

# 딥러닝을 활용한 철근 가격 단기에측범위 확대에 관한 실험적 연구

## Experimental Study on the Expansion of the Short-term Prediction Range of Rebar Prices Using Deep Learning

이 용 성\*      김 경 환\*\*  
Lee, Yong-Seong      Kim, Kyung-Hwan

### Abstract

This study presents a method for expanding the prediction range of rebar price prediction using the short-term prediction method of deep learning. In general, the prediction range of a short-term prediction is dependent on the time interval of the data to be entered, so it can be expanded by adjusting the time interval of the data. However, as the range of forecasts increases, the size of the data decreases, which can lead to overfitting that cannot guarantee good results. The average accuracy of the forecasts is approximately 98.49% when the scope of the forecasts is extended from 1 month to 2 and 3 months with the proposed approach presented in this study. In addition, this approach could be used as a basis for expanding the predictive range of deep learning in a study that predicts prices with time series data including common building materials.

키워드 : 딥러닝, 적층 양방향 LSTM, 예측범위 확장, 철근 가격

Keywords : Deep learning, Stack bidirectional LSTM, Prediction range expansion, Rebar price

### 1. 서 론

#### 1.1 연구의 배경 및 목적

건설공사의 원가에서 자재비가 차지하는 비율은 상당히 높다 (Park & Jung, 2016). 특히 자재비 항목 중 철근은 국제유가, 환율, 철근 수입물가 등과 같은 거시경제 흐름에 영향받아 시세가 달라진다 (Choi et al., 2009). 또한 철근 단가는 다른 재료에 비해 가격 인상요인이 있으면, 이를 즉시 시장에 반영할 수 있는 시장 지배력을 가지기 때문에 단가변동이 자주 발생한다 (Choi & Kwon, 2008). 그래서 이와 같은 자재를 수급하기 위한 계획은 최소한의 비용으로 최대의 이윤을 남기는 것을 목적으로 하는 건설기업에서는 필수적이다. 하지만 자재의 구매단계에서 사용계획에 따른 구매 시기를 결정하는 기존의 방식은 경험과 직관을 바탕으로 이뤄지기 때문에 이윤 또는 손실에 대한 예측이 불확실성을 내포한다 (Lee et al., 2008). 또한, 이러한 불확실한 예측의 과정으로 인해 발생할 수 있는 문제를 건설사업관리 측면에서 가정해보면 자재의 투입 시기가 결정된 이후 시세의 변동과 자재구

매 시기에 대한 판단오류로 계획된 비용보다 실투입비가 많아질 수 있는데, 이는 책정된 예산에 상당한 부담으로 작용할 수 있다. 그리고 공사의 규모와 면적이 커질수록 이 부담이 가중된다. 따라서 이러한 비용을 산정하는 기존의 방식은 주문 시기와 주문량, 주문 시기에 책정된 단가를 대상으로 관련 비용함수를 최적화하여 기업이윤을 남기고자 하는 의사결정의 한 범주로 이해할 수 있다 (Lee et al., 2008). 그렇기 때문에 기존의 방식보다 정확한 주문 시기에 대한 판단과 이에 따른 책정된 단가의 예측이 가능하다면 이러한 불확실성을 가지는 예측의 체계성을 보완할 수 있을 것으로 사료된다.

이에 본 연구에서는 철근의 주문 시기와 주문 시기에 책정된 단가를 사전에 파악하기 위한 새로운 대안으로 딥러닝의 시계열 예측을 활용하고자 한다. 딥러닝의 시계열 예측은 시간에 따라 관측 및 기록된 값들로 이루어진 데이터를 컴퓨터에 학습시켜 입력데이터에 포함되지 않은 다음의 미래값을 추정한다 (Kim & Baek, 2019). 그리고 최근 금융, 기상, 교통, 통신 등과 같은 여러 분야에서 딥러닝 알고리즘과 특정 값을 예측하기 위해 사용할 일 (Day), 월(Month), 연(Year) 등의 시간 단위로 구성된 데이터로 사용한 데이터의 시간 단위 만큼에 미래 특정 값을 예측하는 연구가 진행된 바 있다 (Fischer & Krauss, 2018; Liu et al., 2018; Lv et al., 2014; Polson & Sokolov, 2017). 따라서 철근 가격을 예측하기 위해 시계열 데이터

\* 건국대학교 일반대학원 박사수료

\*\* 건국대학교 건축학부 교수, 공학박사

(Corresponding author : Department of Architecture, Konkuk University, [kykim@konkuk.ac.kr](mailto:kykim@konkuk.ac.kr))

를 구성하고, 딥러닝 관련 기술에 관한 이론적 고찰을 통해 모델을 구성하면 딥러닝을 활용한 철근 가격의 예측이 가능할 것으로 사료된다. 하지만 철근 가격은 한국물가정보에서 2000년 1월을 시작으로 현시점까지 월별로 제공되기 때문에 한 달 단위의 시계열 데이터 구성만 가능하다. 그리고 이 데이터와 딥러닝을 활용해 미래의 철근 가격을 예측하게 되면 결과적으로 사용한 데이터에 마지막 시점 이후의 한 달을 예측하게 된다. 하지만 전반적인 건설 프로젝트의 기간과 비교할 때 철근의 주문 시기와 주문 시기에 책정된 단가를 사전에 파악하려면 딥러닝의 예측 기간 확대가 필요하다. 이에 따라 본 연구에서는 딥러닝의 활용하여 철근 가격을 예측할 때 예측범위를 추가로 확대할 수 있는 방법을 제안하고자 한다.

