IoT 스트리밍 센서 데이터에 기반한 실시간 PM10 농도 예측 LSTM 모델

Real-time PM10 Concentration Prediction LSTM Model based on IoT Streaming Sensor data

저자 (Authors)
김삼근, 오택일
Sam-Keun Kim, Tack-Il Oh

출처 (Source)
한국산학기술학회 논문지 19(11), 2018.11, 310-318(9 pages)

발행처 (Publisher)
한국산학기술학회
Korea Academy Industrial Cooperation Society

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07578397

APA Style
김삼근, 오택일 (2018). IoT 스트리밍 센서 데이터에 기반한 실시간 PM10 농도 예측 LSTM 모델. 한국산학기술학회 논문지, 19(11), 310-318

Journal of the Korea Academia-Industrial cooperation Society
Vol. 19, No. 11 pp. 310-318, 2018

https://doi.org/10.5762/KAIS.2018.19.11.310
ISSN 1975-4701 / eISSN 2288-4688

IoT 스트리밍 센서 데이터에 기반한 실시간 PM10 농도 예측 LSTM 모델

김삼근[1*], 오택일[2]
[1]한경대학교 컴퓨터시스템연구소&컴퓨터공학과, [2]한경대학교 컴퓨터공학과

Real-time PM10 Concentration Prediction LSTM Model based on IoT Streaming Sensor data

Sam-Keun Kim[1*], Tack-Il Oh[2]
[1]**Computer System Institute, Department of Computer Engineering, Hankyong National University**
[2]**Department of Computer Engineering, Hankyong National University**

요 약 최근 사물인터넷(IoT)의 등장으로 인터넷에 연결된 다양한 기기들에 의해 대규모의 데이터가 생성됨에 따라 빅데이터 분석의 중요성이 증가하고 있다. 특히 실시간으로 생성되는 대규모의 IoT 스트리밍 센서 데이터를 분석하여 새로운 의미있는 미래 예측을 통해 다양한 서비스를 제공하는 것이 필요하게 되었다. 본 논문은 AWS를 활용하여 IoT 센서로부터 생성되는 스트리밍 데이터에 기반하여 실시간 실내 PM10 농도 예측 LSTM 모델을 제안한다. 또한 제안 모델에 따른 실시간 실내 PM10 농도 예측 서비스를 구축한다. 논문에 사용된 데이터는 PM10 IoT 센서로부터 24시간 동안 수집된 스트리밍 데이터이다. 이를 LSTM의 입력 데이터로 사용하기 위해 PM10 시계열 데이터로부터 30개의 연속된 값으로 이루어진 시퀀스 데이터로 변환한다. LSTM 모델은 바로 인접한 공간으로 이동해 가는 슬라이딩 윈도우 프로세스를 통하여 학습한다. 또한 모델의 성능 개선을 위해 24시간마다 수집한 스트리밍 데이터에 대해 점진적 학습 방법을 적용한다. 제안한 LSTM 모델의 성능을 평가하기 위해 선형회귀 모델 및 순환형 신경망(RNN) 모델과 비교한다. 실험 결과는 제안한 LSTM 예측 모델이 선형 회귀보다 700%, RNN 모델보다는 140% 성능 개선이 있음을 보여주었다.

Abstract Recently, the importance of big data analysis is increasing as a large amount of data is generated by various devices connected to the Internet with the advent of Internet of Things (IoT). Especially, it is necessary to analyze various large-scale IoT streaming sensor data generated in real time and provide various services through new meaningful prediction. This paper proposes a real-time indoor PM10 concentration prediction LSTM model based on streaming data generated from IoT sensor using AWS. We also construct a real-time indoor PM10 concentration prediction service based on the proposed model. Data used in the paper is streaming data collected from the PM10 IoT sensor for 24 hours. This time series data is converted into sequence data consisting of 30 consecutive values from time series data for use as input data of LSTM. The LSTM model is learned through a sliding window process of moving to the immediately adjacent dataset. In order to improve the performance of the model, incremental learning method is applied to the streaming data collected every 24 hours. The linear regression and recurrent neural networks (RNN) models are compared to evaluate the performance of LSTM model. Experimental results show that the proposed LSTM prediction model has 700% improvement over linear regression and 140% improvement over RNN model for its performance level.

Keywords : Long Short Term Memory, PM10, Prediction model, Recurrent Neural Network, Sequence data

*Corresponding Author : Sam-Keun Kim(Hankyong National Univ.)
Tel: +82-31-670-5163 email: skim@hknu.ac.kr
Received September, 28, 2018 Revised October 2, 2018
Accepted November 2, 2018 Published November 30, 2018

1. 서론

최근, 많은 사물인터넷(IoT: Internet of Things) 애플리케이션들이 헬스 케어, 물류, 스마트 홈, 스마트 시티, 농업, 교육 등의 다양한 분야에서 쏟아져 나오고 있다. 이들 대부분 애플리케이션들의 핵심 요소는 예측을 위한 지능적인 학습 알고리즘이라고 할 수 있다. 본 논문에서는 AWS[1]를 활용하여 실시간으로 생성되는 IoT 스트리밍 센서 데이터를 수집하여 실내의 미세먼지 농도를 실시간으로 예측하는 서비스를 위한 LSTM(Long Short-Term Memory)[2] 기반의 미세먼지 농도 예측 모델을 제안하고, 제안 모델에 따른 실시간 미세먼지 농도 예측 서비스를 구축한다.

