

코인 가격 초단기 예측을 위한 딥러닝 모델 비교 연구

A Comparative Study of Deep Learning Models
for Coin Price Short-Term Prediction



팀명 : 자본주의의 노예

연세대학교 글로벌창의융합대학 경영학부 김진우
연세대학교 글로벌창의융합대학 경영학부 안병천

목 차

I. 서 론	3
II. 문헌고찰	3
III. 분석기법 개요	6
IV. 연구모형	
4.1 연구모형	8
4.2 연구절차	8
V. 실증분석 및 결과	
5.1 사용 데이터에 대한 정의 및 요약	10
5.2 사용 데이터의 기초 통계분석	10
5.3 연구모형의 예측, 분류모형의 분석 결과	10
5.4 성과를 비교분석	11
VI. 결론	
6.1 실증결과 요약	13
6.2 연구결과의 공헌도 및 활용방안	13
6.3 연구결과의 한계점 및 향후 개선 방안	13
VII. 참고문헌	13

본 연구는 연세대학교 원주 LINC+ 사업단에서 지원받아 작성되었습니다.

I. 서론

최근 코인(가상화폐)에 대한 범국민적 관심이 높아지고 있으며, 코인의 거래량과 유동성이 증가하고 있다. 이에 기술적 분석을 통해서 코인 시장을 분석하려 한다. 코인과 주식은 공통적으로 차트데이터를 이용하여 기술적 분석이 가능하다는 점을 이용하여 주식 분석 딥러닝 사례를 참고하여, 코인의 가격 동향을 예측하는 모델들에 적용하고 이에 따른 결과를 비교하고자 한다.

주식 가격 예측을 위한 분석 방법은 크게 기본적 분석(fundamental analysis)과 기술적 분석(technical analysis)으로 구분할 수 있다. 기본적 분석은 현재 경제 상태, 전체 산업 상황, 기업의 재무상태 등과 같이 주식 가격에 영향을 미치는 기본 조건 및 결정 요인을 분석하여 미래 주식 가격의 추세를 예측하는 방법이지만, 기술적 분석은 오직 주식의 변동 가격 움직임 자체만을 분석하여 미래의 주식 가격 추세를 예측하는 방법이다(Murphy, 2011).

본 연구에서는 코인 시장 역시 주식시장과 비슷하게, 모든 정보가 현재 코인 가격에 반영되지 못한 비효율적 시장으로 간주하고 기술적 분석을 통해 코인 가격을 예측하고자 한다.

본 연구에서는 코인 시장은 주식시장과 다르게, 재무제표와 같은 정보가 코인 시장에 명확히 존재하지 않아 현재로서는 기본적 분석이 어렵고 이에 따라 코인 시장을 비효율적 시장으로 간주하여 기술적 분석을 통해 코인 가격을 예측하고자 한다.

기존의 기술적 분석은 주식 가격의 변동성 예측을 위해 다양한 차트를 활용하는데 최근 특히 캔들스틱(candlestick) 차트가 많이 활용되고 있다. 캔들스틱 차트는 수치 데이터인 재무 데이터를 투자자들에게 의미 있는 패턴으로 전달함으로써 패턴 인식을 선호하는 투자자에게 강력한 예측 방법을 제공할 뿐 아니라 추세 변화의 시점을 빠르게 파악할 수 있다는 장점도 제공하고 있다. 이에 따라 많은 연구들이 인공지능 기법을 활용하여 캔들스틱 차트를 분석함으로써

주식가격을 예측하고자 시도하였다. 예를 들면, Guo et al.(2018)은 이미지 분류에 특화된 합성곱 신경망(CNN: convolutional neural network)을 사용하여 캔들스틱 차트를 분석함으로써 주식 가격의 등락을 예측하고자 하였으며, Zhipeng and Chao(2019)는 서포트 벡터 머신(SVM: support vector machine)을 활용하여 캔들스틱 차트를 분석함으로써 주식 가격의 등락 예측을 시도하였다(배상현, 채병규, 2021).

하지만 기존 연구들은 주식가격의 시계열적 특성을 반영한 예측을 못하고 있다. 주식 가격의 변화와 같이 고려해야 하는 데이터의 시계열 기간이 길 경우 은닉층의 과거의 정보가 마지막까지 전달되지 못하는 장기 의존성(long term dependencies) 문제가 발생할 수 있다. 이러한 장기 의존성 문제 해결을 위해 장단기 메모리(LSTM: long-short term memory) 방법이 개발되어 활용되고 있으나 이 방법 역시 장기 의존성 문제를 완전히 해결하지 못한다는 한계가 있다(Bao et al., 2017). 장기 의존성은 예측 모델의 학습 성능을 저하시키고 결과적으로 주가 예측의 정확도를 저해한다. 따라서 좋은 성과를 보이는 예측 모델을 만들기 위해서는 장기 의존성 문제를 극복해야 한다. 주가 예측 연구 중 류동기 외.(2017)은 이러한 한계를 극복하기 위해 양방향 장단기메모리(BLSTM, Bidirectional Long Short-Term Memory)를 사용하여 LSTM의 성능을 향상시켰다.

본 연구에서는 데이터의 공간적 특성을 고려하는데 좋은 성과를 보이는 합성곱 신경망(CNN)과 시간적 특성을 반영하는 데 좋은 성과를 보이는 장단기 메모리(LSTM), 양방향 장단기 메모리(BLSTM), 이를 결합한 형태의 분석모델들을 사용한 캔들스틱 차트 기반의 코인 가격 예측 모델들을 비교, 분석하고 어떤 분석모델이 코인 시장 분석에 적합한지 알아보려 한다.

II. 문헌고찰(선행 연구)

최근 빅데이터 및 인공지능 관련 기술이 비약적으로 발전함에 따라 다양한 인공지능 - 특히 딥러닝(deep learning) - 기법에 대한 연구가 활발하게 진행되고 있다. 주식 가격 예측 연구도 이러한 추세에 부

응하여 다양한 인공지능 기법을 활용함으로써 주식이
 가격 예측의 정확도를 높이고자 하는 시도가 이루어지
 고 있다.(배상현, 채병규, 2021)

초기 인공지능 연구는 컴퓨팅 파워 등의 한계로 인
 해 단순 주식시장 변수들을 입력값으로 하는 인공 신
 경망(ANN: artificial neural network)을 주로 활용
 하였다(Niaki and Hoseinzade, 2013).

그러나 기술이 발달함에 비정형 데이터를 활용하기
 위해 지도학습을 바탕으로 이미지 내에 오브젝트를
 빠르게 학습할 수 있는 합성곱 신경망(CNN)을 사용하
 거나 주가의 시계열적 특성을 반영하기 위해 순차적
 인 데이터를 처리할 경우 과거의 연산 값을 전달하는
 순환 신경망(RNN: recurrent neural network)을 사
 용하기 시작한 연구들이 등장하였다(chung and
 Shin, 2020, Ding and Qin, 2020).