## 1.2 연구의 방법

연구의 방법으로는 건축프로젝트에서 이뤄지고 있는 자재의 계획, 구매단계에 관한 문헌 고찰을 통해 딥러닝의 활용 범위를 고찰한다. 그리고 본 연구에 사용할 모델을 구성하기 위해 딥러닝을 활용하여 특정 값을 예측하는 연구에서의 다루고 있는 예측방식에 대한 내용과 시계열 데이터를 사용하는 순환신경망 딥러닝 모델을 구성하기 위한 이론적 고찰을 실시한다. 그리고 이를 통해 예측범위를 확대하기 위한 방법을 가정한 후, 모델을 구성하고 실험을 진행한다. 이후 실험 결과를 통해 가장 우수한 예측성능을 나타낸 결과에 데이터의 범위를 벗어난 n개월 예측값을 한국물가정보에서 제공하는 실제 철근 가격과 비교하여 최종적인 예측성능을 판단한다.

## 2. 건설프로젝트의 자재 사용계획 및 구매단계에서 딥러닝의 활용 범위에 관한 고찰

건설프로젝트에서 자재를 현장에 조달하기 위해 자재의 구매 시기를 결정하는 기존의 방식은 자재를 사용하기 위한 계획수립과 구매단계를 거치는 프로세스를 통해 결정된다 (Lee & Choi, 2007). 이에 대한 세부적인 프로세스는 Figure 1과 같다.

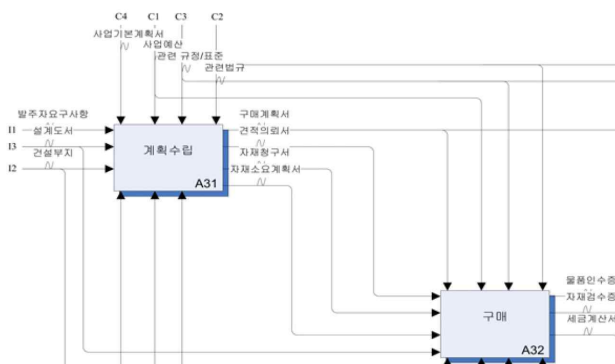


Figure 1. Plan establishment stage<sup>1)</sup>

Figure 1과 같이 자재의 구매절차는 계획수립단계에서 구매계획, 견적 의뢰, 자재청구, 자재 소요계획의 과정에 의해 구매가 이뤄진다. 그리고 견적 의뢰와 자재청구의 중간 과정에 있는 자재의 계약방식은 건설회사로 직접 판매하는 형태와 도·소매유통대리점을 통해 판매하는 간접형태로 이뤄지고 있으며 견적을 의뢰하는 시점에 책정된 자재의 단가와 사전에 계획된 물량에 따라 가격협상을 통해 구매가 이뤄진다 (Park & Jung, 2016; Lee et al., 2006). 그래서 자재를 구매할 때 구매 시점에 책정된 단가로 결정되는 자재구매의 과정들은 책정된 단가에 대한 선택 폭이 제한적이다. 그렇기 때문에 계획된 예산을 초과하지 않는 범위에서 자재비를 산정하는 기존의 방식이 기업의 이윤과 손실에 대한 불확실성을 가진다. 또 기존의 계획된 물량에 대한 자재 단가를 추정하여 사업예산 편성하는 과정도 미래의 자재 단가에 대한 의사결정이 책임자의 경험과 판단에 의존적이다. 그렇기 때문에 딥러닝을 활용하여 자재의 미래 단가를 예측한다는 것은 두 가지 측면에서 의미가 있다.

첫 번째로 자재의 물가 변동을 사전에 인지하여 계획된 자재의 투입 시점 이전에 자재구매를 위한 적절한 시기의 판단이 가능해질 수 있다. 두 번째로 자재비용을 계획할 때 기존의 경험과 판단을 기반으로 한 인간의 불확실한 의사결정의 체계성을 보완함으로써 보다 정확한 예산편성을 가능하게 할 수 있을 것으로 사료된다.

## 3. 딥러닝 예측방법에 관한 문헌 고찰

딥러닝 학습방법의 한 종류인 지도학습은 명시적인 정답이 주어진 상태에서 데이터의 특성을 토대로 컴퓨터를 학습시켜 값을 예측한다 (Olson & Delen, 2008). 기존의 딥러닝을 활용하여 값을 예측하는 연구에서 특정 값을 예측하기 위해 사용한 시계열 데이터의 시간 단위와 예측방식에 대해 알아보기 위해 문헌 고찰을 실시하였다.

예측방식에 관한 선행연구들을 살펴보면 대부분 단기 예측에 관한 연구가 주를 이루고 있다. 단기예측 방식은 딥러닝 학습에 입력된 데이터의 다음을 예측하는 방식이다. 단기예측 방식의 기존 연구로는 일 단위인 Brent crude oil, 환율의 시계열 데이터로 입력된 데이터 시점의 다음 시점인 하루 후를 예측하는 방식으로 연료 가격을 단기예측하였다 (Lahari et al., 2018). 그리고 Seo & Yeom(2019)는 대기오염 측정망을 통하여 측정된 도심지역의 일 단위 데이터로 입력된 데이터 시점의 다음 시점인 하루 후를 예측하는 방식으로 미세먼지의 농도를 예측하였다. Lee et al.(2020)은 한 달(Month) 간격의 거시경제 데이터를 활용해 입력된 데이터 시점의 다음 시점인 한 달 후를 예측하는 방식으로 철광석 가격을 단기예측하였다. Lee & Kim(2019)은 월별, 계절별, 연도별로 정리

1) 이재민, & 최중수. (2007). 자재구매 및 조달관리가 공정에 미치는 영향분석: 철골공사 프로젝트를 중심으로. 대한건축학회 논문집-구조계, 23(12), pp. 141-148.