미세먼지는 지름 10㎛ 이하인 먼지를 말하며 환경법령에서는 PM10으로 부른다[3]. PM10은 인체의 피부 및 호흡기 질환에 직접적인 원인이 되기도 하며, 세계적인 기후 변화에 따라 점차 증가하는 추세이므로 이에 대한 대처가 필요하다. 이러한 실외의 미세먼지를 예측하기 위해 선형회귀 및 신경망(ANN: artificial neural networks) 모델에 기반한 많은 연구들이 진행되었다[4,5].

PM10은 실외뿐만 아니라 실내에서도 인체에 악영향을 미친다. 따라서 생활공간, 특히 헬스 케어 시설이나 지하철 역내와 같이 사람들이 밀집된 실내 공간에서 PM10의 농도를 모니터링하거나 제어하는 시스템들이 연구되었다[6,7]. 또한 PM10을 제어하기 위한 시스템 구현을 위해 IoT 기술을 적용하는 연구도 있었다[8].

본 논문에서는 LSTM을 이용한 IoT 스트리밍 센서 데이터 기반의 PM10 농도 예측 모델을 제안한다. 사용된 데이터는 AWS IoT를 이용하여 IoT 환경에서 PM10 스트리밍 센서 데이터를 수집한 것이다. 이렇게 수집한 시계열(time series)데이터를 LSTM 모델의 입력으로 사용할 수 있도록 시퀀스(sequence) 데이터로 변환시킨다. 변환된 시퀀스 데이터를 이용하여 LSTM 모델을 학습시키고, 학습된 모델을 이용하여 PM10 농도 예측을 수행한다. 순환형 신경망(RNN: Recurrent Neural Networks)[9]은 긴 시퀀스를 학습하는데 있어서 여러 시간 스텝을 지나가면서 기억을 못하는 기울기 소실 문제(vanishing gradient problem)[10]가 발생하여 데이터의 장기 의존성(long term dependency)을 반영하지 못한다는 문제점을 안고 있다. LSTM은 RNN의 이러한 단점을 극복한 모델이다.

PM10 스트리밍 데이터를 LSTM의 입력으로 사용하기 위해 먼저 1초 단위로 측정한 값들을 20초 단위 평균값으로 계산하고, 이를 30개의 연속된 값으로 구성된 시퀀스 데이터로 변환시킨다. 이렇게 변환된 시퀀스 데이터는 LSTM의 입력 데이터로 사용되며, 출력 데이터는 다음 20초 후의 PM10 농도를 예측한다.

LSTM 모델은 슬라이딩 윈도우(sliding window) 프로세스에 기반하여 학습한다. 슬라이딩 윈도우는 바로 인접한 데이터 셋으로의 이동 과정을 말한다. 활성함수로서 tanh[11]가 학습에 사용되고, 모델의 성능은 Adam 최적화 함수[12]에 의해 최적화된다. 본 논문에서는 LSTM 예측 모델의 성능을 측정하기 위해 RMSE(Root Mean Square Error)[13]를 사용한다. RMSE는 실제 값과 학습된 모델의 예측 값 사이의 차이를 의미한다. 모델의 성능은 RMSE가 0에 가까이 갈수록 좋다. 따라서 RMSE는 기존 연구 모델들과의 성능 비교에 자주 사용된다. 본 논문에서 제안한 LSTM 모델을 선형회귀 모델 및 RNN 신경망 모델과 비교한다. 실험 결과는 제안한 LSTM 예측 모델이 선형 회귀와 RNN 모델보다 훨씬 효율적임을 보여주었다.

본 논문의 구성은 다음과 같다: 2장에서는 관련연구를 분석하고, 3장에서는 사용된 예측 모델을 기술한다. 4장은 데이터 셋을 설명하며, 5장에서는 제안한 LSTM을 이용한 PM10 농도 예측 모델을 기술한다. 6장에서는 예측 모델의 평가를 수행하고, 7장에서는 본 논문의 결론 및 향후 연구과제에 대해 기술한다.

2. 관련연구

2.1 PM10

미세먼지는 TSP, PM10, PM2.5로 분류된다. PM10은 장시간 동안 공기 중에 부유하는 먼지 입자의 지름이 10㎛ 이하인 것을 말한다. 미세먼지는 기후 환경 변화의 영향으로 지속적으로 증가하고 있는 추세이다. PM10은 또한 스모그나 황사에 기인하여 급격히 증가할 수 있다[14].

PM10은 인체의 피부 및 호흡기 질환에 직접적인 원인이 되기도 한다. 이에 따라 WHO[15]는 PM10 가이드라인을 만들어 배포하였고, 아울러 미국에서는 사람들의 건강을 보호하기 위한 가이드라인을 제시하고 클린 에어

환경을 조성하였다[16].

