

양방향 LSTM과 데이터 조합탐색 및 딥러닝 관련 기법을 활용한 철근 가격 단기예측에 관한 실험적 연구

이용성¹ · 김경환^{2*}

¹건국대학교 일반대학원 건축학과 박사수료 · ²건국대학교 건축학부 교수

Experimental Study on the Short-Term Prediction of Rebar Price using Bidirectional LSTM with Data Combination and Deep Learning Related Techniques

Lee, Yong-Seong¹, Kim, Kyung-Hwan^{2*}

¹Graduate Student, Department of Architecture, Konkuk University

²Professor, Department of Architecture, Konkuk University

Abstract : This study presents a systematic procedure for developing a short-term prediction deep learning model of rebar price using bidirectional LSTM, Random Search, data combination, Dropout. In general, users intuitively determine these values, making it time-consuming and repetitive attempts to explore results with good predictive performance, and the results found by these attempts cannot be guaranteed to be excellent. With the proposed approach presented in this study, the average accuracy of short-term price forecasts is approximately 98.32%. In addition, this approach could be used as basic data to produce good predictive results in a study that predicts prices with time series data based on statistics, including building materials other than rebars.

Keywords : Rebar Price, Bidirectional LSTM, Data Combination, Hyperparameter Random Search, Price Prediction, Dropout

1. 서론

1.1 연구의 배경 및 목적

건축재료비는 자재를 실제 사용하기 이전 물량에 자재 단가를 적용하여 비용을 계획한다. 이때 자재의 단가는 계획 당시의 시세를 반영하여 결정하기 때문에 자재를 실제 사용하기 위해 구매할 때 시세의 변동으로 예상하던 가격과 차이가 나는 경우가 발생한다(Choi et al., 2009). 건설사업관리 측면으로 보면 이러한 차이는 매월 수행된 작업에 대한 예산보다 실투입비가 낮거나 높아질 수 있는 상황을 가정해 볼 수 있는데, 재료비의 상승으로 실투입비가 높아지게 되면 현시점에서 예상되는 공사종료 시점의 최종 비용상승으로 직결된다. 특히 건축재료 중 구조체 제작에 필수적으로 사용되는 철근의 경우 타 건축재료들과 비교해 상대적으로 가격

변동 폭이 크고, 사용량이 많아서 채산성을 고려해야 하는 건설업체에 철근의 가격변동은 상당한 부담으로 작용할 수 있다. 또한 과거 2008년 철근 가격 폭등으로 인해 여러 건설업체의 공사가 중단되는 사태까지 발생해 시세변동에 대한 대비가 항상 필요하다는 것에 모두 공감할 것이다(Choi & Kwon, 2008). 이러한 문제를 해결하기 위해 기존의 인간에 직관과 경험으로 이뤄지는 예산에 대한 전통적 예측방식을 대체할 새로운 방법이 필요하다.

한편 기술의 발전으로 인공지능 기술이 인간의 판단과 속도, 정확도를 뛰어넘는 결과를 일부의 영역에서 얻고 있다. 인공지능의 기술 중 하나인 딥러닝은 고도의 수학적 지식이나 프로그래밍 능력을 요구하지 않기 때문에 이 기술을 적극적으로 기업 내 자산에 활용하거나 융합하려는 시도가 활발해지고 있다. 또한 Lasso (Least Absolute Shrinkage and Selection Operator), 서포트벡터머신(Support Vector Machine; SVM)등과 같은 전통적인 회귀분석 모델보다 예측능력이 우수함을 입증하는 연구들이 제시되고 있어 딥러닝 모델에 대한 관심이 높아지고 있으며(Yoon. J., 2019), 범용성 또한 뛰어난 번역, 음성인식, 자율주행차, 로봇틱스 등

* **Corresponding author:** Kim, Kyung-Hwan, Department of Architecture, Konkuk University, Seoul 05029, Korea
E-mail: kykim@konkuk.ac.kr
Received August 24, 2020: **revised** September 28, 2020
accepted October 19, 2020

다양한 분야에서 활용되고 있다(Jeong, 2017). 그리고 최근 일정한 시간 간격을 기준으로 관측치를 기록한 데이터를 딥러닝을 이용해 컴퓨터에 학습시켜 입력된 데이터의 다음을 예측하는 연구가 활발히 진행되고 있다. 특히 이 방식을 통해 국제유가, 주택가격, 주식 등과 같이 변동성이 큰 가격을 단기예측하여 결과로 제시하는 기존의 연구가 있어(Lahari et al., 2018; Chen et al., 2017; Pawar et al., 2019), 철근 가격예측을 위한 방법으로써 활용 가능할 것으로 판단된다. 또한 이 방식의 활용은 철근 가격으로 인해 달라질 수 있는 기존의 매월 수행된 작업에 대한 예산을 추정하는 방식보다 정확한 추정이 가능할 것으로 사료되며, 기존의 경험과 직관을 기반으로 한 전통적인 방법의 예측 체계성을 보완할 새로운 대안이 될 수 있다.

이에 본 연구에서는 딥러닝을 활용하여 철근 가격을 단기 예측하는 방법을 제안하고자 한다.

1.2 연구의 방법

연구의 방법으로는 기존의 연구에서 딥러닝을 적용하는 방법과 사용한 모델, 예측성능을 높이기 위한 딥러닝 관련 기법에 대한 문헌 고찰을 실시한다. 이후 본 연구에서 사용할 딥러닝에 관련된 기법들에 대한 이론적 고찰을 실시한다. 그리고 이를 토대로 딥러닝 모델을 구현하고, 철근 가격을 예측하기 위한 데이터를 선정하여 모델에 적용 후, Test 예측결과와 데이터의 범위를 벗어난 값을 예측하여 실제값과 비교하여 예측성능을 평가한다.