또한, 기존 RNN 모델의 그라디언트 소실
 (vanishing gradient) 문제를 해결하기 위해
 LSTM(Long Short Term Memory Networks) 모델
 을 사용하는 연구도 진행되고 있으며 단방향 LSTM
 예측에서 발생하는 오류를 줄이고, 양방향으로 예측하
 는 BLSTM(Bi-direction LSTM)을 사용하는 연구가
 진행되고 있다(류동기 외, 2017, 홍성혁, 2020). 또한
 과거 예측 연구에 사용되었던 모델을 혼합하여 사용
 하는 방법 또한 자주 활용되고 있다(Kim and Kim,

2019, 배상현, 채병규, 2021). <표1>은 딥러닝을 사용
 하여 주식 가격을 예측하는 기존 연구를 정형과 비정
 형으로 나누어 요약한 표이다.

높아지는 코인 시장에 대한 관심과 함께 2017년 이
 후 코인 시장의 가격을 분석하려는 연구들이 등장하
 고 있다. Seo(2018)는 기존 주식 가격 예측에 사용되
 던 딥러닝 모형들(MLP, RNN, LSTM, GRU)을 이용하
 여 비트코인의 가격을 예측하고자 하였으며,
 Heo(2019)는 그라디언트 부스팅을 이용하여 비트코인
 을 포함한 7개의 코인의 10분 후 가격의 상승, 하락
 을 예측하는 연구를 진행하였다. 가장 최근 연구인
 Kim(2021)은 LSTM을 이용하여 비트코인의 일별 가
 격 자체를 예측하고자 하였다.

하지만 이 세 가지 외에 국내에서 코인 시장의 가
 격을 예측하려는 연구를 찾아보기 힘들다. 앞선 세 가
 지 연구는 데이터셋으로 비트코인 혹은 비트코인을
 포함한 코인을 활용하였으며 모두 코인의 가격이라는
 정형적인 데이터를 가지고 가격 자체 또는 가격의 등
 락을 예측하고자 하였다. 아래의 <표2>는 딥러닝을
 사용하여 코인 가격을 예측하는 기존 연구를 요약한
 표이다.

<표 1> 머신러닝 기반 주식가격 예측 기존 연구 요약

구분	저자	데이터셋		분석모델	예측형태	분석기간
정형	Niaki and Hoseinzade(2013)	S&P 500	Technical indexes	ANN	가격상승 또는 하락	1994.3 ~ 2008.6
	Persio and Honchar (2016)	S&P 500	Technical indexes	ANN, WCNN, RNN	가격상승 또는 하락	2006 ~ 2016
	Zhong and Enke (2017)	S&P 500	Technical indexes	ANN, PCA	가격상승 또는 하락	2003.6 ~ 2015.5
	Ryu, Hwang, Choi (2017)	Kospi 100	Technical indexes	BLSTM	가격상승 또는 하락	2009 ~ 2017
	Song(2018)	S&P 500	Technical indexes	RNN, SVM, XGBoost	가격상승 또는 하락	2010.1 ~ 2017.12
	Chung and Shin (2020)	KOSPI index		GA, CNN	가격자체	2000.1 ~ 2016.12.31

정형	Ding and Qin (2020)	Shanghai composite index Petro China ZTE		LSTM	가격자체	NA
	Hong (2020)	NASDAQ index		BLSTM	가격자체	2019.08.28.~ 2020.08..27
비정형	Bao et al.(2017)	S&P 500; Hang Seng index; CSI 300; Nikkei 225; Nifty 50	Candlestick chart, Technical indexes	AE+LSTM	가격자체	2008.7 ~ 2016.9
	Guo et al.(2018)	TAIFEX	Candlestick chart	AE+CNN	가격상승 또는 하락	1998 ~ 2016
	Liu and Song(2018)	10 China Market index stock; 10 White Horse stock	Candlestick chart	SVM, CNN, DNN	가격상승 또는 하락	2006.1 ~ 2017.8
	Shin(2018)	KOSPI		CNN, LSTM	가격상승 또는 하락	NA
	Kim and Kim(2019)	S&P 500	Candlestick chart	CNN, LSTM	가격자체	2016.10 ~ 2017.10
	Kusuma et al.(2019)	S&P BSE SENSEX; NIFTY 50	Candlestick	CNN	가격상승 또는 하락	NA
	Bae and Choi(2021)	S&P 500	Candlestick chart, VIX chart	VAE+ Attention model and Dense layer	가격상승 또는 하락	1993.7 ~ 2019.7

ANN: artificial neural network, WCNN: wavelet CNN, PCA: principal component analysis, XGBoost: extreme gradient boosting, EB: event embedding, GRU: gated recurrent unit, DNN: deep neural network

<표 2> 머신러닝 기반 코인 가격 예측 기존 연구 요약

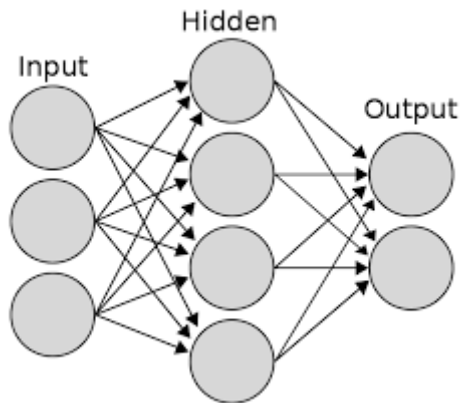
구분	저자	데이터셋	분석모델	예측형태	분석기간
정형	Seo (2018)	BitCoin(일별)	MLP, RNN, LSTM, GRU	가격상승 또는 하락	2013.9 ~ 2017.8
	Heo (2019)	BTC, ETH, XRP, BCH, LTC, DASH, ETC(10분)	그레디언트 부스팅	가격상승 또는 하락	2017.6 ~ 2018.5
	KIM (2021)	BitCoin	LSTM	가격자체	2013 ~ 2021

III. 분석기법 개요

문헌 조사를 통해 선행 연구들이 가격을 예측하기 위해 다양한 딥러닝 모델을 사용하고 있음을 알 수 있다. 본 연구에서는 이러한 다양한 모델들을 도지 코인 가격 데이터에 적용하여 실험한다. 본 연구에서 사용할 딥러닝 모델들은 다음과 같다.