2.2 PM10 제어 시스템

미세먼지는 실내에서도 건강에 악영향을 끼친다. 따라서 우리가 생활하고 있는 실내에서 PM10의 농도를 측정하여 이를 효과적으로 제어하는 시스템이 필요하다. [8]은 PM10을 제어하기 위한 효율적인 시스템을 구현하기 위해 IoT 기술을 적용하였다. 이 연구에서는 PM10의 농도 변화를 모니터링 하다가 PM10 농도가 $80\mu g/m^3$ 이상이 되면 FAN을 구동시키는 시스템을 제안하였다. 환경부에서는 PM10 농도가 $81\mu g/m^3$ 이상일 경우부터 '미세먼지 나쁨 구간'의 시작으로 분류하고 있다.

2.3 PM10 예측 모델

[17]은 Feed-Forward ANN, 선형회귀, 클러스터링 알고리즘을 이용하여 PM10을 예측하는 연구를 수행하였다. 사용된 데이터는 핀란드 헬싱키의 2003년 9월 1일부터 11월 31일까지의 시계열 데이터이다. 시계열 데이터의 특징들로는 PM10, O_3, 상대 습도, 바람의 세기 및 방향 등이 있다. [18]은 ANN과 SVM(Support Vector Machines)을 이용하여 홍콩 시내 도로변의 PM10 농도를 예측하였다. [19]에서는 SVM을 이용하여 2010-2012년 기간 동안에 북경에서 측정된 다양한 기상 인자들(기압, 상대 습도, 온도, 풍속)에 기반하여 PM10 및 PM2.5를 예측하는 모델을 제안하였다.

많은 기존 연구들이 선형회귀 및 기계학습의 ANN 모델을 이용하여 실외의 장기 PM10 시계열 데이터 기반의 예측 모델을 제안했다. 그러나 IoT 스트리밍 데이터에 기반하여 실내의 PM10 농도를 실시간으로 예측하기 위한 LSTM 모델에 관한 연구는 활발하게 진행되지 않았다. 본 논문에서는 클라우드 환경에서 IoT 스트리밍 센서 데이터에 기반한 실내 PM10 농도 예측 LSTM 모델을 제안한다.

3. 예측 모델

3.1 Linear Regression

선형회귀는 변수들 사이의 관계를 예측하는데 사용되는 통계적 방법이다. 다양한 형태의 선형회귀 모델이 있다. 단순 선형회귀 모델은 단일 독립변수를 사용하지만, 다중 선형회귀 모델은 여러 개의 독립 변수를 사용한다.

일반적으로 선형회귀 모델은 식 (1)처럼 단순히 입력 특징 값들의 가중치 합을 계산함으로써 예측을 수행한다.

$$\hat{y}=h_\theta(\vec{x})=\theta^T \cdot \vec{x} \qquad (1)$$

여기서 θ는 모델의 파라미터 벡터이고, θ^T는 θ의 치환(transpose), $\vec{x}$는 인스턴스의 특징 벡터, $\theta^T \cdot \vec{x}$는 θ^T와 $\vec{x}$ 벡터의 내적, h_θ는 모델 파라미터 θ를 이용한 가설 함수이다.

3.2 순환형 신경망(RNN)

RNN은 순차(sequential) 정보를 처리하는데 사용된다. RNN은 은닉층의 값을 ANN 내부의 메모리에 기억한 후 다음 순서의 입력 데이터로 학습할 때 이용한다.

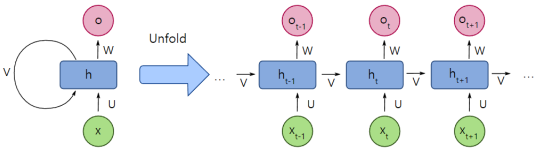

Fig. 1. RNN Architecture [9]

Fig. 1은 일반적인 RNN 구조이다. x_t는 시간 스텝에서의 입력 값으로 메모리 부분의 h_t를 거쳐 다음 스텝에 반영되어 계산된다. 즉 이전 스텝의 값과 현재 시간 스텝의 입력 값에 의해 계산된다. o_t는 시간 스텝 t에서의 출력 값으로, 매 시간 스텝마다 출력 값이 나온다. RNN은 음성 인식, 번역, 이미지 캡셔닝(captioning) 등에 사용되며, LSTM, GRU(Gated Recurrent Unit)[20] 모델 등으로 분류된다.

RNN의 학습 방법인 경사하강법(gradient descent)은 미분 값인 기울기 변화량을 이용하는데 이 변화량이 매우 작다면 기울기가 거의 0이 되고 이전 층의 모든 기울기들도 0이 된다. 따라서 여러 시간 스텝이 지날수록 주요 정보를 기억하지 못하게 되고, 결국 먼 과거의 상태는 현재 스텝의 학습에 도움을 주지 못하게 되어 학습이 더 이상 진행되지 못하는 문제가 발생한다.

식 (2)는 RNN에 대한 수학공식이다.