2. 딥러닝을 활용한 기존의 연구 고찰

딥러닝을 활용하여 값을 예측하는 기존의 연구에서는 특정 값을 예측하기 위해 예측대상과 관련 있는 데이터와 이를 순환신경망 계열의 딥러닝 모델에 학습시켜 예측결과를 제시하는 연구가 주를 이루고 있다. 또한 순환신경망 계열의 여러 딥러닝 모델에 동일한 데이터를 학습시켜 모델의 예측성능을 비교한 연구도 찾아볼 수 있었다. 이에 대한 기존의 연구로는 Lahari et al. (2018)는 국제유가인 Brent crude oil, 환율의 시계열 데이터와 딥러닝 모델인 Recurrent Neural Network (RNN)로 기름 가격을 예측하였고, Cen et al. (2019)는 West Texas Intermediate (WTI)와 Brent crude oil의 일일 가격, 종가, 개시가의 시계열 데이터와 딥러닝 모델인 Long Short Term Memory (LSTM)로 원유가격을 예측하였다. 그리고 Chen et al., (2017)은 Beijing, Shanghai, Guangzhou, and Shenzhen의 주택가격 시계열 데이터와 딥러닝 모델인 LSTM, Stack LSTM으로 주택가격을 예측하였고, Joo et al. (2018)과 Pawar et al. (2019)은 시가, 고

가, 저가, 거래량 등의 주식 시계열 데이터에 딥러닝 모델인 RNN, LSTM and Bidirectional Recurrent Neural Network (BRNN)으로 주가를 예측하여 모델별 예측성능을 비교 제시하였다.

그리고 여러 딥러닝 관련 기법들을 활용하여 예측정확도를 높이기 위한 연구도 진행된 바 있다. 이에 대한 기존의 연구로는 Srivastava et al. (2014)은 모델 신경망에 뉴런들을 확률적으로 사용하지 않게 하여, 예측성능을 향상시키는 방법인 Dropout을 제안하였고, Zhang et al. (2018)과 Jang et al. (2020)은 모델의 예측성능을 최대로 끌어올리기 위해 사용자가 직접 설정해줘야 하는 값인 하이퍼 파라미터를 자동으로 교차 검증하는 Grid Search를 이용하여 탐색된 결과를 제시한 연구가 있었다.

그리고 우수한 예측성능을 나타내는 데이터를 선정하는 방법으로는 딥러닝 모델에 사용할 후보 데이터를 선정하고, 이를 몇 가지 Case로 조합하여 가장 우수한 예측성능을 나타낸 데이터셋과 예측결과를 제시한 연구가 있었다(Jang et al., 2020; Lee, 2019; Mou, 2019).

따라서 기존의 연구에서 제시하는 데이터셋을 선별하는 방법과 Grid Search, Dropout을 순환신경망 계열의 딥러닝 모델에 적용하여 알고리즘을 구현하면 철근 가격의 단기 예측이 가능할 것으로 사료된다. 하지만 이 방법들을 기존의 연구에서 제시한 그대로 반영하여 사용하는 것은 예측대상과 이 대상을 예측하기 위해 사용되는 데이터, 딥러닝 모델, 하이퍼 파라미터, Dropout 적용방법 등이 모두 다르기 때문에 우수한 예측결과를 얻을 수 있다고 확신하긴 어렵다. 또한 이 방법들을 포괄적으로 조합하여 모든 경우를 고려하는 것이 우수한 예측결과를 탐색하기 위한 가장 좋은 방법일 수 있으나, 위에 언급된 방법들을 모두 반영하여 진행하면 고려대상 수의 곱만큼 Grid Search의 탐색 범위와 결과를 탐색하기 위한 시간이 급격히 늘어나기 때문에 이를 효율적으로 줄일 방법이 필요하다.

3. 딥러닝에 관한 이론적 고찰

3.1 지도학습

지도학습(Supervised Learning)은 명시적인 정답이 주어진 상태에서 컴퓨터를 학습시키는 방법으로, 데이터의 특성을 토대로 값을 예측한다(Olson, 2008). 본 연구에서는 기존의 가격예측 연구에서 많이 사용된 Supervised Learning의 한 종류인 RNN, LSTM, BRNN 대한 이론적 고찰을 실시하였다.

3.1.1 RNN

순환신경망은 은닉계층에 이전 정보를 기억시킬 수 있는

방향을 가지고 순환구조를 이루는 인공신경망의 한 종류이다. 음성, 문자 등 순차적인 데이터 처리에 적합하며, 이전 출력 결과의 영향을 받아 현재까지 계산된 결과를 메모리에 기억한다는 장점이 있다(Schuster & Paliwal, 1997). 학습은 입력을 비선형 관계로 변환하여 다음 단계의 입력으로 전달되며, 출력값은 $-1 \sim 1$ 사이의 작은 값을 나타낸다. 그리고 출력값이 곱해지는 단계가 반복되면 이 값이 소실되어 학습이 더 이상 이루어지지 않는 기울기 소실(Vanishing gradient)가 발생한다(Gao et al., 2019). 이는 과거의 학습결과가 사라지는 장기 의존성(Long-Term dependencies) 문제를 발생시킨다(Trinh et al., 2018).