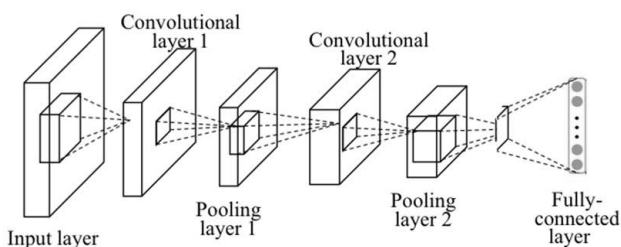
3.1 ANN (Artificial Neural Network)

ANN은 시냅스의 결합으로 네트워크를 형성한 인공 뉴런(노드)이 학습을 통해 시냅스의 결합 세기를 변화시켜, 문제 해결 능력을 가지는 모델 전반을 가리킨다.



3.2.1 CNN (Convolutional Neural Network)

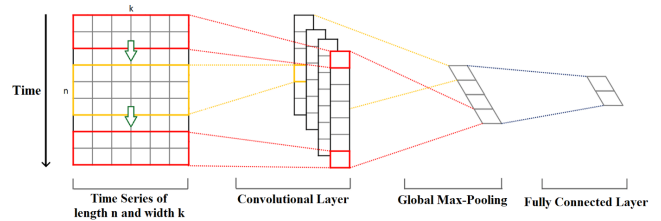
CNN은 이미지 처리에 탁월한 성능을 보이는 신경망이다. CNN은 크게 합성곱층(Convolution layer)과 풀링층(Pooling layer)으로 구성된다.



3.2.2 1D CNN

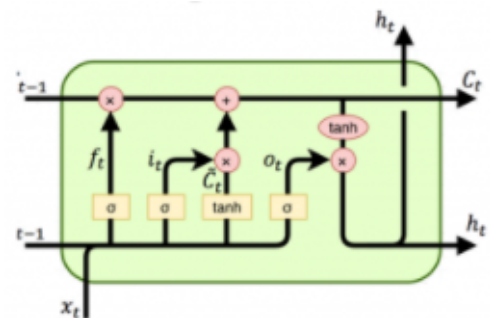
CNN은 주로 이미지 처리에 사용되지만 CNN의 일

종인 1D CNN은 자연어 처리에 특화되어있다. 일방향적인 데이터처리와 배열로 인해 일반적인 CNN과 달리 시계열적 특성이 남아있다.



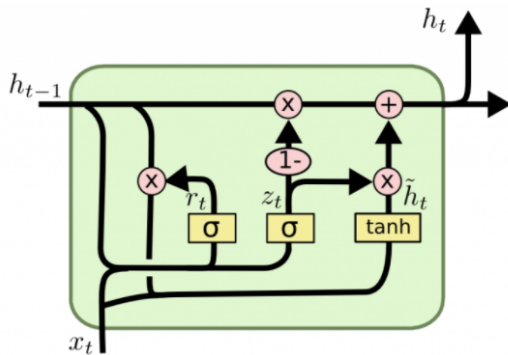
3.3 LSTM (long-short term memory)

LSTM은 RNN(Recurrent Neural Network)에서 파생된 알고리즘이다. RNN은 입력과 출력을 시퀀스 단위로 처리하는 모델의 일종으로 은닉층의 노드에서 활성화 함수를 통해 나온 결과값을 출력층 방향으로 보내면서, 다시 은닉층 노드의 다음 계산의 입력으로 보내는 특징을 갖고 있다. 하지만 RNN의 시점(time step)이 길어질수록 앞의 정보가 뒤로 충분히 전달되지 못하는 현상이 발생한다. 이러한 현상을 해결하기 위해서 은닉층의 메모리 셀에 입력 게이트, 망각 게이트, 출력 게이트를 추가하여 불필요한 기억을 지우고, 기억해야 할 것들을 정한 것이 LSTM이다.



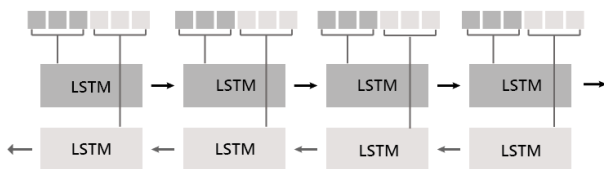
3.4 GRU (Gated Recurrent unit)

LSTM의 구조에서, 불필요한 부분을 제거하는 과정에서 생겨난 아키텍처로 LSTM은 출력, 입력, 삭제 게이트라는 3개의 게이트가 존재했지만, GRU는 업데이트 게이트와 리셋 게이트 두 가지 게이트만이 존재한다.



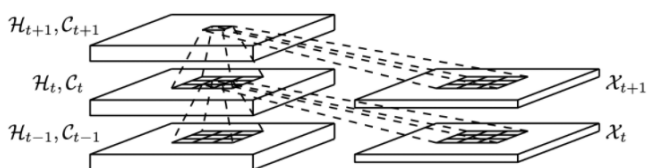
3.5 BLSTM (Bi-directional LSTM)

LSTM은 입력 순서를 시간 순대로 입력하기 때문에 결과물이 직전 패턴을 기반으로 수렴하는 경향을 보인다는 한계가 있다. 이를 극복하기 위해 제안된 알고리즘이 BLSTM이다. BLSTM은 출력값에 대한 손실을 최소화하는 과정에서 모든 파라미터를 동시에 학습되는 종단간 학습 가능하며, 단어와 구(Phrase)간 유사성을 입력벡터에 내재화하여 성능 개선되고, 데이터 길이가 길어도 성능이 저하되지 않는다는 특징이 있다.



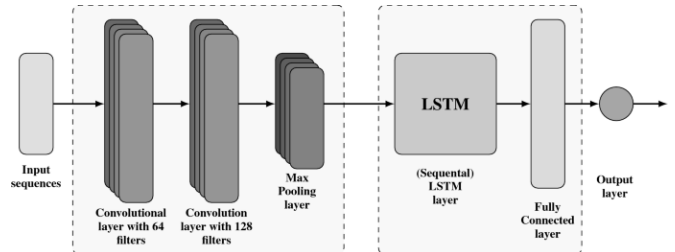
3.6 ConvLSTM (Convolutional LSTM),

ConvLSTM은 기존 LSTM의 행렬 곱이 행해지던 항에서 모든 행렬 곱이 합성곱층으로 대체된 것이다. 입력-망각-출력 게이트와 셀 입력, 셀 출력, 셀 상태는 모두 3차원 텐서이다. 그래서 ConvLSTM은 셀 자체에서 공간적인 의미(CNN의 특성)와 시간적인 의미(LSTM의 특성)를 동시에 나타낼 수 있는 모델이다.