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y) \quad (2)$$

여기서 x_t는 입력 벡터, h_t는 은닉층 벡터, y_t는 출력 벡터, W, U, b는 파라메타 행렬과 벡터, σ_h, σ_y는 활성화 함수이다.

3.3 Long Short Term Memory(LSTM)

RNN의 강점은 현재의 작업을 이전 정보와 연결할 수 있다는 것이다. 그러나 RNN은 학습과정에서 긴 시퀀스를 처리하는데 주요 정보가 여러 시간 스텝을 지나가면서 소실되는 기울기 소실 문제가 발생하여 데이터의 장기 의존성(long-term dependency)을 고려하는데 한계가 있다. 따라서 RNN의 장기 의존성 문제를 해결하기 위해 LSTM이 연구되어 왔다.

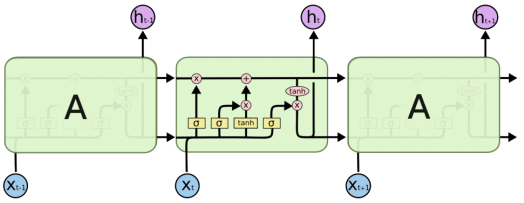

Fig. 2. LSTM Architecture [2]

LSTM은 Fig. 2에서처럼 재귀적 구조를 가진 4개의 층으로 구성되어 있다. LSTM의 코어는 게이트(gate)를 통해 들어오는 연속된 셀 상태이다. 연속된 셀 상태는 컨베이어 벨트라고도 한다. 컨베이어 벨트를 통해 들어오는 정보는 변경 없이 전달된다. LSTM은 입력 게이트, Forget 게이트, 출력 게이트를 통해 정보를 추가하거나 삭제할 수 있다. 결론적으로, 게이트는 정보를 선택적으로 전달하는 역할을 하고, 이를 이용하여 이전 데이터를 제거함으로써 학습을 계속한다.

식 (3)은 LSTM의 수학 공식이다. LSTM은 게이트 벡터를 이용하여 계산한다. f_t는 Forget 게이트 벡터로서 이전 셀 상태를 기억하는 가중치 역할을 한다. i_t는 입력 게이트 벡터로서 새로운 정보를 획득하는 가중치 역할을 한다. 반면에, o_t는 출력 게이트 벡터이며, 출력 후보를 선택하는 역할을 한다. x_t는 입력 벡터이고, h_t는 출력 벡터이다. c_t는 셀 상태 벡터이고, W, U, b는 각각 파라메타 행렬 및 벡터이다. f_t, i_t, o_t는 게이트 벡터들이다. LSTM에서 2 종류의 활성화 함수가 사용된다. σ는 Sigmoid 함수이고, tanh는 하이퍼볼릭 탄젠트 함수이다.

$$f_t = \sigma(W_f \bullet [h_{t-1,} x_t] + b_f)$$
$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)$$
$$\widetilde{C_t} = \tanh(W_C \bullet [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t) \quad (3)\ [2]$$

4. 데이터 셋

본 논문에서는 LSTM 예측 모델의 입력으로 사용하기 위한 데이터를 AWS IoT 환경에서 IoT 센서로부터 PM10 스트리밍 데이터를 수집한다. Fig. 3은 실시간으로 축적되는 데이터 처리를 위해 IoT 기기에서 측정한 PM10 스트리밍 데이터를 AWS IoT와 자체 제작한 Thing-server(소켓 서버)를 통해 Amazon S3[1]에 저장하는 아키텍처를 보여준다.

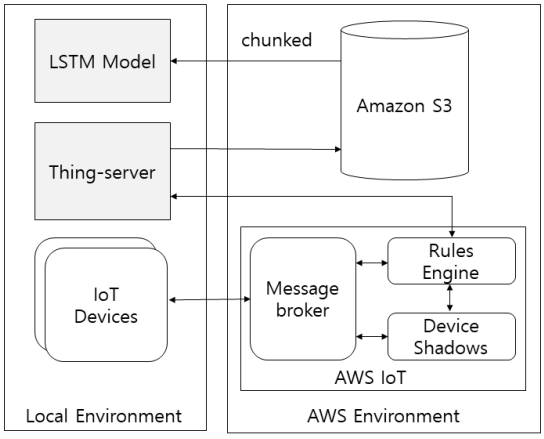

Fig. 3. Training data acquisition from AWS IoT

Fig. 3에서 IoT 기기는 실시간 측정값을 Thing-server로 전송하고, Thing-server는 각 클라이언트에서 전송한 1초마다 측정한 값을 20초 단위의 평균값으로 계산하고 매일 24시간 분량의 파일로 만들어서 매일 자정(00시 00분 00초)에 CSV 형태로 Amazon S3 버킷에 저장한다. S3에 저장된 데이터는 시계열 데이터이므로

로컬 환경에 구성된 LSTM 모델의 입력으로 사용될 수 있도록 시퀀스(sequence) 데이터로 변환된다. 변환된 시퀀스 데이터는 슬라이딩 윈도우 프로세스를 통해 다양한 훈련 데이터 셋을 만드는데 사용될 수 있다.

