와 KOSDAQ 시장의 거래 종목(1,733~2,664개)의 일별, 분별

가격, 비가격 데이터 및 투자자별 매수, 평균단가 데이터 등을 취합, 활용하였다. 실험 1에서는 다양한 변수들을 생성하여, 그래디언트 부스팅 기법 중 XGboot, LightGBM 및 CatBoost의 성능을 비교 평가하였으며, 분석 결과 LightGBM이 가장 양호한 예측성능을 보이는 것으로 확인되었다. 실험 2에서는 실험 1의 결과를 바탕으로, 유전 알고리즘을 이용한 변수 선택 여부 × 일별 매수 종목 개수에 따른 네 개 전략들의 트레이딩 시뮬레이션을 수행하였으며, 이 전략들은 CAGR, MDD, 샤프지수 및 변동성 측면에서 KOSPI, KOSDAQ 시장 평균보다 우수한 투자성과를 보여주었다. 또한, 유전 알고리즘을 활용한 변수 선별이 투자성과 향상에 효과적임을 확인할 수 있었으며, 이는 유전 알고리즘과 그래디언트 부스팅기법이 효과적으로 결합, 활용될 수 있다는 사례를 보여주었다는 점에서 학문적 의의가 있다.

기존 기계학습을 활용한 알고리즘 트레이딩 관련 연구들에서는 대부분 KOSPI, KOSPI200 등 지수 한두 개 또는 10~20여 개 정도 개별종목들의 일별(일봉), 월별(월봉) 가격 데이터를 활용하여 모델을 생성하였다. 본 연구에서는 일정 수준의 거래량이 수반되는 KOSPI와 KOSDAQ 시장의 모든 종목을 대상으로, 일별, 분별 가격·비가격 데이터 및 투자자별 매수, 평균단가 데이터 등 다양한 데이터를 조합하고 다수의 변수를 생성하였다. 이러한 대규모 데이터를 대상으로 학습, 예측 및 비교평가 통해 생성된 전략들이 시장 평균수익률을 넘어서는 가능성을 보여주었다는 점에서 본 연구의 실무적 의의가 있다고 생각된다.

본 연구의 제한점과 이를 바탕으로 한 후속 연구를 위한 제언은 다음과 같다. 첫째, 본 연구에서는 2013년에서 2022년 사이의 비교적 짧은 기간 내 데이터를 대상으로 분석이 이루어졌다. 이보다 더 긴 기간의 데이터가 취합, 축적되어 이를 대상으로 학습, 예측이 이루어진다면 좀 더 의미 있는 결과를 얻을 수 있을 것이다. 단, 2015년 6월 15일부터 기존 ±15%였던 가격변동 폭이 ±30%로 확대되었는데, 이 시점 전과 후는 주가 변동 양상에 차이를 보일 것으로 판단되며, 과거 데이터 확충, 활용 시 이러한 점을 고려해야 할 것으로 생각된다. 둘째, 본 연구에서는 그래디언트 부스팅 기법들을 비교, 평가하고 유전 알고리즘의 결합이 효과적인가를 확인하는 데 초점을 두어, 비교적 적은 수의 단기 전략만을 대상으로 분석이 이루어졌다. 추후 중장기 투자 및 다양한 전략을 활용한 분석 및 트레이딩 시뮬레이션이 시행된다면 더 실무적으로 의미 있는 결과가 도출되리라 기대된다.

REFERENCES

[1] D. W. Hah, Y. M. Kim, and J. J. Ahn, "A study on KOSPI 200 direction forecasting using XGBoost model," Journal of the Korean Data & Information Science Society, Vol. 33, No, 3, pp. 655-669, May 2019. DOI: 10.7465/jkdi.2019.30.3.655

[2] T. Kimoto, and K. Asakawa, "Stock market prediction system with modular neural networks," Proceedings of the international joint conference on neural networks, pp. 1-6, Jan. 1990. DOI: 10.1109/IJCNN.1990.137535

[3] K. Kamijo., and T. Tanigawa., "Stock price pattern recognition—a recurrent neural network approach," In: Proceedings of the international joint conference on neural networks. pp. 211–5, June 1990. DOI: 10.1109/IJCNN.1990.137572

[4] P. F. Pai, and C. S. Lin, "A hybird ARIMA and support vector machines model in stock price forecasting," Omega, Vol. 33, pp. 497-505, Dec. 2005. DOI: 10.1016/j.omega.2004.07.024

[5] H. J. Kim, and K. S. Shin, "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets," Applied Soft Computing, Vol. 7, pp. 569-576, Mar. 2007. DOI: 10.1016/j.asoc.2006.03.004

[6] Y. C. Yoon, N. R. Jo, and S. D. Lee, "Forecasting algorithm using an improved genetic algorithm based on backpropagation neural network model," Journal of the Korean Data & Information Science Society, Vol. 28, pp. 1327-1336, Nov. 2017. DOI: 10.7465/jkdi.2017.28.6.1327

[7] S. Siami-Namini, N. Tavakoli and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394-1401, 2018. DOI: 10.1109/ICMLA.2018.00227

[8] S. R. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics, pp. 1643-1647, Apr. 2017. DOI: 10.1109/ICACCI.2017.8126078