된 기상요소 데이터로 데이터의 시간 간격에 따라 입력된 데이터의 다음 값을 예측하는 단기에측방식으로 태양광 발전량을 단기, 중기, 장기로 나눠 예측하였다.

이상의 선행연구를 분석한 결과 순환신경망의 단기에측범위는 입력데이터의 시간 간격에 따라 결정되는 것으로 파악되며, 예측범위를 확대하기 위해서 딥러닝 학습에 사용할 시계열 데이터의 시간 간격을 재구성하면 원하는 기간만큼 예측이 가능할 것으로 판단된다.

#### 4. 딥러닝 관련 이론적 고찰

##### 4.1 시계열 데이터

시계열 데이터는 정해진 시간 간격에 따라 변경 또는 관측된 값으로 구성된다. 그리고 과학, 엔지니어링, 비즈니스, 금융, 경제, 의료 등 다양한 분야에서 현재와 미래에 대한 분석 및 예측에 사용하고 있다 (Liao, 2005).

##### 4.2 Recurrent Neural Networks

순환신경망(Recurrent Neural Networks, RNN) 은닉계층에 이전 정보를 기억시킬 수 있는 순환구조를 이루는 인공신경망의 한 종류로 이전 출력의 영향을 받아 현재까지 계산된 결과를 메모리에 기억한다. 학습은 입력데이터를 비선형 관계로 변환하여 다음 단계의 입력으로 전달하며 출력값은 -1 ~ 1 사이의 작은 값으로 출력한다 (Schuster & Paliwal, 1997).

##### 4.2.1 Long Short-Term Memory

RNN의 출력값이 곱해지는 단계가 반복되면 이 값이 소실되어 학습이 이루어지지 않는 기울기 소실(Vanishing gradient)이 발생한다 (Gao et al., 2019). 이는 과거의 학습결과가 사라지는 장기 의존성(Long-Term dependencies) 문제를 발생시키는데, 이 문제를 극복하기 위해 제안된 순환신경망 구조가 Long Short-Term Memory (LSTM)이다. LSTM은 셀 상태를 유지하면서 Input, Output, Forget 게이트라 불리는 구조에 의해 제어되며, 이를 통해 RNN의 문제점을 방지하여 시계열 데이터와 같은 방대한 양의 데이터도 기울기 소실 문제없이 처리한다 (Hochreiter & Schmidhuber, 1997).

##### 4.2.2 Bidirectional Recurrent Neural Networks

RNN은 예측결과가 직전 패턴을 기반으로 나타내는 경향을 보이는 한계가 있다. 이러한 RNN의 단점을 보완하기 위해 Bidirectional Recurrent Neural Networks (BRNN)이 제안되었다. BRNN은 순방향과 역방향으로 분리된 서로 연결되지 않은 두 개의 은닉층을 가지며, 순환신경망을 통해 학습시킨다. 입력값은 두 개의 은닉계층으로 전달되고, 출력계층에서는 두 개의 은닉층의 출력이 하나로 합쳐져 입력받은 후 가중치 연산과 최종 출력을 한다 (Schuster & Paliwal, 1997).

##### 4.3 하이퍼 파라미터 최적화

하이퍼 파라미터는 사용자가 직접 조절하여 설정해야

하는 입력데이터의 크기, Cell 내부의 은닉층 수, 학습량, 학습률 등의 값들을 지칭한다. 이 값들은 입력되는 조합에 따라 딥러닝 모델의 예측성능이 달라지는데, 이때 모델의 예측성능을 최대로 끌어올리기 위해 수행되는 하이퍼 파라미터 탐색 과정을 하이퍼 파라미터 최적화라 한다 (Bergstra & Bengio, 2012). 그리고 이러한 최적화 기법 중 사용자가 직접 시도할 하이퍼 파라미터 입력값을 선정하고, 이를 사용하여 예측결과를 몇 차례 측정한 후 가장 높은 예측성능을 나타낸 입력값을 채택하는 방법을 Manual search라 한다 (Larochelle et al., 2007). 또한 시도할 하이퍼 파라미터 후보 입력값을 범위로 선정하고, 이를 무작위로 조합하여 자동으로 탐색하는 과정을 Random search라 한다 (Bergstra & Bengio, 2012).

##### 4.4 과대적합

과대적합(Overfitting)은 딥러닝 모델이 데이터를 너무 상세하게 학습하는 것을 말한다. 이는 학습데이터가 부족하거나, 데이터의 특성에 비해 모델이 너무 복잡한 경우 발생한다. 그리고 이 문제의 결과는 학습성능은 우수하지만, 학습데이터 이외에 다양한 변수에 대한 대응력이 떨어져 일반화된 모델의 예측성능을 감소시키기 때문에 필수적으로 확인이 필요한 사항이다 (Tensorflow.org, 2020).