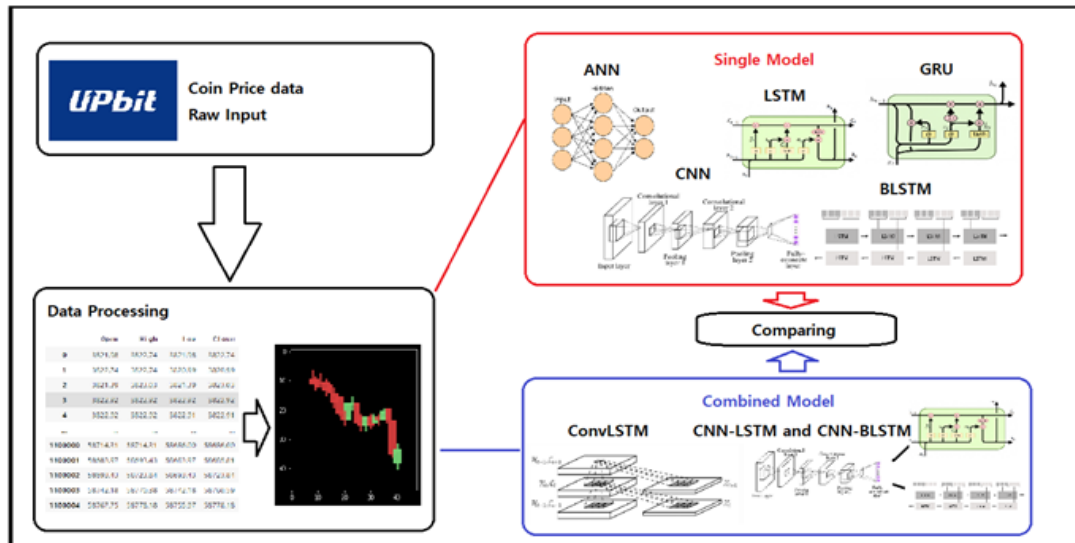
3.7 CNN-LSTM

캔들스틱 차트 이미지가 입력된 CNN 신경망은 이미지의 패턴을 도출해내고, 도출된 데이터는 LSTM 층을 지나 시계열적 특성을 지닌 채 결과를 나타낸다.



3.8 CNN-BLSTM

CNN-LSTM과 비슷한 구조로 LSTM대신 BLSTM을 사용하여 장기 의존성 문제를 비교적 완화시켰다.



<그림 1> 연구모형

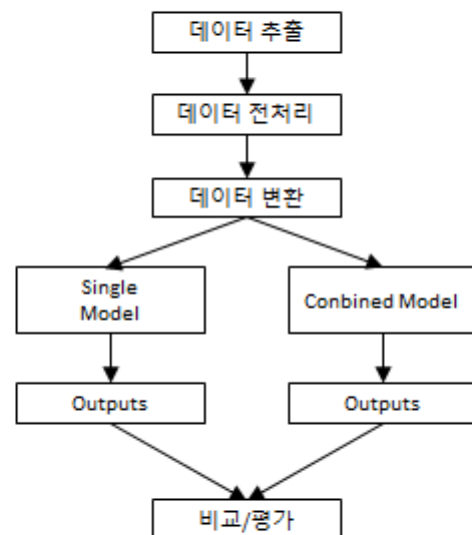
IV. 연구모형 및 분석절차

4.1 연구모형

본 연구를 위해 업비트에서 제공하는 Open API를 이용하여 도지코인의 분별 가격 데이터를 추출하였다. 추출된 가격 데이터를 우선 전처리한 뒤 캔들스틱 이미지를 형성하였고 이를 5가지의 단일 모델과 3가지의 결합 모델의 입력 값으로 입력한 후 학습시켰다. 이렇게 학습된 각 모델을 이용하여 1분 뒤 가격의 증감의 예측값을 구하게 된다. 이 예측값을 실제 값과 비교하여 정확도를 구한 뒤 이를 근거로 모델을 평가, 분석한다. 이러한 연구 방법 및 절차를 요약한 연구 모형은 위의 <그림 1>과 같다.

4.2 연구절차

본 연구는 <그림 2>에 제시된 것처럼 진행되었다. 사용할 가격 데이터를 추출한 뒤, 결측값 제거 및 캔들스틱 차트로의 변환을 위한 전처리를 진행한다. 이후 전처리된 데이터를 라벨링이 된 캔들 스틱 차트로 변환하고 이를 각 모델에 학습시킨 뒤, 최종적으로 비교/평가로 결과를 도출해낸다.



<그림 2> 연구절차

4.2.1 데이터 추출

업비트 Open API를 이용하여 도지 코인의 분봉 데이터를 추출하였다. Open API에서 한번의 요청으로 받을 수 있는 데이터량은 최대 400개이기 때문에 여러번의 요청으로 얻은 값을 결합하여 최종적인 데이터베이스를 생성하였다. KOSPI 종합 지수의 일별 가격 데이터는 FinanceDataReader를 이용하여 추출하였다.

도지코인은 1분 단위로 7일치의 데이터를 수집하여 사용하였으며 KOSPI 종합지수는 일별 단위로 20년, 40년 동안의 데이터를 수집하여 사용하였다.

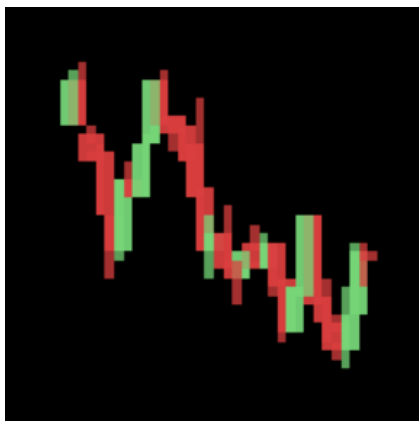
4.2.2 데이터 전처리

추출된 가격 데이터를 캔들 스틱 차트로 변환하기 위한 작업을 진행하였다. 데이터 프레임의 index가 날짜로 되어있는 경우, 캔들스틱 차트로 변환되지 않기 때문에 index를 제거하였다.

4.2.3 데이터 변환

데이터 전처리 결과로 생성된 숫자형태의 가격 데이터는 'mpl_finance' 패키지를 통하여 캔들스틱 차트 이미지로 변환하는데 사용되었다. 세부적인 설명으로 캔들차트 차트 형성을 위해 가격 데이터의 시가, 고가, 저가, 종가 데이터를 사용하였고, 종가가 시가보다 높을 시 '1'의 레이블을, 낮을 시 '0'의 레이블을 부여하였다.

사용한 이미지 크기는 48x48이며 sequence length를 30으로 설정하였기에 한 이미지에 30개의 캔들스틱이 저장되었다. 사용한 데이터의 예는 아래의 <그림 3>으로, 캔들스틱 차트에서 색의 의미는 초록색은 종가가 시가보다 높은 상승을 뜻하며, 빨간색은 종가가 시가보다 낮은 하락을 표현한다.



