

Abstract

사물 인터넷 환경에서 각종 센서들을 통해 얻은 환경에 대한 정보들은 사용자의 신체로부터 직접적으로 수집된다는 점에서 개인정보 침해 등 여러 불편함이 있을 수 있다. 이에 좀 더 간접적인 방법으로 사용자 주변 상황에 대한 정보를 수집해 행동 인식에 사용해볼 수 있다. 이때, 행동 인식은 특징을 추출하고, 분류하는 과정으로 구성되며, 기존 연구는 이를 위한 알고리즘으로 KNN, SVM 을 적용하기도 했지만 한계가 존재한다. 이에 CNN, LSTM 과 같은 딥러닝 신경망 기술이 적용할 수도 있으나, 이 연구에서는 데이터의 특성이나 시간적 비용 측면에서 더 적합한 Convolutional LSTM (CNN + LSTM) 모델을 사용한다.

1. 서론

헬스케어, 이상 행동 탐지 등에 유용한 기술로 사용되는 상황 인지 기술의 핵심은 사용자의 행동, 상태를 정확하게 인식하는 것이다. 이때, 사용자 환경에 대한 정보를 수집하는 도구로써, 스마트폰, 웨어러블 기기와 같이 사용자가 직접 소지해야 하는 형태가 아닌, 센서를 통해 간접적으로 측정하는 방법을 선택한다.

또한, 행동을 정확하게 분류하려면 입력데이터로부터 효율적인 특징 추출이 필수적이다. 최적의 알고리즘 검정을 위해, 이 연구에서는 테스트베드를 직접 구축했다고 한다. 이렇게 구축한 정보들은 채널이 다양하기 때문에 회귀와 같은 선형적 방법으로는 분류가 어렵고, CNN, LSTM과 같이 다채널 분석에 적합한 알고리즘을 사용해야 하는데, 이 연구에서는 이 둘을 앙상블한 Convolutional-LSTM 모델을 사용한다.

2. 배경 및 관련 연구

2.1 사물 인터넷

사람과 사물, 사물과 사물 간 상호 소통, 지능적 서비스 제공하는 인프라; IoT (Internet of Things). 헬스케어, 에너지 분야에서 많이 활용된다. 특히, 이 연구에서는 Thingspeak라는 사물인터넷 오픈 플랫폼과 연동해 웹 기반 데이터 수집 및 실시간 데이터 시각화를 제공한다고 한다.

2.2 행동 인식

최근 센싱, 실시간 네트워크 기술의 발달로 행동 인식에 사용되는 데이터를 충분히 확보할 수 있게 돼, 이러한 정보들을 바탕으로 인간의 행동을 인식하는 연구가 한창이다. 이때, 적절한 행동 인식을 위해서는 입력 데이터로부터 특징을 효율적으로 잡아낼 수 있어야 하기 때문에 전체 데이터 중 특징이 드러나는 부분만 분할하는 작업을 거친다. 분할 방법으로는 고정 길이 슬라이딩 윈도우 (Sliding window)를 적용한다. 한편, 기존 UCI, WISDM과 같은 오픈 데이터셋을 이용한 연구에서는 가변적 길이로 분할을 하기도 했는데, 데이터의 특성에 따라 적절하게 분할하는 것이 정확도 향상 측면에서 중요하다. 이 연구에서는 직접 구축한 테스트베드를 사용하고, 기존 고정길이 분할의 한계를 해결한 제안을 한다.

2.3 딥러닝 기반 행동 양식

기존 행동 인식 분류 방식이었던 rule based reasoning, probabilistic model, Support Vector Machine 과 달리, 범용적으로 사용할 수 있는 딥러닝 모델을 사용한다. 여기서 이미지 데이터를 다루는데 최적화된 CNN은 3차원 구조의 입력값을 받고, 클래스 점수로 이루어진 하나의 점수 벡터를 출력한다. 이러한 특징으로, CNN은 다채널 데이터를 학습할 수 있다. 한편, RNN은 현재 문제에 이전 정보를 활용하는 연속적인 특징을 갖는다. 이에 연속성이 있는 데이터를 학습하는데 용이하지만, 데이터 간 거리가 먼 경우 역전파 기울기 감소로 장기 의존성 문제가 발생한다.

이러한 문제에 대해 LSTM 모델은 셀 스테이트 (Cell State)를 추가해 이전 셀에서의 정보들을 보존할 수 있다. (포갠 게이트에 의해 기억 정도 결정) 이 연구에서는 CNN과 LSTM을 앙상블 한 Convolutional-LSTM 신경망을 사용한다.