##### 4.5 Dropout

Srivastava et al.(2014)는 과대적합을 방지하는 기술인 Dropout을 제안하였다. 사용방법은 0.1 ~ 0.9까지의 입력 비율에 따라 신경망에 뉴런들을 확률적으로 사용하지 않음으로 일반화 성능이 향상시킨다. 입력 비율 1.0은 Dropout을 사용하지 않는 디폴트 값을 의미한다.

##### 4.6 데이터 정규화: MinMaxscaler scaler

데이터 간의 편차를 줄이기 위해 모든 입력데이터는 정규화(Normalization)를 진행하여 0 ~ 1 사이의 값으로 조정하여 적용한다. 정규화에 사용된 방법은 MinMaxscaler로 Equation 1과 같다 (Chung et al., 2019).

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

$x$  : 원본데이터

$x_{\min}$ : 데이터의 최소값

$x_{\max}$ : 데이터의 최대값

$X$  : 스케일 변환데이터

##### 4.7 모델 성능평가

딥러닝 모델의 예측성능을 측정 및 평가하기 위해 평균 제곱근 오차인 Root Mean Square Error(RMSE)를 이용하였다. RMSE는 모델의 예측값과 실제값의 차이를 다룰 때 사용하는 척도로 Equation 2와 같다 (Joo & Choi, 2018). RMSE는 수치가 낮을수록 예측값과 실제값의 오차가 작다는 것을 의미한다.

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

$R$  : 실제값

$P$  : 예측값

$n$  : 데이터의 개수

## 5. 딥러닝 환경 구성

### 5.1 데이터셋 구성

본 연구에서는 철근 가격예측을 위해 보통철근(일반 SD300, D10mm, 0.560), 철근 수입물가지수, 철근 생산량, 국제유가 3종, 생산자물가지수(일반 철근), 고철 단가와 같은 철근 생산과 수요에 직·간접적 연관성이 있는 월별 시계열 통계자료로 데이터를 임의로 선정하여 데이터셋을 구성한다. 그리고 본 연구에서 사용하는 데이터의 출처와 이 데이터 항목들을 구분하기 위해 사용하는 ID는 Table 1과 같다. 모든 데이터는 총 207행으로 2003년 1월부터 2020년 3월까지 1개월 간격으로 Comma-Separated Values(CSV) 파일로 정리하였다. 그리고 데이터를 식(1)을 통해 정규화하여 모델에 적용한다.

Table 1. Data properties

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	Rebar price (SD300, general)	Korea Price Information

### 5.2 딥러닝 모델 구성

본 연구에서는 철근 가격예측의 예측범위 확대에 적합한 딥러닝 모델을 선택하기 위해 앞에서 설명한 RNN, LSTM과 LSTM, BRNN을 결합한 형태의 양방향 LSTM(Bi-LSTM)에 Cell Layer의 계층을 추가확대 가능하도록 Stack bidirectional dynamic RNN을 추가하여 모델을 3가지 형태로 각각 구성한다. 그 외에 시스템 구성과 사용된 패키지 정보는 Table 2와 같다. 그리고 본 연구의 딥러닝 모델에서 사용되는 하이퍼 파라미터는 Table 3과 같다. 하이퍼 파라미터 최적화는 Manual search를 이용하여 탐색할 입력값의 범위를 큰 폭으로 줄이고, Random search를 이용하여 축소된 입력값 범위 내에서 하이퍼 파라미터를 무작위로 조합하여 예측결과를 탐색할 수 있도록 설정하였다. 또 Random search에는 철근 생산과 수요

에 직·간접적 연관성이 있는 임의로 선정한 통계자료 데이터 중 철근 가격에 영향을 미치는 인자를 탐색하기 위해 데이터를 조합(Combination)하여 각각의 데이터셋으로 구성하고, 조합별로 구성된 데이터를 무작위로 선택하여 하이퍼 파라미터와 같이 탐색할 수 있도록 설정하였다. 데이터의 조합 수는 Table 4와 같다. 그리고 Dropout은 Cell에 다음 행에 위치시켜 학습 과정의 Cell 연산을 확률적으로 사용하지 않게 설정하였다. 마지막으로 입력 데이터는 전체 데이터의 70%는 학습에 나머지 30%는 15%씩 분할하여 Test1, Test2로 구성한다.

Table 2. System configuration specifications and package information

Classification	Program information
Operation system	Window 10
Programming Tool	Visual Studio Code
Graphics Processing Units(GPU)	RTX 2080 Ti
General-Purpose computing on GPU	Cuda 10.0
Language	Python 3.6.5
Deep Learning Library	Tensorflow- GPU 1.15.0
matplotlib 3.1.1	Data Visualization
numpy 1.16.5	Array
Pandas 0.25.1	Data Time Interval Reconfiguration
sklearn 0.23	RandomizedSearchCV Train_test_split

Table 3. Hyperparameter and description

Hyper parameter	Description
Sequence length(Seq)	Row length of input data
Learning rate(Lr)	Gradient Descent Learning speed
Hidden layer(Hd)	Hidden layer inside the Cell
Input(Inp)	Input data size
Output(Otp)	Output data size
Iterations(Iter)	Iterative learning
number_layer(nL)	Number of LSTM Cell layers
keep_prob(kp)	Dropout rate
Month	Prediction period