3.1.2 BRNN

RNN은 입력 순서를 시간 순서대로 처리하기 때문에 예측결과가 직전 패턴을 기반으로 하는 경향을 보이는 한계가 있다. 이러한 한계를 해결하기 위해 BRNN이 제안되었다. BRNN은 서로 연결되지 않는 두 개의 은닉층을 가지 있으며, 순방향과 역방향으로 분리된 순환신경망을 통해 학습시키기 때문에 예측정확도를 향상시킨다. 입력값은 두 개의 은닉계층으로 전달되고, 출력계층에서는 두 개의 은닉계층의 출력이 하나로 합쳐져 입력 받은 후 가중치 연산과 최종출력을 한다(Schuster & Paliwal, 1997).

3.1.3 LSTM

LSTM은 RNN의 기울기가 사라지는 Long-Term dependencies 문제를 극복하기 위해 제안된 순환신경망 구조이다. 셀 상태를 유지하면서 Input, Output, Forget 게이트라 불리는 구조에 의해 제어된다. 이를 통해 기울기 소실을 방지하고 효과적으로 흐를 수 있게 하며, 시계열 데이터와 같은 긴 순서의 데이터도 기울기 소실 문제없이 처리하게 한다(Hochreiter & Schmidhuber, 1997).

3.2 시계열 데이터

시계열 데이터는 시간에 따라 변경된 값으로 구성된다. 즉, 관측치가 시간적 순서를 가지는 데이터를 말한다. 그리고 과학, 엔지니어링, 비즈니스, 금융, 경제, 의료 등 다양한 분야에서 광범위하게 사용되고 있다(Liao, 2005). 또한 과거에 관측된 값을 통해서 현재와 미래를 예측하는데 사용된다.

3.3 데이터 정규화

데이터 간의 편차를 줄이기 위해 모든 입력데이터는 normalization를 진행하여 $0 \sim 1$ 사이의 값으로 조정하여 적용한다. Normalization에 사용된 방법은 MinMaxscaler로 식(1)과 같다.

$$X = (x - x_{\min}) / (x_{\max} - x_{\min}) \quad (1)$$

x : 데이터
 x_{\min} : 데이터의 최소값
 x_{\max} : 데이터의 최대값
 X : 스케일변환데이터

3.4 하이퍼 파라미터 최적화

하이퍼 파라미터는 딥러닝 모델의 입력데이터의 크기나 Cell 내부의 은닉층 수, 학습량 등과 같은 값들을 지칭한다. 또한 이 값들은 사용자가 직접 조절하여 설정해야 딥러닝 모델의 예측성능을 최대로 끌어올릴 수 있다(Claesen & De Moor, 2015). 이러한 값들을 탐색하기 위해 시도할 하이퍼 파라미터 후보 입력값을 선정하고, 이를 사용하여 예측결과를 몇 차례 측정한 후 가장 높은 성능을 나타낸 하이퍼 파라미터 입력값을 채택하는 방법을 Manual Search라 한다(Larochelle et al., 2017). 그리고 시도할 하이퍼 파라미터 후보 입력값 후보 값을 선정하고, 이를 조합하여 자동으로 탐색하는 과정을 Grid Search라 하고, 시도할 하이퍼 파라미터 후보 입력값을 범위로 선정하고 Random으로 선택 조합하여 자동으로 탐색하는 과정을 Random Search이라 한다(Bergstra & Bengio, 2012). 특히 Random Search는 입력된 값의 모든 경우의 수를 탐색하는 Grid Search보다 입력된 값의 범위 내에서 생성된 난수 조합으로 탐색하기 때문에 탐색의 범위가 넓을 경우 Grid Search보다 더 잘 탐색한다고 한다.

3.5 과대적합, 과소적합

과대적합(Overfitting)이란 딥러닝 모델에서 데이터를 과하게 학습하는 것을 말한다. 이는 학습 데이터가 부족하거나, 데이터의 특성에 비해 모델이 너무 복잡한 경우 발생한다. 그리고 이 문제의 결과는 학습성능은 우수하지만, Test의 예측성능을 저하시킨다. 이를 해결하기 위해서는 학습에 사용할 데이터를 늘리거나 모델의 복잡도를 낮추는 기법, 데이터의 이상치를 제거하는 방법, Dropout을 적용하는 방법 등이 있다.

과소적합(Underfitting)은 모델이 너무 단순할 경우 데이터를 제대로 학습하지 못할 때 발생하는 문제를 지칭한다. 이는 학습성능에 비해 Test의 예측성능이 크게 떨어지는 경우를 예로 들 수 있다. 이를 해결하기 위해서는 파라미터가 더 많은 모델을 선택하는 방법과 모델의 하이퍼 파라미터 입력값을 낮추는 방법, 과대적합 되기 전의 시점까지 충분히 학습시키는 방법 등이 있다.

3.6 Dropout

Srivastava et al. (2014)는 과적합 문제를 해결하기 위해 Dropout을 제안하였다. 이 방법은 신경망에 뉴런들을 확률적으로 사용하지 않음으로써 과적합을 방지하는 기술이다. 그리고 이는 모델의 일반화 성능을 향상시킨다. Dropout의 사용방법은 0.1 ~ 0.9까지의 입력 비율에 따라 신경망에 뉴런들을 확률적으로 사용하지 않는다. 입력 비율 1.0은 Dropout을 사용하지 않는 디폴트 값을 의미한다.