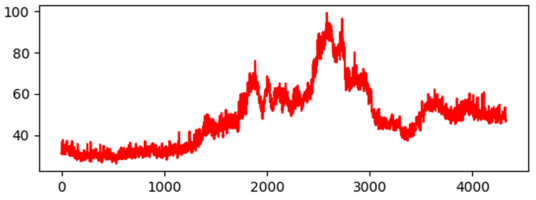

Fig. 4. PM10 raw data

Fig. 4는 테스트를 위해 일정 크기의 생활공간에 설치한 IoT 센서로부터 하루 24시간 동안 측정한 PM10 스트리밍 데이터를 보여준다. 각 측정값은 기기로부터 1초마다 20초 단위의 평균값을 계산한 것이다. LSTM의 입력으로 사용하기 위해서는 시퀀스 데이터로 변환해야 한다.

5. LSTM 기반 PM10 농도 예측 모델

5.1 시퀀스 데이터 생성

LSTM은 시퀀스 데이터를 이용하여 데이터를 학습한다. 따라서 IoT 시계열 데이터 스트림을 LSTM의 입력 데이터로 사용할 수 있도록 시퀀스 데이터로 변환시켜야 한다. 본 논문에서는 슬라이딩 윈도우 방법을 이용하여 시계열 데이터 스트림을 시퀀스 데이터 셋으로 변환시킨다.

슬라이딩 윈도우 방법을 이용하여 시계열 스트리밍 데이터를 LSTM 모델의 입력으로 사용되는 시퀀스 데이터로 변환할 수 있다. N개의 시계열 데이터 스트림을 $X=<x_1,...,x_i,...,x_N>$이라 하자. 여기서 x_i는 1초마다 센서로부터 들어오는 값들을 20초 단위의 평균값으로 계산한 값이다. 윈도우 길이를 l, 윈도우 시프트(shift)를 r이라고 하면, 시간 i일 때 l개의 연속된 값을 갖는 시퀀스를 $s_i=<x_i,...,x_{i+l-1}>\ (i=1,...,N-l+1)$로 정의할 수 있다. 두 번째 시퀀스부터는 $s_{i+r}=<x_{i+r},...,x_{i+r+l-1}>$로 정의한다. 본 논문에서는 이와 같은 변환 규칙에 의해 스트리밍 데이터 셋(X)을 훈련 데이터 셋(시퀀스 집합)인 $S=\{s_1,...,s_{N-l+1}\}$로 변환시켰다. 변환된 전체 데이터 셋(S)에서 80%의 훈련 데이터 셋을 LSTM 모델의 입력으로 사용하여 학습시키고, 20% 테스트 셋으로 LSTM의 일반화 성능을 평가하였다.

5.2 PM10 예측 LSTM 모델

Fig. 5는 PM10 예측 LSTM 모델의 전체 흐름도를 보여준다. LSTM은 RNN에 기반하고 있으며, 많은 게이트들이 연결된 셀들로 구성된다. 시퀀스 데이터가 입력 게이트를 통해 들어오면 mini-batch를 사용하여 반복적으로 학습한다. 첫 단계로 Forget 게이트에서는 셀 상태를 기억할지 잊을지를 결정한다. 출력 게이트에서는 우선 셀 상태에서 어떤 부분들을 출력할지 결정하기 위해 Sigmoid 층을 동작시킨 후 -1과 1 사이의 값을 갖도록 tanh 유닛을 넣어 다음 시간 스텝의 예측값을 출력한다.

RNN은 과거의 숨겨진 값으로부터 예측값을 계산하는데, 오래된 과거의 값은 기울기 값을 계산하기 어렵게 하는 기울기 소실 문제를 겪는다. LSTM의 Forget 게이트는 단지 부분 데이터만 사용되도록 함으로써 이러한 기울기 소실 문제를 해결한다.

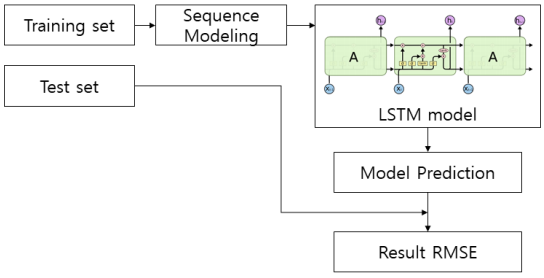

Fig. 5. Flow chart of the proposed PM10 prediction LSTM model

Table 1. LSTM_with_peepholes Parameter Settings: sliding_ window=30, window_shift=1

Index	Hidden node	Batch Size	Epoch	MSE	RMSE
1	**300**	**30**	**30**	**4.49**	**2.01**
2	300	30	50	5.42	2.26
3	300	30	70	11.70	3.15
4	300	30	90	6.67	2.50
5	300	30	110	7.93	2.75

Index	Hidden node	Batch Size	Epoch	MSE	RMSE
1	100	30	30	5.78	2.38
2	200	30	30	13.30	3.57
3	200	7	30	12.10	3.18
4	**300**	**30**	**30**	**4.49**	**2.01**
5	300	7	30	35.00	5.54

RMSE 값은 Table 1의 파라메타 리스트의 값들을 변경하면서 측정하였다. Table 1은 20초 단위 평균값의 30개의 연속된 값들을 갖는 시퀀스들로 구성된 훈련 셋으로 학습시켰다.