<그림 3> 캔들스틱 차트 예시

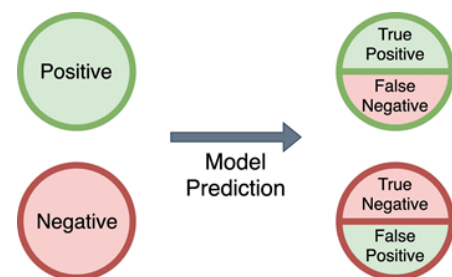
4.2.4 딥러닝 모형

본 연구에서는 딥러닝 모형으로 다양한 단일 모델과 결합 모델을 사용하였다. 단일 모델은 ANN, 1D CNN, CNN, GRU, LSTM, BLSTM을 사용하였고, 결합 모델은 ConvLSTM, CNN-LSTM, CNN-BLSTM을 사용하였다. 보고서를 제출해야 할 시점에서는 각 모델들을 데이터에 최적화하지는 못 하였고 사용데이터가 적기 때문에 대부분의 모형은 하나의 층으로 구성되어 단순화된 모습을 가지고 있다. 사용한 모델의 아키텍처(Architecture)는 아래의 <그림 5>와 같다.

4.2.5 학습 및 성과 측정

딥러닝 모형을 이용해서 이미지를 학습하기 위해 배치 크기(Batch Size)는 16, 학습률(Learning rate)는 0.0001을 사용하였다. 반복 학습 횟수(Epoch)는 16으로 설정하였다.

기존 연구들이 성능 평가로 활용한 값이 정확도이기 때문에 기존 연구를 참고하여 정확도를 사용하였다. 정확도 측정은 다음날 종가의 상승이나 아니냐를 맞추는 binary classification 문제에 맞게 BinaryAccuracy를 이용하였다.



$$\text{정확도} = \frac{TP + TN}{TP + FN + FP + TN}$$

<그림 4> BinaryAccuracy 과정과 정확도 식

<그림 5> 모델별 Architecture

input_7 (InputLayer)	[(None, 48, 48, 3)]	0
flatten_5 (Flatten)	(None, 6912)	0
dense_11 (Dense)	(None, 32)	221216
dense_12 (Dense)	(None, 1)	33
=====		
Total params: 221,249		
Trainable params: 221,249		
Non-trainable params: 0		

(1) ANN

input_13 (InputLayer)	[(None, 48, 48, 3)]	0
conv2d_2 (Conv2D)	(None, 48, 48, 48)	1344
max_pooling2d_2 (MaxPooling2)	(None, 24, 24, 48)	0
dropout_6 (Dropout)	(None, 24, 24, 48)	0
conv2d_3 (Conv2D)	(None, 24, 24, 96)	41568
max_pooling2d_3 (MaxPooling2)	(None, 12, 12, 96)	0
dropout_7 (Dropout)	(None, 12, 12, 96)	0
flatten_8 (Flatten)	(None, 13824)	0
dense_16 (Dense)	(None, 1)	13825
=====		
Total params: 56,737		
Trainable params: 56,737		
Non-trainable params: 0		

(3) CNN

module_wrapper_6 (ModuleWrap)	(16, 2304, 50)	8250
dense_6 (Dense)	(16, 2304, 1)	51
=====		
Total params: 8,301		
Trainable params: 8,301		
Non-trainable params: 0		

(4) GRU

bidirectional (Bidirectional)	(None, 2304, 40)	3840
time_distributed (TimeDistri)	(None, 2304, 1)	41
=====		
Total params: 3,881		
Trainable params: 3,881		
Non-trainable params: 0		

(6) BLSTM

conv_lstm2d (ConvLSTM2D)	(None, 48, 48, 3, 40)	59200
batch_normalization (BatchNo)	(None, 48, 48, 3, 40)	160
conv3d (Conv3D)	(None, 48, 48, 3, 1)	1081
=====		
Total params: 60,441		
Trainable params: 60,361		
Non-trainable params: 80		

(7) ConvLSTM

input_12 (InputLayer)	[(None, 48, 48, 3)]	0
conv1d_3 (Conv1D)	(None, 48, 48, 32)	512
flatten_7 (Flatten)	(None, 73728)	0
dense_14 (Dense)	(None, 16)	1179664
dense_15 (Dense)	(None, 1)	17
=====		
Total params: 1,180,193		
Trainable params: 1,180,193		
Non-trainable params: 0		

(2) 1D CNN

lstm (LSTM)	(None, 2304, 32)	4608
dropout_2 (Dropout)	(None, 2304, 32)	0
lstm_1 (LSTM)	(None, 2304, 32)	8320
dropout_3 (Dropout)	(None, 2304, 32)	0
lstm_2 (LSTM)	(None, 2304, 32)	8320
dropout_4 (Dropout)	(None, 2304, 32)	0
lstm_3 (LSTM)	(None, 2304, 32)	8320
dropout_5 (Dropout)	(None, 2304, 32)	0
dense_7 (Dense)	(None, 2304, 1)	33
=====		
Total params: 29,601		
Trainable params: 29,601		
Non-trainable params: 0		

(5) LSTM

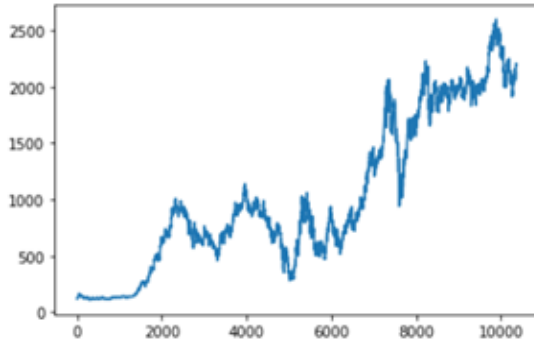
conv1d_1 (Conv1D)	(None, 2304, 32)	512
lstm_4 (LSTM)	(None, 16)	3136
dense_8 (Dense)	(None, 16)	272
dense_9 (Dense)	(None, 1)	17
=====		
Total params: 3,937		
Trainable params: 3,937		
Non-trainable params: 0		

(8) CNN-LSTM

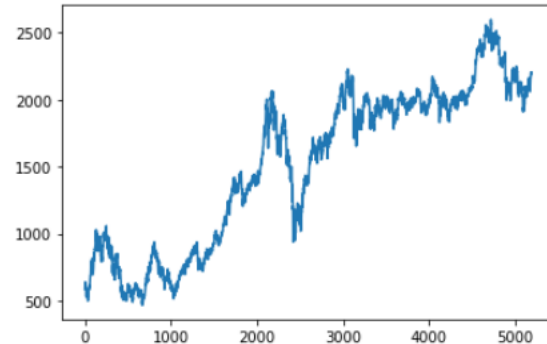
input_12 (InputLayer)	[(None, 2304, 3)]	0
conv1d_7 (Conv1D)	(None, 2304, 32)	512
bidirectional_1 (Bidirection)	(None, 2304, 40)	8480
time_distributed_1 (TimeDist)	(None, 2304, 1)	41
=====		
Total params: 9,033		
Trainable params: 9,033		
Non-trainable params: 0		