3.7 모델 성능평가

딥러닝 모델의 예측성능을 측정하기 위해 평균 제곱근 오차인 Root Mean Square Error (RMSE)를 이용하였다. RMSE는 모델의 예측값과 실제값의 차이를 다룰 때 사용하는 측도로 식(2)과 같다.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (R_t - P_t)^2} \quad (2)$$

R_t : 실제값
 P_t : 예측값
 n : 데이터의 개수

R_t 는 실제 값이고, P_t 는 예측값을 나타낸다. 이 두 값의 차가 음수일 경우를 고려하여 제공하고, 예측된 결과 n 만큼 나누어 평균 계산 후, 루트를 사용하여 RMSE를 산출한다. RMSE는 값이 낮을수록 모델의 예측성능이 높다는 것을 의미한다.

4. 철근 가격예측을 위한 데이터 정보

본 연구에서는 한국의 철근 가격예측을 위해 철근 가격(보통철근 일반 SD300, D10mm, 0.560), 철근 수입물가지수, 철근 생산량, 고철 수입량, 국제유가 3종(\$/Ton), 생산자물가지수(일반 철근), 고철 단가(\$/Ton), 건축 착공현황과 같은 철근 생산과 수요에 직·간접적 연관성이 있는 한국의 시

Table 1. Variable and Independent Variable

ID	Name	Source
A	import price index(rebar and bar steel)	Korea National Statistical Office
B	the producer price index(rebar)	e-National Indicators
C	Dubai crude oil	e-National Indicators
D	Brent oil	e-National Indicators
E	West Texas Intermediate	e-National Indicators
F	scrap iron(Unit price)	e-National Indicators
G	scrap iron(the amount of imports)	e-National Indicators
H	rebar production volumes	Korea National Statistical Office
I	construction statistics	Korea National Statistical Office
J	Rebar price 0.560 SD300, general	Korea Price Information

계열 통계자료를 기반으로 데이터셋을 구성한다. 그리고 환율은 단위가 \$/Ton로 표시되는 데이터에 곱하여 원화 단위를 적용하였다.

〈Table 1〉은 데이터의 ID와 출처를 보여준다. 이 데이터 집합은 2003년 1월부터 2020년 3월까지 한 달 간격으로 CSV (Comma-Separated Value) 파일로 정리하였다. 전체 데이터는 MinMaxScaler를 사용하여 정규화하였다.

5. 딥러닝 모델 구축 및 시스템 환경정보

본 연구에서는 LSTM과 BRNN을 결합한 양방향 LSTM (Bi-LSTM)을 적용한다. 앞에서 설명한 것처럼 LSTM과 BRNN은 개선된 RNN으로 LSTM은 기울기 소실 문제를 극복하며, BRNN은 순방향과, 역방향의 은닉층 연산을 반영하여 예측성능을 향상시킨다. 예측값은 시퀀스 길이로 읽은 데이터의 다음 값에 해당하는 철근 가격이다. 즉, 매 시점마다 1개월 후의 철근 가격을 예측하도록 설정되었다. 그리고 이 모델의 주요 하이퍼 파라미터는 〈Table 2〉와 같다. 그 외에 시스템 구성에 대한 사양은 〈Table 3〉과 같다.

Table 2. Hyper parameter and Description

Hyperparameter name	Description
Sequence length (Seq.)	Row length of input data
Learning rate (Lrate)	Gradient descent learning speed
Hidden layer (Hid.)	Hidden layer inside the cell
Iterations (Iters.)	Iterative learning
keep_prob(DO)	Dropout rate

Table 3. System configuration and specifications

Divide	Program information
OS	Window 10
Programming Tool	Visual Studio Code, Jupyter
virtual environment	anaconda 4.7.12
General-Purpose computing on Graphics Processing Units, GPU	CUDA 10.0
Language	Python 3.6.5
Deep Learning Library	Tensorflow 1.15.0 - GPU
Algorithm	Bidirectional Dynamic RNN, LSTM

또한 하이퍼 파라미터 탐색기와 결과를 시각화하기 위한 Python 패키지 모듈은 〈Table 4〉와 같다. Dropout 레이어는 Cell의 다음에 위치시켜 Cell의 출력부에만 적용되게 설정하였다. 그리고 하이퍼 파라미터 입력값과 데이터를 전달받을 수 있도록 모델을 함수로 정의하고, 데이터를 조합한 집합에서 무작위로 데이터셋을 선택하여 선택된 데이터셋이 전체 기간의 70%를 학습에 나머지 30%를 테스트에 사용하도록 자동분할 되게 설정하였다. 마지막으로 정의된 모델의 하이

퍼 파라미터를 탐색할 수 있도록 'Random Search'와 연결하였다.

Table 4. Package information

Package	Package information
matplotlib 3.1.1	Results Visualization
numpy 1.16.5	Array
sklearn 0.23	RandomizedSearchCV

6. 실험 방법

철근 가격을 예측하기 위해 본 연구에서는 선정한 10개의 데이터를 조합하여 Random Search를 이용해 각각의 데이터 조합에서 무작위로 선택된 데이터와 하이퍼 파라미터 최적화를 동시에 진행한다. 데이터 조합에서 종속변수인 'J'는 예측대상 데이터로 모든 데이터셋에 포함되기 때문에 데이터 조합 시 이는 제외하였다. 실험에 사용되는 데이터의 조합 수는 <Table 5>와 같다.