[9] S. Ö. Arik and P. Tomas, "Tabnet: Attentive interpretable tabular learning," Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 8. May 2021. DOI: 10.48550/arXiv.1908.07442

[10] Y. G. Lee, J. Y. Oh, and G. B. Kim, "Interpretation of Load Forecasting Using Explainable Artificial Intelligence Techniques," The Transactions of the Korean Institute of Electrical Engineers, Vol. 69, No. 3, pp. 480-485, Mar. 2020. DOI: 10.5370/KIEE.2020.69.3.480

[11] H. Kim, "The Prediction of PM2.5 in Seoul through XGBoost Ensemble," Journal of the Korean Data Analysis Society, Vol. 22, No. 4, pp. 1661-1671, Apr. 2020. DOI: 10.7780/kjrs.2021.37.2.11

[12] S. B. Jha, R. F. Babiceanu, V. Pandey, and R. K. Jha, "Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study," arXiv: 2006.10092v1, June 2020. DOI: 10.48550/arXiv.2006.10092

[13] S. I. Jang and K. C. Kwak, "Comparison of Safety Driver Prediction Performance with XGBoost and LightGBM," in Proceeding of Korea Institute of Information Technology Conference, pp. 360-362, Jan. 2019.

[14] J. S. Heo, D. H. Kwon, J. B. Kim, Y. H. Han and C. H. An, "Prediction of cryptocurrency price trend using gradient boosting," KIPS transactions on software and data engineering, Vol. 7, No. 10, pp. 387-396, Oct. 2018. DOI : 10.3745/KTSDE. 2018.7.10.387

[15] D. Opitz, R. Maclin, "Popular ensemble methods: An empirical study," Journal of Artificial Intelligence Research, Vol. 11, pp. 169−198, Aug. 1999. DOI: 10.1613/jair.614

[16] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits and Systems Magazine, Vol. 6, No. 3, pp. 21−45, Sept. 2006. DOI: 10.1109/MCAS.2006.1688199

[17] L. Rokach, "Ensemble-based classifiers," Artificial Intelligence Review, Vol. 33, No. 1, pp. 1−39, Nov. 2010. DOI: 10.1007/s10 462-009-9124-7.

[18] K. M. An, Y. C. Lee, "Corporate Innovation and Business Performance Prediction Using Ensemble Learning," The Journal of Information Systems, Vol. 30, No. 4, pp. 247-275, Dec. 2021. DOI: 10.5859/KAIS.2021.30.4.247

[19] D. S. Siroky, "Navigating Random Forests and Related Advances in Algorithmic Modeling," Statistics Surveys, Vol. 3, pp. 147-163, Nov. 2009. DOI: 10.1214/07-SS033

[20] G. Creamer and Y. Freund, "Automated Trading with Boosting and Expert Weighting," Quantitative Finance, Vol. 10, No. 4, pp. 401-420, Dec. 2010. DOI: 10.1080/14697680903104113

[21] B. Gorman, "A Kaggle Master Explains Gradient Boosting," http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting, Jan. 2017.

[22] S. Basak, S. Kar, S. Saha, L. Khaidem and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers," The North American Journal of Economics and Finance, Vol. 47, pp. 552-567, Jan. 2019. DOI: 10.1016/j.najef.2018.06.013

[23] T. Chen, C. Guestrin, "XGBoost: A scalable tree boosting system," KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, Jun. 2016. DOI: 10.1145/2939672.2939785

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma and T. Y, Liu, "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, Vol. 30, pp. 3146-3154, Dec. 2017.

[25] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Advances in neural information processing systems, 31. 2018.

[26] M. J. Cheon, H. J. Choi, J. W. Park, H. Choi, D. H. Lee and O. Lee, "A Study on the traffic flow prediction through Catboost algorithm," Journal of the Korea Academia-Industrial cooperation Society, Vol. 22, No. 3, pp. 58-64, Mar. 2021. DOI: 10.5762/ KAIS.2021.22.3.58

[27] S. H. Hong, and K. S. Shin, "Using GA based Input Selection Method for Artificial Neural Network Modeling:Application to Bankruptcy Prediction," Journal of Intelligence and Information Systems, Vol. 9, No. 1, pp. 227-247, Jun. 2003. UCI: G704-000721.2003.9.1.010

[28] J. K. Ohk, and K. J. Kim, "Integrated Corporate Bankruptcy Prediction Model Using Genetic Algorithms," Journal of Intelligence and Information Systems, Vol. 15, No. 4, pp. 99-121, Dec. 2009. UCI: G704-000721.2009.15.4.011

[29] M. A. Albadr, S. Tiun, M. Ayob, and F. Al-Dhief, "Genetic algorithm based on natural selection theory for optimization problems. Symmetry," Vol. 12, No 11, pp. 1758-1789, Oct. 2020. DOI: 10.3390/sym12111758.

Authors

Phil-Sik Jang received the B.E. degree in Naval Architecture from Seoul National University in 1990 and received M.S. and Ph.D. degrees in Industrial Engineering from KAIST in 1992 and 1998, respectively.

Dr. Jang joined the faculty of the School of Computer Science at Sehan University, Korea, in 1997. He is currently a Professor in the Dept. of Air Transportation and Logistics, Sehan University. He is interested in HCI, metaverse and bigdata analysis.