Table 4. Number of data combinations

Combination	6C2	6C3	6C4	6C5	6C6
Number of cases	15	20	15	6	1

6. 순환신경망 예측범위 확대를 위한 데이터 시간 단위 재구성

본 연구에서 사용하는 1달 단위 시계열 데이터의 시간 간격을 재구성하는 방법은 다음 Figure 2와 같다. Figure 2를 살펴보면 정의된 함수에 입력값인 dataset은 재구성이 필요한 데이터셋의 원본이고, month는 재구성할 데이터의 시간 간격을 결정한다. 그리고 데이터를 쉽고 직관적으로 작업할 수 있도록 유연한 데이터 구조를 제공하는 파이썬 패키지인 Pandas의 'ix'는 레이블의 위치 정수를 사용하여 데이터를 인덱싱하는데 이 위치 정수 입력값이 month이다. 이를 적용한 예로 현재 2003년 1월부터 2020년 3월까지의 데이터를 3 month를 입력값으로 사용하여 데이터의 시간 간격을 재구성하면 Figure 3과 같은 결과를 얻을 수 있다. Figure 3의 (A)와 (B)를 살펴보면 데이터 시간 간격 재구성 적용 전후로 3의 배수에 해당하는 값들만 존재하는 것을 알 수 있다.

```
def rebuild_dataset(dataset, month):
    dataset = dataset[::-1]
    dataset = dataset.ix[::month,:]
    dataset = dataset[::-1]
    return dataset
```

Figure 2. Data time reconstruction method

단순히 ix로 데이터를 재구성할 경우 최신의 마지막 데이터가 반영되지 않을 수 있기 때문에 Figure 2의 정의된 함수 내에 'dataset[::-1]'을 통해 전체 데이터의 순서를 뒤집어 맨 앞의 0번 데이터가 206번이 되도록 하여 제일 오래된 앞 데이터가 반영되지 않도록 설정하였다. 그리고 ix를 적용하여 데이터의 시간 간격을 조정 후 'dataset[::-1]'를 다시 적용하여 원래의 시간 순서로 되돌린다. 이러한 방식을 적용하여 1개월의 시간 간격을 나타내는 데이터는 Figure 4와 같이 3개월의 시간 간격을 나타내는 데이터로 재구성된다.

	a	b		a	b
0	50.24	51.11	0	50.24	51.11
1	50.71	51.11	3	54.41	56.28
2	52.55	51.11	6	53.25	56.25
3	54.41	56.28	9	53.62	56.25
4	53.73	56.25	12	58.71	62.14
...	...	...	...	...	...
202	102.88	109.47	192	100.45	115.77
203	102.85	108.61	195	102.40	112.80
204	101.57	107.97	198	103.32	112.30
205	104.16	108.49	201	104.35	110.53
206	106.44	108.56	204	101.57	107.97

(A) Before

(B) After

Figure 3. Comparison of before and after data time interval reconstruction

	a	b	c	d	e	f	g	h	i	j
2	52.55	51.11	31114.08	35254.26	39018.06	217045.8	562000	993	13751	365000
5	53.42	56.25	32971.50	35298.90	39565.80	195243.0	559000	1023	18765	402000
8	52.69	56.25	29212.54	30592.66	32432.82	210468.3	461000	877	9825	402000
11	55.44	56.31	32915.76	35658.74	38282.46	258794.2	700000	913	9349	402000
14	63.70	74.31	35200.62	38755.08	42080.22	355446.0	552000	1015	12975	493000
...	...	...	...	...	...	...	...	...	...	...
194	100.80	112.62	75938.19	76051.70	66062.82	355286.3	633000	955	17351	720000
197	103.79	111.70	71300.46	72746.10	63162.09	386824.5	567000	906	16444	730000
200	103.94	111.84	73087.82	74523.26	68183.40	377999.2	425000	704	14924	705000
203	102.85	108.61	75050.36	75397.28	69152.72	367735.2	519000	790	14485	630000
206	106.44	108.56	41026.38	41026.38	37130.70	427307.4	491000	764	17704	610000

Figure 4. 3-month interval data reconstructed

## 7. 실험 방법

딥러닝을 활용한 철근 가격예측의 예측범위 확대를 위한 실험을 진행한다. 실험은 총 3가지의 방식으로 진행된다.

첫 번째 실험은 본 연구에서 선정한 데이터에 적합한 모델을 결정하기 위한 실험으로 RNN, LSTM, Stack Bi-LSTM에 각각 모든 데이터조합 6C6 데이터셋을 적용하여 출력된 결과의 예측성능을 비교하고 앞으로의 실험에 사용할 한 가지 모델을 선택한다.

두 번째 실험은 임의로 선정된 데이터를 조합하여 이를 딥러닝에 적용하고 출력된 결과 중에 학습성능이 우수하면서 Test1, 2의 예측성능이 학습성능과 비슷한 결과를 보이는 데이터조합을 탐색하기 위한 실험으로 모든 조합에 포함되는 예측대상인 철근 가격을 제외하고 나머지 입력변수를 조합하여 데이터셋을 생성해 이를 사용하여 예측결과를 각각 출력한다. 최종적으로 학습성능이 가장 우수한 결과를 나타낸 조합을 선택하고, 이 데이터셋을 세 번째 실험의 예측범위 확대를 위한 실험에 사용한다.

세 번째 실험은 앞 전의 실험 결과의 데이터셋에 시간 간격을 재구성하여 모델에 적용한 후 조정된 시간 간격에 따른 단기예측결과를 출력하는 실험으로 출력된 결과를 통해 데이터의 범위를 벗어난 예측값과 한국물가정보에서 제공하는 해당 시점의 실제값과 비교하여 정확도(%)로 평가한다.