Table 5. Number of data combinations

combination	9C2	9C3	9C4	9C5	9C6	9C7	9C8	9C9
Number of cases	36	84	124	126	84	36	9	1

탐색 과정은 Random Search에 의해 각각의 조합에서 데이터를 무작위로 선택하고, 선택된 데이터에 대한 하이퍼 파라미터도 무작위로 조합하여 탐색을 수행한다. 또한 이 각각의 과정에는 Test 구간의 실제값과 예측값에 의한 RMSE가 산출되고, 데이터셋의 가장 마지막 부분을 시퀀스 길이 만큼 읽어와 학습된 모델에 입력데이터로 사용하여 데이터셋의 범위를 벗어난 이후의 1달 가격, 즉 2020년 4월을 예측하여 출력한다. 이는 최적화된 모델의 실제 예측성능을 확인하기 수행된다. 그리고 철근 시세가 실제 2020년 1월부터 4월까지 610,000원으로 가격의 변화가 없는 상태이기 때문에 가장 마지막 데이터의 철근 가격이 오르는 추세인 2003년 1월부터 2019년 11월까지, 하락하는 추세인 2003년 1월부터 2017년 7월까지 2가지를 범위로 한 데이터셋을 기존의 데이터셋에서 추가 생성해 앞 전의 과정과 같은 방식으로 예측 결과를 출력 및 평가하여 입력된 데이터에 대한 등락을 올바르게 예측하는지 확인한다. Random Search의 하이퍼 파라미터 탐색 범위는 학습률 0.1 ~ 0.001, 시퀀스 길이 2 ~ 40, 은닉층 수 10 ~ 500, 학습량 100 ~ 7000, Dropout 비율 0.1 ~ 0.99의 최소, 최대값 범위 내에서 무작위로 선택되고, 각각의 조합별 100번씩 총 800번의 이 과정을 반복한다. 그리고 각각의 조합에서 탐색한 100개의 결과는 다음 식(3)에 의해 우

선 선별한다.

$$P = 1 - (TrainR - TestR) \tag{3}$$

TrainR : 학습결과 RMSE

TestR : Test결과 RMSE

P : 과대적합 선별, $0.999 \leq P \leq 1$

식(3)은 학습 RMSE와 Test RMSE의 차가 1과 0.999 이 내의 점점 결과를 선별하기 위해 사용한 식으로, 이에 대한 설명을 위해 <Fig. 1>을 추가하였다. <Fig. 1>의 결과는 Dropout의 적용 전후의 학습량 1부터 1000까지 1씩 증가할 때마다 Train RMSE와 Test RMSE를 측정하여 측도의 수치 변화를 기록한 결과이다. Dropout을 사용하지 않으면 <Fig. 1>의 (A)와 이를 부분적으로 확대한 (C)와 같이 일반적으로 Test RMSE가 Train RMSE 이상으로 감소하지 못한다. 하지만 Dropout을 사용할 경우 <Fig. 1>의 (B)와 이를 확대한 (D)와 같이 학습량이 증가할수록 과동의 형태로 서서히 감소하며, Train RMSE 이상으로 Test RMSE가 감소하는 것을 확인할 수 있다. 이 때문에 Test RMSE는 Dropout의 적용 전보다 감소할 수 있다. 하지만 학습성능과 테스트성능의 차이가 크면 과대적합에 의한 모델이 일반화(Generalization)되지 못한 상태이기 때문에 데이터의 범위를 벗어난 값을 예측할 경우 원하는 결과와 다른 값을 예측할 가능성이 있다. 그래서 본 연구에서 사용하는 하이퍼 파라미터 탐색방법인 Random Search는 Dropout의 입력값을 동시에 탐색하기 때문에 탐색결과 내에서 학습성능과 테스트성능의 차가 작은 결과를 식별할 수 있도록 식(3)을 추가하였다. 이후 이를 통해 선별된 결과는 Test RMSE가 낮은 순으로 단계적으로 선별하여 최종선택한다. 마지막으로 최종선택된 결과에 기록된 데이터의 범위를 벗어난 이후의 1달 가격을 실제값과 비교하여 모델의 예측성능을 평가한다.

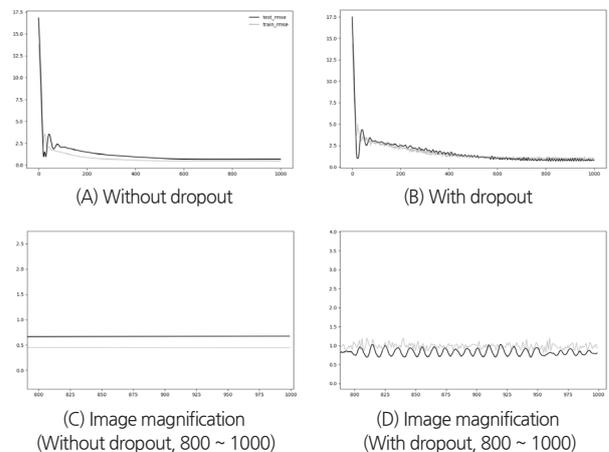


Fig. 1. Example of Visualization Comparison

7. 실험 결과

철근 가격예측을 위해 사용한 10개의 데이터를 조합하여 Random Search로 탐색한 각각의 데이터 조합결과에서 *P*값과 Test RMSE 순으로 선별한 실험 결과는 <Table 6>과 같다. 그리고 이 중 데이터의 가장 마지막 철근 가격이 오르는 추세인 2003년 1월부터 2017년 7월까지의 데이터셋을 사용해 진행한 실험 결과에서 가장 우수한 예측성능을 나타낸 9C4 데이터 B, F, G, I에 2013년 4월 ~ 2017년 7월까지의