(9) CNN-BLSTM

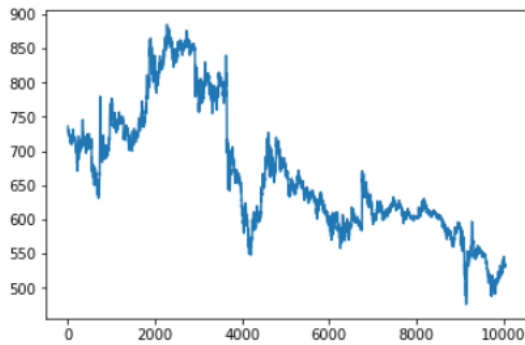
<그림 6> 사용 데이터 요약



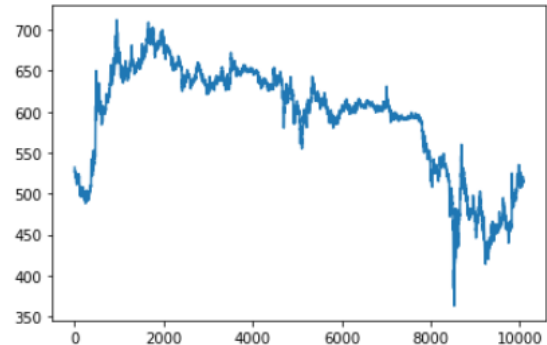
(1) KOSPI(1979-01-01 ~ 2019-12-30)



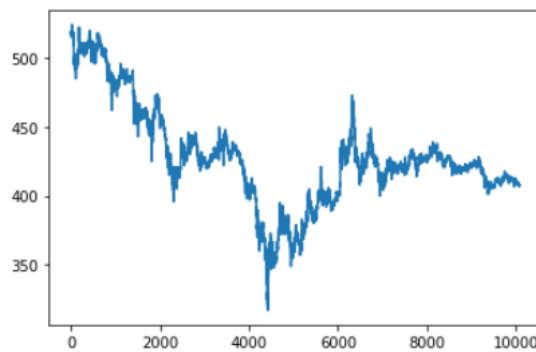
(2) KOSPI(1999-01-01 ~ 2019-12-30)



(3) DOGE(2021-05-07 ~ 2021-05-14)



(4) DOGE(2021-05-14 ~ 2021-05-21)



(5) DOGE(2021-05-21 ~ 2021-05-28)

V. 실증분석 및 결과

5.1 사용 데이터에 대한 정의 및 요약

본 연구에서는 KOSPI의 1999-01-01 ~ 2019-12-30, 20년 동안의 데이터와 1979-01-01 ~ 2019-12-30, 40년 동안의 데이터 중 시가, 고가, 저가, 종가 데이터를 수집하였으며 다양한 형태를 가진 도지코인 데이터에 딥러닝 모델을 접목시켜보기 위해서 다른 세 가지 기간의 일주일동안의 도지코인 분당 가격 데이터를 수집하여 사용하였다.

KOSPI 지수 데이터의 양은 각각 5185개와 10364개이며, 도지코인의 일주일 분당 데이터량은 기본적으로 24 x 60 x 7의 값인 10080개지만, 1차시 기간의 데이터 양은 업비트의 내부 문제로 10041개이다. KOSPI의 40년 데이터를 사용한 이유는 데이터양(10364개)을 도지코인의 데이터양(10080개)과 비슷하게 맞추기 위해서이다.

5.2 사용 데이터의 기초 통계분석

데이터를 딥러닝 모델의 입력값으로 입력하기 전에, 가격 데이터의 종가를 그래프로 그려 데이터가 가진 특성을 비교, 분석하였다. KOSPI 지수 데이터는 우상향하는 추세를 보이며 DOGE 코인의 가격 데이터셋 3가지는 모두 다른 형태를 가지고 있음을 확인할 수

있다. 위의 <그림 6>는 사용한 데이터의 종가를 이용하여 그린 그래프이다.

5.3 연구모형의 예측, 분류모형의 분석 결과

본 연구에서 딥러닝 모델을 통해 KOSPI, 도지코인의 가격을 예측, 분류한 결과는 다음과 같다. 사용한 데이터는 KOSPI 2가지, 도지코인 3가지로 총 5가지이며 실험은 각 데이터셋 당 2번씩 진행하였다.

5.3.1 KOSPI 데이터를 활용한 분석 결과

아래의 <표 3>는 KOSPI 지수를 사용하여 앞서 언급한 9개의 딥러닝 모델들을 실험한 결과를 정리한 표이다. 학습은 Google Colab에서 텐서플로우와 케라스로 실험을 진행하였다.

우선 KOSPI의 40년간 데이터를 사용한 결과와 20년간 데이터를 사용한 결과를 비교했을 때 유의미한 차이가 없었으며, 두 기간에서 나타난 성과의 평균 값으로 비교하였을 때 모든 모델들의 성과 역시 큰 차이를 보이지 않았다. 대부분의 딥러닝 모델들은 50% 근처의 정확도를 보였다. 또한 단일 모델과 결합 모델의 성과 차이도 유의미하게 나타나지는 않았다.

<표 3> KOSPI 데이터 결과

KOSPI								
기간		1979-01-01 ~ 2019-12-20			1999-01-01 ~ 2019-12-20			종합
데이터수		10364			5185			
실험		1차시	2차시	평균	1차시	2차시	평균	평균
단일모델	BLSTM	53.47%	50.86%	52.16%	48.93%	53.22%	51.07%	51.62%
	1DCNN	48.63%	49.56%	49.09%	51.16%	53.54%	52.35%	50.72%
	ANN	50.22%	51.21%	50.71%	48.44%	47.63%	48.03%	49.37%
	CNN	48.34%	48.57%	48.45%	50.41%	46.72%	48.56%	48.51%
	GRU	48.68%	47.69%	48.18%	47.27%	48.30%	47.78%	47.98%
	LSTM	46.46%	49.11%	47.78%	48.88%	46.71%	47.80%	47.79%
결합모델	CNN-BLSTM	46.33%	49.22%	47.77%	50.70%	53.55%	52.13%	49.95%
	ConvLSTM2D	48.35%	49.79%	49.07%	49.66%	49.31%	49.48%	49.28%
	CNN-LSTM	47.72%	50.16%	48.94%	50.64%	46.51%	48.58%	48.76%