LSTM 모델을 최적화하기 위해 Adam 최적화 함수를 사용한다. 이 함수는 보통 Non-stationary 시계열 데이터를 최적화하는데 사용된다. Non- stationary는 시계열 데이터가 어디로 흘러가는지 알지 못하는 경우(예: stock market)를 의미한다. 이 함수는 사용자가 임의로 파라메타를 설정한다 해도 최적화를 위해 학습률뿐만 아니라 모멘텀까지 스스로 조정한다.

본 논문에서는 모델의 예측 성능을 개선하기 위해 점진적 학습(incremental learning)[21] 방법을 적용하였다. 최초 모델 학습에서는 하루 24시간 동안 수집한 PM10 스트리밍 데이터를 이용하여 학습시키고, 다음 24시간 동안 수집한 데이터에서부터는 점진적 학습 방법을 반복적으로 적용시킨다. 즉 매일 00:00:00초부터 시작해서 다음 날 23:59:59초까지 측정된 약 4,320개의 스트리밍 데이터를 하나의 CSV 파일로 저장하고, 이 파일을 Amazon S3의 Training bucket에 업로드한다. 전날까지 학습된 모델 파일은 TensorFlow Serving에서 사용되고, 다음 날 마지막으로 저장된 모델 파일이 점진적 학습을 위해 사용된다.

5.3 실시간 예측 서비스

Fig. 6은 AWS IoT[1]에서 실시간으로 생성되는 IoT 스트리밍 데이터의 실시간 예측 서비스에 대한 플로차트를 보여준다. 서비스 아키텍처는 기기로부터 스트리밍 데이터를 받아 들이는 AWS IoT, 훈련 데이터 저장을 위한 Amazon S3 저장장치, Local 컴퓨팅 환경에서의 학습을 위한 LSTM 모델, 모델을 배치하기 위한 Docker[22], 실시간 예측 서비스를 제공하기 위해 Docker상에 설치된 TensorFlow Serving[23], 애플리케이션과 TensorFlow Serving 사이의 통신을 담당하는 Thing-server(소켓 서버)로 구성하였다.

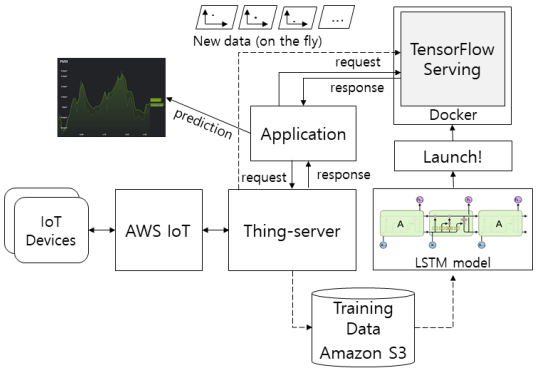

Fig. 6. Flow chart of real-time PM10 prediction service

Docker는 일종의 가상머신 프로그램으로, 여기서 사용되는 가상머신 이미지는 TensorFlow Serving 프로그램이 설치되어 있는 이미지 파일이다. 이 가상머신을 Docker에서 학습된 모델들을 인자로 주어 실행시키면 TensorFlow Serving REST API 서버가 실행된다. 이 서버에 입력 값을 요청하게 되면 예측값을 얻을 수 있다. 이 TensorFlow Serving은 Thing-server와 통신할 수 있다. Thing-server에서 기기 데이터와 함께 예측값을 요청하면, TensorFlow Serving이 응답을 Thing-server에게 보내주고, Thing-server는 이 값을 요청 애플리케이션에게 보낸다. 해당 애플리케이션에서는 FAN을 ON시키는 것과 같은 다양한 서비스를 적용할 수 있다.

본 논문에서 사용한 툴은 Python으로 구축된 딥 러닝 툴인 TensorFlow[24]이다.

6. 실험 및 평가

이 장에서는 실험에 사용한 성능평가 알고리즘 및 훈련 데이터 셋, 테스트 셋을 이용한 다양한 실험결과를 기술하고, 선형회귀, RNN, LSTM 모델의 성능 비교를 수행한다. 성능 평가에 사용한 알고리즘은 MSE(Mean Square Error)와 RMSE (Root Mean Square Error)이다.

PM10 예측에 사용한 전체 데이터 개수는 4,320개이다. 훈련 데이터 셋과 테스트 데이터 셋은 반복된 시퀀스 데이터로 구성하였다. 훈련 셋은 전체 데이터의 80%인 3,456개를 사용하였고, 테스트 셋은 나머지 20%인 864개를 사용하였다.