Test RMSE는 0.02375로 이를 표준화(Standardization)하면 약 24,700원의 평균 제공근 편차를 나타내며, 이를 시각화한 결과는 <Fig. 2>와 같다. 그리고 데이터셋의 범위를 벗어난 이후의 1달 가격인 2017년 8월의 예측값은 668,400원으로 실제값 675,000원과 약 6,600원의 차이로 9C4의 Test RMSE를 표준화한 범위 내에 값을 예측하였으며, 이는 실제값의 약 99.02%의 정확도로 예측하였다. 데이터의 가장 마지막 철근 가격이 하락하는 추세인 2003년 1월부터 2019년 11월까지의 데이터셋을 사용해 진행한 실험 결과에서 가장 우수

Table 6. data combinations and random search results

Combin	dataset	Lrata	Seq	Hid	Iters	Do	Train RMSE	Test RMSE	Standardization Test RMSE (Won)	<i>P</i>	Test RMSE Rank	Pred (Won)
2017. 7												
9C2	E, G	0.07377	17	321	5079	0.60690	0.02853	0.02803	29,200	0.99950	8	645,400
9C3	C, D, G	0.07429	3	425	6286	0.68566	0.02775	0.02700	28,100	0.99925	6	650,700
9C4*	B, F, G, I	0.05101	14	275	4423	0.70756	0.02419	0.02375	24,700	0.99956	1	668,400
9C5	A, E, F, G, I	0.02458	7	334	6943	0.64823	0.02698	0.02675	27,900	0.99978	5	645,000
9C6	B, C, E, F, G, I	0.01192	34	447	4922	0.66360	0.02471	0.02471	25,700	0.99999	2	672,500
9C7	A, C, E, F, G, H, I	0.01185	27	356	4728	0.60280	0.02669	0.02642	27,500	0.99972	4	684,700
9C8	A, B, C, D, E, F, G, H	0.04129	37	235	6373	0.53722	0.02629	0.02618	27,300	0.99988	3	665,800
9C9	A, B, C, D, E, F, G, H, I	0.08264	29	459	1868	0.45788	0.02775	0.02771	28,800	0.99996	7	680,300
2019. 11												
9C2	B, I	0.03767	8	477	2969	0.65006	0.03134	0.03111	23,600	0.99979	7	661,900
9C3	D, G, I	0.02398	38	269	2661	0.69087	0.02917	0.02845	25,000	0.99927	6	657,500
9C4	E, F, G, I	0.05652	38	427	2336	0.93379	0.02688	0.02630	27,400	0.99942	3	654,700
9C5	B, D, E, G, H	0.08402	32	335	1819	0.71480	0.02753	0.02715	28,300	0.99961	4	650,600
9C6	A, C, D, F, G, I	0.08586	9	181	3796	0.61096	0.03299	0.03278	26,300	0.99980	8	655,200
9C7	A, C, D, E, F, G, I	0.03758	37	162	2645	0.92502	0.02591	0.02571	26,800	0.99980	2	644,300
9C8	A, B, C, D, E, G, H, I	0.09968	16	295	1019	0.91113	0.02881	0.02763	25,800	0.99903	5	655,500
9C9*	A, B, C, D, E, F, G, H, I	0.06567	37	32	4782	0.85645	0.02543	0.02527	26,300	0.99984	1	649,000
2020. 3												
9C2	B, G	0.06848	20	496	6900	0.66560	0.02822	0.02735	25,500	0.99912	6	624,500
9C3	F, G, H	0.06769	35	353	1220	0.79602	0.02711	0.02696	28,500	0.99984	5	616,200
9C4	C, D, E, I	0.03825	10	326	2416	0.93971	0.02708	0.02667	21,900	0.99958	4	601,600
9C5*	A, C, E, G, H	0.08589	12	162	5499	0.80873	0.02554	0.02466	23,000	0.99911	1	603,200
9C6	A, D, E, F, G, H	0.05481	34	106	2106	0.78411	0.02530	0.02468	25,700	0.99937	2	627,500
9C7	B, C, D, F, G, H, I	0.08439	19	214	4529	0.37001	0.02774	0.02767	25,900	0.99994	7	628,000
9C8	A, B, D, E, F, G, H, I	0.07108	29	476	2015	0.52974	0.02658	0.02596	27,000	0.99938	3	628,700
9C9	A, B, C, D, E, F, G, H, I	0.08087	12	132	5388	0.42928	0.03193	0.03103	29,000	0.99910	8	619,900

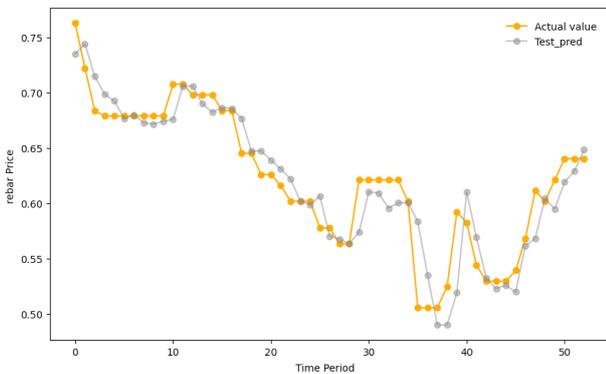


Fig. 2. Test period prediction (2013.4.~2017.7.)