실험의 모든 각각의 과정은 Random Search에 의해 하이라 파라미터를 무작위로 조합하여 50번씩 탐색을 수행한다. 하이라 파라미터의 탐색 범위는 학습률 0.0001 ~ 0.01, 시퀀스 길이 2 ~ 20, 은닉층 수 10 ~ 100, Dropout 비율 0.1 ~ 0.99, 신경망 층수 1 ~ 2, 학습량은 5000이다. 이는 Manual Search를 통해 일부 한정하였다. 그리고 모델에서 출력되는 결과는 학습량이 5000까지 1씩 증가할 때마다 출력되고, 다음 Equation 3에 의해 필터링 된다.

$$TrainRMSE \leq TestRMSE 2 \leq TestRMSE 1 \quad (3)$$

식(3)은 과대적합을 고려하기 위해 설정하였다. 과대적합으로 인한 문제는 데이터의 범위를 벗어난 예측값이 실제값과 전혀 다른 예측값을 나타낼 가능성이 있다. 그리고 출력결과는 모델의 하이라 파라미터와 학습, Test1, 2 RMSE, 데이터셋의 범위를 벗어난 다음의 예측결과, Test 구간의 시각화가 포함된다. 기록된 결과에서는 학습성능이 가장 우수한 한 개의 결과를 최종 선택한다.

## 8. 실험 결과

### 8.1 철근 가격예측에 적합한 예측모델 선택 실험

본 연구에서 선정한 데이터에 적합한 딥러닝 모델 선택을 위한 각각의 모델별 단기예측실험 결과는 Table 5와



같다. Table 5는 본 연구에서 선정한 Table 4의 6C6 조합 데이터를 3가지 모델에 적용하여 각각의 모델별로 1개월을 예측하였을 때 가장 우수한 예측성능을 나타내는 단기예측 결과를 정리한 것으로, 모델별 가장 우수한 학습 RMSE를 나타낸 Stack Bi-LSTM이 Test1, 2에 대한 RMSE 모두 우수한 결과를 나타냈다. 그리고 데이터의 범위를 벗어난 1달 후의 예측결과 또한 RNN은 실제값의 약 95.76%, LSTM은 약 92.84%, Stack Bi-LSTM은 약 99.83%로 이 또한 Stack Bi-LSTM이 모든 결과에서 우수하였다. 실험 결과에 따라 본 연구에서 진행할 나머지 실험에는 Stack Bi-LSTM을 사용한다.

Table 5. Experiments results to select prediction models

Model	Train RMSE	Test1 RMSE	Test2 RMSE	Pred	Actual value
RNN	0.03487	0.03551	0.03549	637,000	610,000
LSTM	0.04847	0.04938	0.04894	657,000	
Stack Bi-LSTM	0.02857	0.02950	0.02935	609,000	

### 8.2 데이터조합 탐색 실험

선정된 데이터를 조합(Combination)하여 모두 데이터셋으로 만들고, 이를 통해 1개월의 예측성능이 우수한 데이터조합을 탐색하기 위해 실험을 진행하였다. 실험 결과는 Table 6과 같다.

Table 6. Experiment results for data combination

Combin	Data	Train RMSE	Test1 RMSE	Test2 RMSE	Pred	Actual value
6C2	AE	0.02989	0.03035	0.03016	604,000	610,000
*6C3	ABF	0.02475	0.02554	0.02536	617,000	
6C4	ADEF	0.03287	0.03351	0.03317	619,000	
6C5	ABCDE	0.03308	0.03338	0.03319	622,000	
6C6	ABCDEF	0.02857	0.02950	0.02935	609,000	

\*Best Train RMSE

Table 6을 살펴보면 학습 RMSE가 가장 우수한 데이터조합은 6C3의 ABF로 나타났으며, Test2 구간을 시각화한 결과는 Figure 5와 같다. 그리고 데이터의 범위를 벗어난 2020년 4월 예측값의 정확도는 실제값의 약 98.86%를 나타냈다. 실제값에 대한 예측값의 정확도는 6C6의 ABCDEF가 약 99.83%로 가장 높게 나타났으나 학습, Test1, Test2에 대한 성능이 6C6의 ABCDEF 보다 우수한 6C3의 ABF 데이터셋을 본 연구에서는 최종 선택하였다.

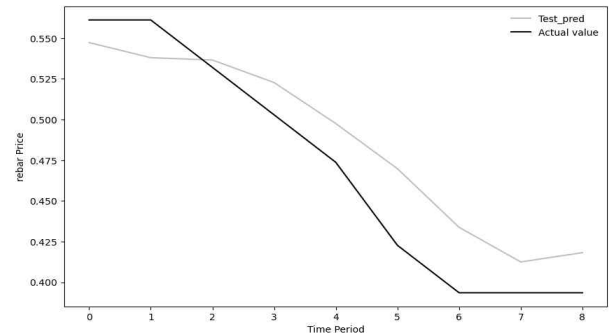


Figure 5. 1-month prediction of test2 with ABF dataset

### 8.3 예측범위 확대를 위한 데이터의 시간 단위 재구성 실험

예측범위를 확대하기 위해 데이터조합 탐색 실험에서 선택된 ABF 데이터셋의 시간 간격을 재구성하여 단기예측한 실험 결과는 Table 7과 같다. 그리고 각각의 데이터의 범위를 벗어난 개월 별 예측값을 시각화한 결과는 Figure 6과 같다.