<표 4> DOGE COIN 데이터 결과

DOGE COIN											
기간		2021-05-07 ~ 2021-05-13			2021-05-14 ~ 2021-05-20			2021-05-21 ~ 2021-05-27			종합
데이터수		10041			10080			10080			
실험		1차시	2차시	평균	1차시	2차시	평균	1차시	2차시	평균	평균
단일모델	GRU	64.74%	65.51%	65.13%	64.68%	64.91%	64.80%	72.16%	72.26%	72.21%	67.38%
	LSTM	59.77%	59.29%	59.53%	64.65%	64.58%	64.62%	66.58%	66.37%	66.48%	63.54%
	BLSTM	59.51%	40.42%	49.97%	61.92%	56.91%	59.41%	61.80%	66.39%	64.09%	57.82%
	ANN	51.45%	54.93%	53.19%	48.93%	58.46%	53.70%	46.59%	61.21%	53.90%	53.60%
	CNN	41.82%	46.52%	44.17%	43.07%	36.97%	40.02%	34.51%	46.92%	40.71%	41.63%
	1D CNN	55.88%	41.71%	48.79%	38.07%	37.75%	37.91%	40.42%	35.25%	37.84%	41.51%
결합모델	CNN-BLSTM	59.46%	53.52%	56.49%	44.06%	64.23%	54.15%	66.50%	66.68%	66.59%	59.08%
	ConvLSTM2	53.04%	53.68%	53.36%	57.31%	51.47%	54.39%	55.93%	58.86%	57.40%	55.05%
	CNN-LSTM	40.23%	40.31%	40.27%	64.41%	35.46%	49.94%	66.36%	34.13%	50.24%	46.82%

5.3.2 DOGE COIN 데이터를 활용한 분석 결과

위 <표 4>은 도지 코인 가격 데이터를 사용하여 앞서 언급한 9개의 딥러닝 모델들을 실험한 결과를 정리한 표이다. KOSPI 데이터를 활용한 실험과 마찬가지로 Google Colab에서 텐서플로우와 케라스로 실험을 진행하였다.

도지 코인에 딥러닝 모델들을 적용해보았을 때 KOSPI 지수와는 달리, 각 모델들이 나타내는 정확도가 유의미한 차이를 보였다. 우선 데이터셋간의 정확도 차이를 살펴보면 Data1(2021-05-07 ~ 2021-05-14)과 Data2(2021-05-14 ~ 2021-05-21)을 사용한 실험의 정확도는 큰 차이를 보이지 않았지만 Data3(2021-05-21 ~ 2021-05-28)을 사용한 실험의 정확도는 다른 실험의 정확도에 비해 높았다.

다음으로 모델별 비교를 하면 자료의 시계열적 특성을 반영하는 GRU, LSTM, BLSTM의 정확도가 다른 모델에 비해 훨씬 높은 성과를 보였다. 반면 이미지 분석에 특화된 CNN과 1D CNN은 약 40%의 낮은 정확도를 보였다.

결합모델과 단일모델의 정확도를 비교하면 단일모델의 성과가 더 높게 측정되었지만 결합모델의 성과 역시 단일모델과 비슷하게 긍정적인 성과를 나타내었다.

5.3 성과 비교분석

KOSPI 지수 데이터를 입력값으로 사용한 딥러닝 모형(이하 실험1)의 정확도와 DOGE 코인 가격 데이터를 입력값으로 사용한 딥러닝 모형(이하 실험2)의 정확도가 큰 차이를 보이고 있음을 확인할 수 있다.

실험 1의 경우, 평균 49.33%의 정확도를 나타내고 모형 간, 실험 간 차이가 2% 내외로 큰 변화를 보이지 않는다. 또한 딥러닝 모델 중 시계열적 특성을 고려한 BLSTM이 가장 좋은 성과를 보이고 있다.

반면 실험 2의 경우, 정확도의 평균이 54.05%로 실험 1에 비해 월등히 높은 정확도를 나타낸다. 하지만 모델간의 정확도가 큰 차이를 보이며 CNN-LSTM의 경우 1차시에는 35.46%, 2차시에는 64.41%의 정확도를 가지는 등의 실험 간의 차이도 크게 나타난다. 또한, 정확도가 높은 모델을 살펴보게 되면 시계열적 특성을 고려하는 GRU, LSTM, BLSTM이 다른 모델에 비해 매우 높은 성과를 보여준다. 또한 결합모델 같은 경우 실험 간의 차이는 나타나지만 대체적으로 단일모델보다 좋은 성과를 보인다.

VI. 결론

6.1 실증결과 요약

본 연구에서는 기존에 주가 예측에 사용되었던 딥러닝 모형 중 캔들 스틱 기반, 즉 이미지를 입력값으로 사용하는 모델들을 코인 가격 예측에 적용시켜보았다. 코인 가격의 캔들 스틱을 데이터로 딥러닝 모형을 실험한 결과 KOSPI 지수를 예측했을 때의 정확도보다 높은 정확도를 보이고 있음을 확인할 수 있다.

따라서 이러한 실증적인 근거를 통해 기존 주가 예측에 사용되었던 딥러닝 모델들을 코인 가격을 예측하는데 사용하여도 좋은 성과를 만들어낼 수 있다는 결론을 낼 수 있다.

또한, 오히려 주가를 예측했을 때 정확도보다 코인 시장의 가격을 예측했을 때의 정확도가 높은 것을 본다면 딥러닝 모델의 기술적 분석이 코인 가격을 설명하는 데 더 효과적이라고 할 수 있다.

6.2 연구결과의 공헌도 및 활용방안

과거부터 최근까지 꾸준히 높은 주식시장에 대한 관심으로 인해 주식 가격을 예측하려는 시도가 많이 일어나고 있다. 이에 비해 코인 시장이 관심을 받은 지 오랜 시간이 지나지 않았으며 주식에 비해 인지도가 낮았기 때문에 코인 가격을 예측하려는 연구는 주식 시장에 비해 월등히 저조했다. 하지만 2017년 비트코인 이슈와 최근 발생하고 있는 코인 시장에 대한 폭발적인 관심으로 인해 코인 시장의 가격을 예측하려는 연구가 많이 등장할 것으로 보인다.

이러한 연구의 흐름에 맞게 이번 연구에서는 기존의 주식 시장의 예측에 사용되었던 딥러닝 모형들이 코인 가격에도 효과적일 것인가에 대한 근거가 되어 앞으로 등장할 코인 가격 예측 연구의 기반이 될 것으로 기대할 수 있다. 나아가 최근에 발표되었던 코인 관련 연구를 살펴보면 모두 정형적인 데이터를 사용한 연구를 진행하고 있는데, 본 연구에서는 가격 데이터를 이미지 형태의 캔들스틱 차트로 변형하여 비정형 데이터 기반 연구를 진행하였기 때문에 더욱 더 의미가 있다.