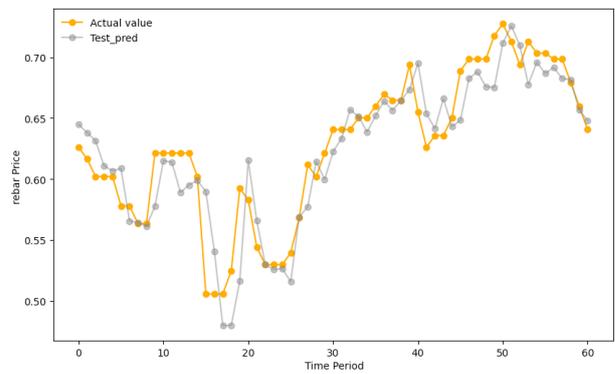


Fig. 3. Test period prediction (2014.11.~2019.11.)

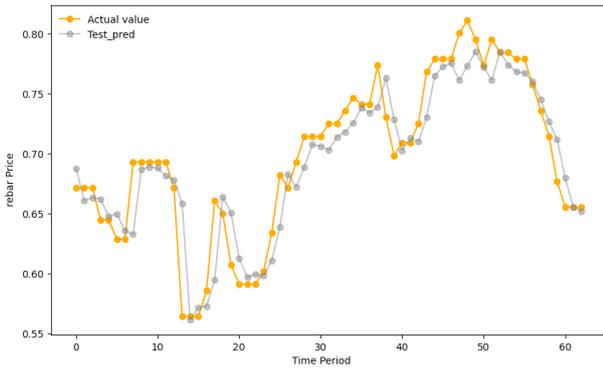


Fig. 4. Test period prediction (2015.1.~2020.3.)

한 예측성능을 나타낸 모든 데이터 집합인 9C9에 2014년 11월 ~ 2019년 11월까지의 Test RMSE는 0.02527로 이를 표준화하면 약 26,300원의 평균편차를 나타내며, 이를 시각화한 결과는 <Fig. 3>과 같다. 그리고 데이터셋의 범위를 벗어난 이후의 1달 가격인 2019년 12월의 예측값은 649,000원으로 실제값 630,000원과 약 19,000원의 차이로 9C5의 Test RMSE를 표준화한 범위 내에 값을 예측하였으며, 이는 실제값의 약 97.07%의 정확도로 예측하였다. 가장 최근 가격인 2003년 1월부터 2020년 3월까지 실험 결과에서 가장 우수한 예측성능을 나타낸 9C5의 데이터 A, C, E, G, H에 2014년 11월 ~ 2019년 11월까지의 Test RMSE는 0.02466로 이를 표준화하면 약 23,000의 평균 제곱근 편차를 나타내며, 이를 시각화한 결과는 다음 <Fig. 4>와 같다. 그리고 데이터셋의 범위를 벗어난 이후의 1달 가격인 2020년 4월의 예측값은 603,200원으로 실제값 610,000원과 약 6,800원의 차이로 9C5의 평균 제곱근 편차 범위 내에 값을 예측하였으며, 이는 실제값의 약 98.89%의 정확도로 예측하였다. 또한 본 연구에서 진행한 데이터 조합과 하이퍼 파라미터 탐색을 동시에 탐색하는 방식은 한가지 데이터셋의 하이퍼 파라미터를 집중탐색하는 방식인 9C9 Test RMSE보다 개선된 결과를 나타낸 2017년 7월의 결과와 2020. 3월의 결과를 통해 이 방식이 Test RMSE의 개선에 효과가 있음을 확인할 수 있었다. 그리고 본 연구에서 제안한 P값은 Dropout을 적용하였을 때 모든 결과에서 학습성능과 테스트성능의 차이가 작은 과대적합이 발생하지 않은 결과의 선별이 가능하였다. 또한 이후 Test RMSE가 낮은 결과를 추가 선별하여 확인한 실제값과 예측값의 잔차는 평균 약 10,900원의 차이를 나타냈다.

8. 결론

본 연구에서는 딥러닝 모델인 Bi-LSTM과 예측성능을 높이기 위한 데이터 조합, Dropout, Random Search, 학습성능과 테스트성능의 점점 값 탐색방법을 이용하여 철근 가격

단기예측방법을 제안하기 위해 실험을 진행하였다. 그리고 데이터의 마지막 철근 가격이 오르는 추세인 2003년 1월부터 2017년 7월까지, 하락하는 추세인 2019년 11월까지, 가격변화가 없는 최근의 철근 가격인 2020년 3월까지를 3가지 데이터셋으로 구성해 데이터셋의 범위를 벗어난 이후의 1달 가격인 2017년 8월, 2019년 12월, 2020년 4월을 예측하였다. 예측결과는 실제값의 약 99.02%, 97.07%, 98.89%의 정확도를 나타냈다. 하지만 이 정확도는 비결정적이며 Random Search는 좋은 결과를 지나칠 수 있는 불확실성이 내재되어 있기 때문에 탐색 수를 증가시키면 추가 개선될 가능성이 충분히 있다.

본 연구에서 제시하는 양방향 LSTM과 데이터 조합탐색 및 딥러닝 관련 기법을 활용한 철근 가격 단기예측에 관한 실험적 연구는 다른 건축재료 보다 상대적 가격변동 폭이 큰 철근의 1달 이후 가격을 예측함으로써 자재의 가격변동으로 달라질 수 있는 실투입비에 대한 대비 및 매월 수행된 작업에 대한 예산산정의 유연한 관리가 가능할 것이다. 또한 철근 이외의 건축재료를 비롯하여 통계기반의 시계열 데이터로 가격을 예측하는 연구에서 본 연구에서 제시한 내용이 우수한 예측결과를 도출하기 위한 기초적 자료로 활용될 수 있을 것이다.