Table 7. Experiment results of reconstructed data time intervals

Month	Train RMSE	Test1 RMSE	Test2 RMSE	Pred	Actual value
2 (2020.5)	0.03270	0.03298	0.03187	654,000	660,000
3 (2020.6)	0.05498	0.06286	0.06197	674,000	670,000
4 (2020.7)	0.00785	0.01124	0.00983	629,000	660,000

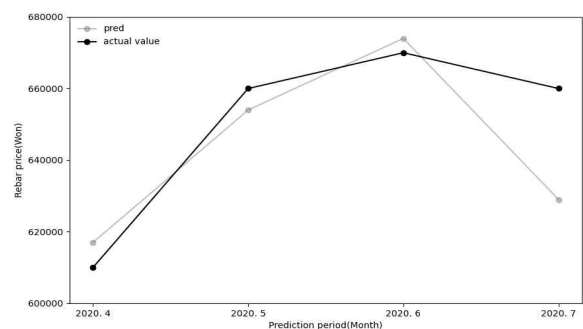


Figure 6. Monthly prediction result

Table 5의 실험 결과 중에 데이터의 시간 간격을 2개월로 재구성한 데이터셋으로 2020년 5월을 예측한 Test2 결과는 RMSE 0.03187로 이 구간을 시각화하면 Figure 7과 같다. 그리고 2020년 5월의 예측값은 약 654,000원으로 실제값에 약 99.09%의 정확도로 예측하였다. 데이터의 시간 간격이 3개월인 데이터셋으로 2020년 6월을 예측한 Test2 결과는 RMSE 0.06197로 이 구간을 시각화하면 Figure 8과 같다. 그리고 2020년 6월의 예측값은 약 674,000원으로 실제값의 약 99.40%를 나타냈다.

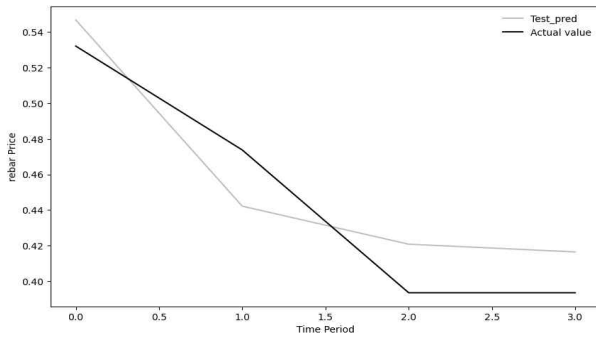


Figure 7. 2-month prediction of test2 with ABF dataset

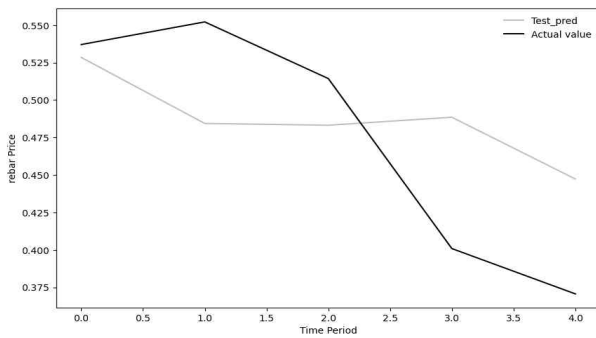


Figure 8. 3-month prediction of test2 with ABF dataset

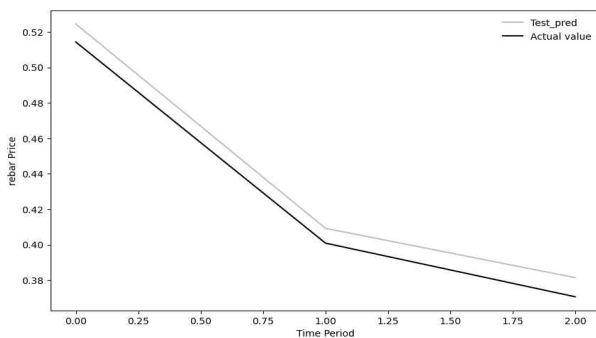


Figure 9. 4-month prediction of test2 with ABF dataset

데이터의 시간 간격이 4개월인 데이터셋으로 2020년 7월을 예측한 Test2 결과는 RMSE 0.00983으로 이 구간을 시각화하면 Figure 9와 같다. 그리고 2020년 7월의 예측값은 약 629,000원으로 실제값의 약 95.30%로 데이터의 범위를 벗어난 예측값의 정확도가 급격히 저하되었다. 이는 데이터의 시간 간격을 재구성하는 과정에서 늘어나는 시간 간격만큼 데이터가 배로 감소하기 때문에 학습성과 Test1 사이에 과적합이 발생하여 예측정확도가 급격히 떨어진 것으로 추정된다. 따라서 본 연구에서 제안한 방식으로 철근 가격의 예측범위를 확대하는 것은 3개월까지를 유효한 것으로 판단된다.

## 9. 결 론

본 연구에서는 딥러닝을 활용한 철근 가격예측의 예측범위를 확대하기 위해 전체 데이터의 시간 간격이 1개월인 데이터셋으로 시간 간격을 2, 3개월 단위로 재구성하여 실험을 진행하였다. 실험 결과 각각의 예측정확도는 약 98.86%, 99.09%, 99.40%로 본 연구에서 제안한 방식의 예측 평균 정확도는 약 99.11%를 나타냈다. 하지만 이 정확도는 좋은 결과를 지나칠 수 있는 Random Search에 단점 때문에 비결정적이며, 탐색 수의 증가와 향후 지속적인 데이터 확보로 예측성능의 추가 개선 여지가 충분하다.