6.3 연구결과의 한계점 및 향후 개선 방안

본 연구는 딥러닝 모델들을 코인 가격 데이터에 최적화시키지 못했다는 한계점이 존재한다. 단순한 딥러닝 모형들을 그대로 가져와 코인 가격 데이터 예측에 사용했기 때문에 해당 연구로 보여진 각 모델별 정확도가 충분히 더 향상될 수 있음을 시사한다.

또한, 실험 환경의 한계 때문에 batch를 기반으로 실험을 진행하였는데, 이에 정확도가 실험별로 차이가 많이나는 문제점이 발생하였다. 데이터의 양 역시 기존 연구들에 비해 1/10 정도를 사용하였기 때문에 추후 연구를 통해 batch가 없는 실험을 많은 데이터를 가지고 실험한다면 더 안정적인 결과를 만들어낼 수 있을 것이다. 나아가 현재에는 학습량(epoch)을 10으로 두었는데 만약 시간이 충분하다면 학습량을 기존 연구들과 비슷하게 100 이상으로 놓고 실험을 진행한다면 더 정확한 결과를 도출할 수 있을 것이다.

본 연구는 기존에 존재하는 딥러닝 모델들을 비교하는 것까지를 진행하였다. 이후 후속 연구를 진행한다면 본 연구에서 높은 성과를 보인 모델들을 가지고 데이터에 최적화시켜, 도지코인을 예측하는 것에 맞춤형 모델 개발하는 연구를 진행할 수 있을 것이다.

VII. 참고문헌 목록

- (1) 이재원 (2018). 거래량 급감 패턴의 감독 학습에 기반한 단기 주가 예측. 정보과학회 컴퓨팅의 실제 논문지, 24(10), 544-553
- (2) 류동기, 황석진, 최준수 (2017). 양방향 장단기 기억 네트워크를 이용한 단기 주가예측. 한국정보과학회 학술발표논문집, 992-994
- (3) 배상현, 최병구 (2021) 변이형 오토인코더와 어텐션 메커니즘을 결합한 차트기반 주가 예측, Information Systems Review, 23:1, 23-43
- (4) 홍성주, 임상수(2019). 한국 주식시장에서 시장 베타와 수익률 간의 관계에 대한 실증 분석, 금융안정연구, vol.20 no.2 pp. 175-202 (28 pages)
- (5) 김영준,김여정,이인선,이흥주(2019). 데이터 증강을 통한 딥러닝 기반 주가 패턴 예측 정확도 향상 방안, 한국빅데이터학회지, 제4권 제2호 pp. 1-12

- (6) 신흥기(2019). CNN과 LSTM이 결합된 멀티모달 강화학습 기반의 주식 거래 시스템, 광운대학교 대학원 pp. 1-29 (29 pages)
- (7) 구도현(2021). 시계열이미지 인코딩을 이용한 딥러닝 기반 주식 가격 군집 분석, 서울대학교 대학원 pp. 1-87 (87 pages)
- (8) 이모세, 안현철(2018), “효과적인 입력변수 패턴 학습을 위한 시계열 그래프 기반 합성곱 신경망 모형”, 지능정보연구, 24, pp. 167~181
- (9) 홍성혁 (2020) “BLSTM을 이용한 주가 예측 시스템 연구,” 한국융합학회논문지. 한국융합학회, 11(10), pp. 19-24
- (10) Murphy, J. J., *Intermarket Analysis: Profiting from Global Market Relationships*, John Wiley & Sons, Toronto, 2011.
- (11) Chu, W. and D. Cai, “Stacked similarity-aware autoencoders”, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2017, pp. 1561-1567
- (12) Niaki, S. T. A. and S. Hoseinzade, “Forecasting S&P 500 index using artificial neural networks and design of experiments”, *Journal of Industrial Engineering International*, Vol.9, No.1, 2013, pp.1-9.
- (13) Persio, L. D. and O. Honchar, “Artifiial neural networks architectures for stock price prediction: Comparisons and applications”, *International Journal of Circuits, Systems and Signal Processing*, Vol.10, 2016, pp. 403-413.
- (14) Zhong, X. and D. Enke, “Forecasting daily stock market return using dimensionality reduction”, *Expert Systems with Applications*, Vol.67, 2017, pp. 126-139.
- (15) Kim T, Kim HY (2019) Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data
- (16) Song, Y., *Stock Trend Prediction: Based on Machine Learning Methods*, Master Thesis, UCLA, 2018, Available at <https://escholarship.org/uc/item/0cp1x8th>.
- (17) Chung, H. and K. S. Shin, “Genetic algorithmoptimized multi-channel convolutional neural network for stock market prediction”, *Neural Computing and Applications*, Vol.32, 2020, pp. 7897-7914.
- (18) Ding, G. and L. Qin, “Study on the prediction of stock price based on the associated network model of LSTM”, *International Journal of Machine Learning and Cybernetics*, Vol.11, 2020, pp. 1307-1317.
- (19) Bao, W., J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory”, *PLOS ONE*, Vol.12, No.7, 2017, e0180944.
- (20) Guo, S. J., F. C. Hsu, and C. C. Hung, “Deep candlestick predictor: A framework toward forecasting the price movement from candlestick charts”, *9th International Symposium on Parallel Architectures, Algorithms and Programming(PAAP)*, 2018, pp. 219-226.
- (21) Liu, H. and B. Song, “Stock price trend prediction model based on deep residual network and stock price graph”, *11th International Symposium on Computational Intelligence and Design (ISCID)*, 2018, pp. 328-331.
- (22) Kusuma et al.(2019),“Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market”.
- (23) 서윤범(2018), 비트코인 가격 등락 예측을 위한 딥러닝 모델 연구, 단국대학교 대학원
- (24) 허주성(2019), 기계학습을 활용한 암호화폐 가격 동향 예측, 한국기술교육대학교 대학원
- (25) 김선웅 (2021) 딥러닝을 이용한 비트코인 투자전략의 성과 분석, 한국융합학회논문지, 12:4,

249-258

(26) 한국거래소. <http://www.krx.co.kr>

VIII. Python code 첨부

DOGE 코인 실험

https://colab.research.google.com/drive/1_J3WRPThuV_gya8KEpvtg6oOx7mgsYnM?usp=sharing

KOSPI 실험

https://colab.research.google.com/drive/1Riqlrn1o_w0-RaNsZ0bIpvQ_BtdcM4WM?usp=sharing