향후 건축자재의 가격변동으로 달라질 수 있는 미래의 여러 시점에 대한 실투입비 검토 및 관리가 가능하도록 딥러닝 예측범위를 보다 확대하기 위한 추가적 연구가 필요하다.

References

Bergstra, J., and Bengio, Y. (2012). "Random search for hyper-parameter optimization." *The Journal of Machine Learning Research*, 13(1), pp. 281-305.

Cen, Z., and Wang, J. (2019). "Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer." *Energy*, 169, pp. 160-171.

Chen, X., Wei, L., and Xu, J. (2017). "House price prediction using LSTM." *arXiv preprint arXiv:1709.08432*.

Choi, M., and Kwon, O. (2008). "Construction material cost increase and countermeasures." *Construction trend briefing by Korea Institute of Construction Industry*, Vol. 2008 No.6, pp. 2-34.

Choi, Y., Yim, H., and Park, B. (2009). "Analysis on the Lotting Price Fluctuation of the Multi-Family Attached House According to the Construction Material Cost Variation." *Journal of The Korean Society of Civil Engineers*, 29(6D), pp. 753-760.

Claesen, M., and De Moor, B. (2015). "Hyperparameter search in machine learning." *arXiv preprint*

- arXiv:1502.02127*.
- Gao, X., Shi, M., Song, X., Zhang, C., and Zhang, H. (2019). "Recurrent neural networks for real-time prediction of TBM operating parameters." *Automation in Construction*, 98, pp. 225-235.
- Hochreiter, S., and Schmidhuber, J. (1997). "Long short-term memory." *Neural computation*, 9(8), pp. 1735-1780.
- Jang, Y., Jeong, I., and Cho, Y. (2020). "Business Failure Prediction of Construction Contractors Using a LSTM RNN with Accounting, Construction Market, and Macroeconomic Variables." *Journal of Management in Engineering*, 36(2), 04019039.
- Jeong, D. (2017). "Trend on Artificial Intelligence Technology and Its Related Industry." *Korea Institute of Information Technology Magazine*, 15(2), pp. 21-28.
- Joo, I., and Choi, S. (2018). "Stock prediction model based on bidirectional LSTM recurrent neural network." *The Journal of Korea Institute of Information, Electronics, and Communication Technology*, 11(2), pp. 204-208.
- Lahari, M. C., Ravi, D. H., and Bharathi, R. (2018). "Fuel Price Prediction Using RNN." *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1510-1514. IEEE.
- Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). "An empirical evaluation of deep architectures on problems with many factors of variation." *Proceedings of the 24th international conference on Machine learning*, pp. 473-480.
- Lee, J. (2019). "A Comparative Study on Stock Price Forecasting Models Using LSTM and Bidirectional Neural Networks." MS thesis, Seoul National University of Science and Technology Graduate School.
- Liao, T. W. (2005). "Clustering of time series data—a survey." *Pattern recognition*, 38(11), pp. 1857-1874.
- Mou, L., Zhao, P., and Chen, Y. (2019). "Short-Term Traffic Flow Prediction: A Long Short-Term Memory Model Enhanced by Temporal Information." *CICTP 2019*, pp. 2411-2422.
- Olson, D. L., and Delen, D. (2008). *Advanced data mining techniques*, Springer-Verlag, Berlin Heidelberg.
- Pawar, K., Jalem, R. S., and Tiwari, V. (2019). "Stock market price prediction using LSTM RNN." *Emerging Trends in Expert Applications and Security*, pp. 493-503. Springer, Singapore.
- Schuster, M., and Paliwal, K. K. (1997). "Bidirectional recurrent neural networks." *IEEE transactions on Signal Processing*, 45(11), pp. 2673-2681.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research*, 15(1), pp. 1929-1958.
- Trinh, T. H., Dai, A. M., Luong, M. T., and Le, Q. V. (2018). "Learning longer-term dependencies in rnns with auxiliary losses." *arXiv preprint arXiv:1803.00144*.
- Yoon, J. (2019). "Effectiveness analysis of credit card default risk prediction using deep learning neural networks." *Financial Research*, 33 (1), pp. 151-183.
- Zhang, Z., Wang, Y., Chen, P., and Yu, G. (2018). "Application of long short-term memory neural network for multi-step travel time forecasting on urban expressways." *CICTP 2017: Transportation Reform and Change—Equity, Inclusiveness, Sharing, and Innovation*, ASCE, pp. 444-454.

요약 : 본 연구는 양방향 LSTM, Random Search, 데이터 조합, Dropout을 이용한 철근 가격 단기예측 딥러닝 모델을 개발하는 체계적인 절차를 제시한다. 일반적으로 사용자가 직관적으로 이러한 값을 결정하여 예측성능이 우수한 결과를 탐색하는데 시간이 많이 걸리고 반복적인 시도를 하게 되는데, 이러한 시도로 찾아낸 결과가 우수하다고 보장할 수 없다. 본 연구에서 제시하는 제안된 접근방식으로 단기 가격예측의 평균 정확도는 약 98.32%이다. 그리고 이 방식은 철근 이외의 건축재료를 비롯하여 통계기반의 시계열 데이터로 가격을 예측하는 연구에서 본 연구에서 제시한 내용이 우수한 예측결과를 도출하기 위한 기초적 자료로 활용될 수 있을 것이다.

키워드 : 철근 가격, 양방향 순환신경망, 데이터 조합, 하이퍼 파라미터 무작위 탐색, 가격예측, Dropout