딥러닝을 활용한 철근 가격예측의 예측범위를 확대에 관한 실험적 연구는 현시점에서 3개월 후까지의 철근 가격을 예측함으로써 철근이 실투입되는 시점 이전의 구매적기를 사전에 판단하여 계획된 예산을 초과하지 않는 범위에서 철근에 대한 자재비의 산정이 가능할 것으로 판단된다. 또한 철근 이외의 건축재료를 비롯하여 통계기반 변동성 시계열 데이터로 가격을 예측하는 연구에서 본 연구에서 제시한 내용이 딥러닝의 예측범위를 확대를 위한 기초적 자료로 활용될 수 있을 것이다.

본 연구에서 제시하는 방식은 두 가지 한계점이 있다. 첫 번째로, 예측범위를 확장하기 위해 데이터의 시간 간격을 재구성하게 되면 늘어나는 시간 간격만큼 데이터가 배로 감소하기 때문에 예측범위 확장에 제한적일 수 있다. 두 번째로, 탐색하고자 하는 개월 수 만큼 개별적 탐색이 필요하기 때문에 탐색에 소요되는 시간이 증가한다. 향후 이 두 가지 단점을 보완할 수 있는 방법에 관한 추가적 연구가 필요하다.

## REFERENCES

1. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization, *The Journal of Machine Learning Research*, 13(1), pp. 281-305.
2. Choi, Y., Yim, H., & 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.
3. Choi, M., & 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.
4. Chung, W., Park, G., Gu, Y., Kim, S., & Yoo, S. (2019). City Gas Pipeline Pressure Prediction Model, *Journal of Society for e-Business Studies*, 23(2), pp. 33-47.
5. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research*,

- 270(2), pp. 654-669.
6. Gao, X., Shi, M., Song, X., Zhang, C., & Zhang, H. (2019). Recurrent neural networks for real-time prediction of TBM operating parameters, *Automation in Construction*, 98, pp. 225-235.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory, *Neural computation*, 9(8), pp. 1735-1780.
8. Joo, I., & 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.
9. Kim, J., & Baek, C. (2019) Bivariate long range dependent time series forecasting using deep learning, *The Korean Journal of Applied Statistics*, 32(1), pp. 69-81.
10. Lahari, M. C., Ravi, D. H., & Bharathi, R. (2018). Fuel Price Prediction Using RNN, *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1510-1514. IEEE.
11. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation, In *Proceedings of the 24th international conference on Machine learning*, pp. 473-480.
12. Lee, D., & Kim, K. (2019). Deep Learning Based Prediction Method of Long-term Photovoltaic Power Generation Using Meteorological and Seasonal Information, *Journal of Society for e-Business Studies*, 24(1), pp. 1-16.
13. Lee, J., & Choi, J. (2007). Analysis of the impact of material procurement and procurement management on the process: Focusing on the steel structure project, *Journal of the Architectural Institute of Korea-Structural Section*, 23.12, pp. 141-148.
14. Lee, J., Yoo, J., Kim, C., Lee, G., & Lim, B. (2008). The method of calculating the order point considering the fluctuations in demand for materials at construction sites. *Journal of the Architectural Institute of Korea-Structural System*, 24(10), pp. 117-125.
15. Lee, S., Kim, S., Lee, J., & Han, C. (2006). Proposal for Developed Procurement and Material management System On Using Previous System Analysis in Plant Engineering, *Korean Society for Construction Management Conference*, pp. 204-209.
16. Lee, W., Kim, Y., Kim, J., & Lee, C. (2020). Forecasting of Iron Ore Prices using Machine Learning, *Journal of the Korea Industrial Information Systems Research*, 25(2), pp. 57-72.
17. Liao, T. W. (2005). Clustering of time series data—a survey, *Pattern recognition*, 38(11), pp. 1857-1874.
18. Liu, H., Mi, X., & Li, Y. (2018). Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network, *Energy Conversion and Management*, 166, pp. 120-131.
19. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: a deep learning approach, *IEEE Transactions on Intelligent Transportation Systems*, 16(2), pp. 865-873.
20. Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*, Springer-Verlag, Berlin Heidelberg.
21. Park, S., & Jung, D. (2016). *Basic Study on the Material Market by Specialized Construction Industry*, Seoul: Korea Institute of Construction Policy and Management.
22. Polson, N.G., & Sokolov, V.O. (2017). Deep learning for short-term traffic flow prediction, *Transportation Research Part C: Emerging Technologies*, 79, pp. 1-17.
23. Schuster, M., & Paliwal, K.K. (1997). Bidirectional recurrent neural networks, *IEEE Trans Signal Process*, 45, pp. 2673-2681.
24. Seo, Y., & Yeom, J. (2019). Comparison of LSTM-based Fine Dust Concentration Prediction Method using Meteorology Data, *Korean Society of Surveying, Geodesy, Photogrammetry, and Cartography 2019*, pp. 117-120.
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, 15(1), pp. 1929-1958.
26. Tensorflow.org. (2020) Overfitting and underfitting, accessed Sep 27, 2020, [https://www.tensorflow.org/tutorials/keras/overfit\\_and\\_underfit.stand](https://www.tensorflow.org/tutorials/keras/overfit_and_underfit.stand).

(Received Oct. 14 2020 Revised Nov. 11 2020 Accepted Dec. 12 2020)