평가 알고리즘으로 MSE와 RMSE를 사용하였다. 이들은 모델에 의해 생성된 예측 데이터와 실측 데이터 사

이의 차이를 보여주는 좋은 방법이다. RMSE의 경우, 값이 0에 가까우면 더 좋은 성능을 의미한다. 훈련 데이터 셋 $\mathbb{X}$에 선형회귀 가설 h의 RMSE는 식 (4)와 같다.

$$RMSE(\mathbb{X},h)=\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\vec{x}^{(i)}\right)-y^{(i)}\right)^2} \quad (4)$$

여기서 m은 인스턴스 개수, $\vec{x}^{(i)}$는 i번째 인스턴스의 전체 특징 벡터(feature vector), $\mathbb{X}$는 데이터 셋의 모든 인스턴스의 모든 특징 값을 포함하는 행렬, h는 시스템의 예측함수, 그리고 $RMSE(\mathbb{X},h)$는 가설 h에 이용하여 예제 집합에서 측정된 비용함수(cost function)이다.

성능 비교에 사용한 모델들은 선형회귀, RNN, LSTM, LSTM_with_peepholes, GRU이다. Table 2의 LSTM과 GRU 결과는 Adam 최적화 함수를 적용한 것이다. 본 논문에서는 PM10을 실시간으로 예측하기 위해 LSTM 모델을 사용하였고, 선형회귀와 RNN과의 비교를 통하여 성능을 평가하였다.

Table 2. Model Evaluation

	MSE	RMSE
Linear Regression	232.68	14.32
RNN	9.92	2.88
LSTM(Adam)	5.01	2.17
LSTM_with_peepholes(Adam)	**4.49**	**2.01**
GRU	7.86	2.65

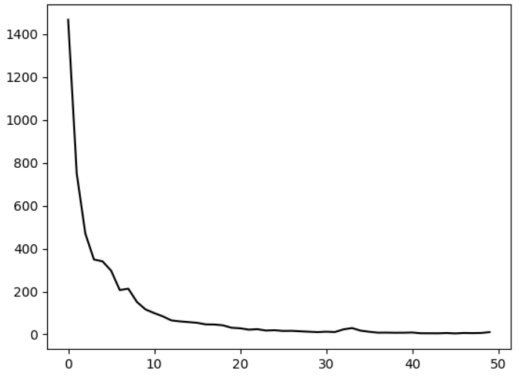

Fig. 7. LSTM loss

Fig. 7은 RMSE 값의 변화량을 그래프로 표현한 것이다. 이 그래프를 보통 LSTM loss라고 한다. 이 그래프는 LSTM 학습에서 훈련 데이터 셋으로 학습된 모델을 테스트 셋에 적용할 때 발생하는 에러 값을 보여준다. 여기서 loss를 반복적으로 학습시킴으로써 LSTM 모델의 RMSE 값을 최적화시킨다.

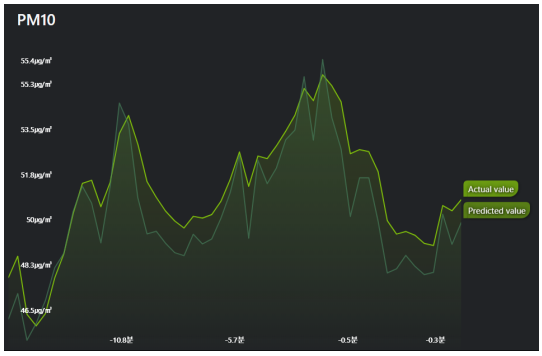

Fig. 8. Example of PM10 actual/prediction graphs

Fig. 8은 실시간으로 측정되는 PM10 실측 그래프와 제안한 PM10 LSTM 예측 모델에 의한 예측 그래프를 함께 보여준다.

7. 결론

본 논문에서는 실내의 PM10 실시간 예측 서비스를 위한 LSTM 기반의 PM10 예측 모델을 제안하고, AWS를 활용하여 IoT 스트리밍 센서 데이터를 수집하여 제안 모델에 따른 실시간 IoT 센서 데이터 예측 분석 서비스를 구축하였다.

제안한 모델은 실시간으로 생성되는 IoT 스트리밍 센서 데이터의 특성에 따른 실시간 예측 서비스를 위해 기계학습 및 ANN 모델들을 생산 환경에 사용할 수 있도록 구현된 TensorFlow Serving을 사용하였다. 또한 다양한 알고리즘 중에서 스트리밍 시계열 데이터에 알맞은 LSTM 모델을 이용하여 IoT 센서 데이터를 학습하여 입력 값들을 분석하고, 애플리케이션에서 FAN을 ON시키는 것과 같은 환경요소에 따른 필요한 정보를 제공할 수 있는 예측 서비스를 구축하였다.

향후에는 PM10 스트리밍 데이터는 물론 PM2.5, 실내외 온습도, 실외의 PM10/PM2.5 등의 다양한 인자들을 추가하여 PM10뿐만 아니라 PM2.5도 예측하는 것을 연구할 것이다.

References

[1] Amazon Web Service, https://docs.aws.amazon.com/

[2] Christopher Olah, Understanding LSTM Networks, 2015, Available From: https://www.cse.iitk.ac.in/users/sigml/lec/Slides/LSTM.pdf.

[3] Korea Environment Corporation(KECO), Available From: http://www.airkorea.or.kr/dictionary_3. (accessed Sept., 12, 2018)

[4] M. S. Souza, P. Coelho, A. da Silva, A. Pozza, "Using Ensembles of Artificial Neural Networks to Improve PM10 Forecasts", *Chemical Engineering Transactions*, Vol. 43, pp. 2161-2166, 2015.
DOI: https://doi.org/10.3303/CET1543361

[5] B. Oancea, S. C. Ciucu, "Time series forecasting using neural networks", *Proceedings of the CKS 2013 International Conference*, pp. 1402-1408, 2014, Available From: https://arxiv.org/ftp/arxiv/papers/1401/1401.1333.pdf.

[6] Yanchen Liua, Zhe Wangc, Zhongchen Zhang, Jiajie Hong, Borong Lin, "Investigation on the Indoor Environment Quality of health care facilities in China", *Building and Environment 141*, pp. 273–287, 2018,
DOI: https://doi.org/10.1016/j.buildenv.2018.05.054

[7] Gyu-Sik Kim, Youn-Suk Son, Jai-Hyo Lee, In-Won Kim, Jo-Chun Kim, Joon-Tae Oh, Hiesik Kim, "Air Pollution Monitoring and Control System for Subway Stations Using Environmental Sensors", *Hindawi Publishing Corporation Journal of Sensors*, Vol. 2016.
DOI: https://doi.org/10.1155/2016/1865614

[8] Jin-Ho Noh, Han-Ho Tack, "The Implementation of the Fine Dust Measuring System based on Internet of Things(IoT)", *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 21, No. 4, pp. 829-835, 2017,
DOI: http://www.dbpia.co.kr/Article/NODE07158660

[9] Recurrent neural network, Wikipedia, 2018, https://en.wikipedia.org/wiki/Recurrent_neural_network

[10] Xavier Glorot, Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, PMLR 9, pp. 249-256, 2010, Available From: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.2059&rep=rep1&type=pdf

[11] Hyperbolic function, Wikipedia, 2018, https://en.wikipedia.org/wiki/Hyperbolic_function

[12] AdamOptimizer, https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer

[13] Root-mean-square deviation, Wikipedia, 2018, Available From: https://en.wikipedia.org/wiki/Root-mean-square_deviation.

[14] D. Dunea, S. Iordache, C. Ianache, "Relationship between airborne particulate matter and weather conditions in targoviste urban area during cold months", Roumanian Journal of Chemistry, Vol. 60, pp. 595-601, 2015, Available From: https://www.researchgate.net/publication/284735048

[15] WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide, 2005, Available From: http://apps.who.int/iris/bitstream/handle/10665/69477/WHO_SDE_PHE_OEH_06.02_eng.pd

[16] NAAQS Table, United States Environmental Protection Agency, 2016, Available From: https://www.epa.gov/criteria-air-pollutants/naaqs-table

[17] D. Vlachogiannis, A. Sfetsos, "Time series forecasting of hourly PM10 values: model intercomparison and the development of localized linear approaches", *WIT Transactions on Ecology and the Environment*, Vol. 86, pp. 85-94, 2006
DOI: https://doi.org/10.2495/AIR06009

[18] Hazrul Abdul Hamid, Ahmad Shukri Yahaya, Nor Azam Ramli, Ahmad Zia Ul-Saufie and Mohd Norazam Yasin, "Short Term Prediction of PM10 Concentrations Using Seasonal Time Series Analysis", MATEC Web of Conferences, Vol. 47, 2016.
DOI: https://doi.org/10.1051/matecconf/20164705001

[19] H. Weizhen, L. Zhengqiang, Z. Yuhuan, X. Hua, Z. Ying, L. Kaitao, L. Donghui, W. Peng, M. Yan, "Using support vector regression to predict PM10 and PM2.5", I*n IOP Conference Series: Earth and Environmental Science*, Vol. 17, 2014.
DOI: https://doi.org/10.1088/1755-1315/17/1/012268

[20] J. Chung, C. Gulcehre, K. H. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *Neural Information Processing Systems*, 2014, Available From: https://arxiv.org/pdf/1412.3555.pdf

[21] Incremental learning, Wikipedia, 2018, https://en.wikipedia.org/wiki/Incremental_learning/

[22] Docker, https://www.docker.com/

[23] TensorFlow Serving, https://www.tensorflow.org/serving/

[24] TensorFlow, https://www.tensorflow.org/

김 삼 근(Sam-Keun Kim) [종신회원]

- 1988년 2월 : 숭실대학교 대학원 전자계산학과 (공학석사)
- 1998년 2월 : 숭실대학교 대학원 전자계산학과 (공학박사)
- 1992년 3월 ~ 현재 : 한경대학교 컴퓨터공학과 교수

<관심분야>
인공신경망, 데이터마이닝, 기계학습, IoT

오 택 일(Tack-Il Oh) [준회원]

• 2012년 3월 ~ 현재 : 한경대학교 컴퓨터공학과 학사과정

<관심분야>
인공지능 주도 개발, 기계학습, IoT