3. 제안하는 실내 행동 인식 시스템

3.1 전체 구조

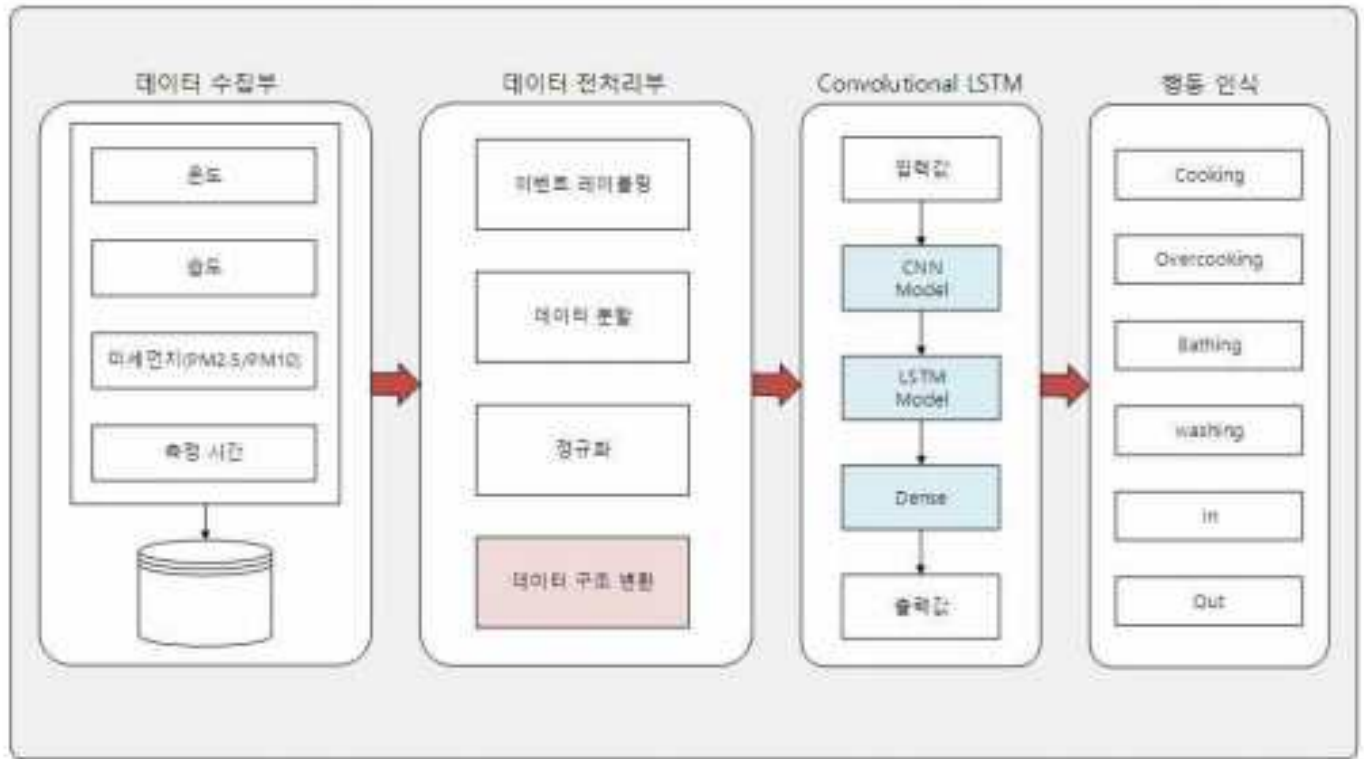


그림 1 제안하는 실내 행동 인식 시스템의 구조도

데이터 수집부: (실수형) 온도, 습도, (정수형) 미세먼지 농도, 측정 시간 -> 데이터베이스에 저장

데이터 전처리부: Convolutional-LSTM의 입력값으로 사용하기 위한 정형화 (슬라이딩 윈도우 기반 분할), 정규화 (측정 단위 차이), 차원 전환

Convolutional LSTM: 히든 레이어 개수, 학습 횟수, 학습률 등의 매개변수를 바꿔가며 최적의 설정을 찾음

3.2 데이터 전처리

3.2.1 행동 수준 정의

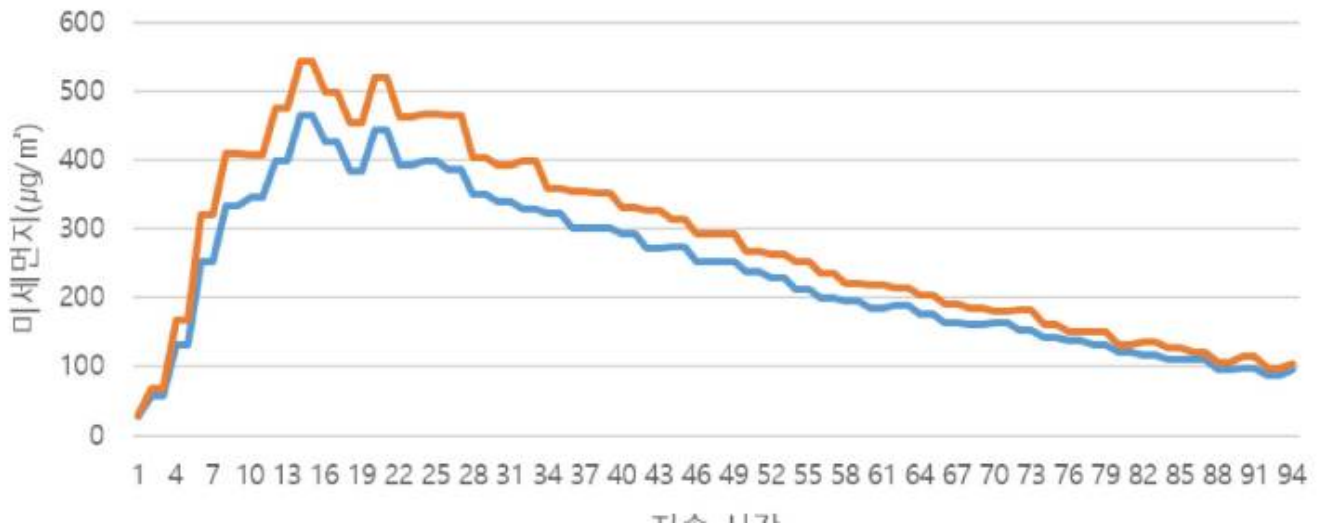
표 1 행동 인식에 사용할 행동 수준 정의

| 행동 수준 | 관측 행동 | 설명 |
|-------|-------------|-------------|
| 0 | Default | 행동 없음 |
| 1 | Cooking | 미세먼지 증가 |
| 2 | Overcooking | 미세먼지 급격한 증가 |
| 3 | Bathing | 습도 일시적 상승 |
| 4 | Washing | 지속적인 습도 상승 |
| 5 | In | 온도 상승 |
| 6 | Out | 온도 하락 |

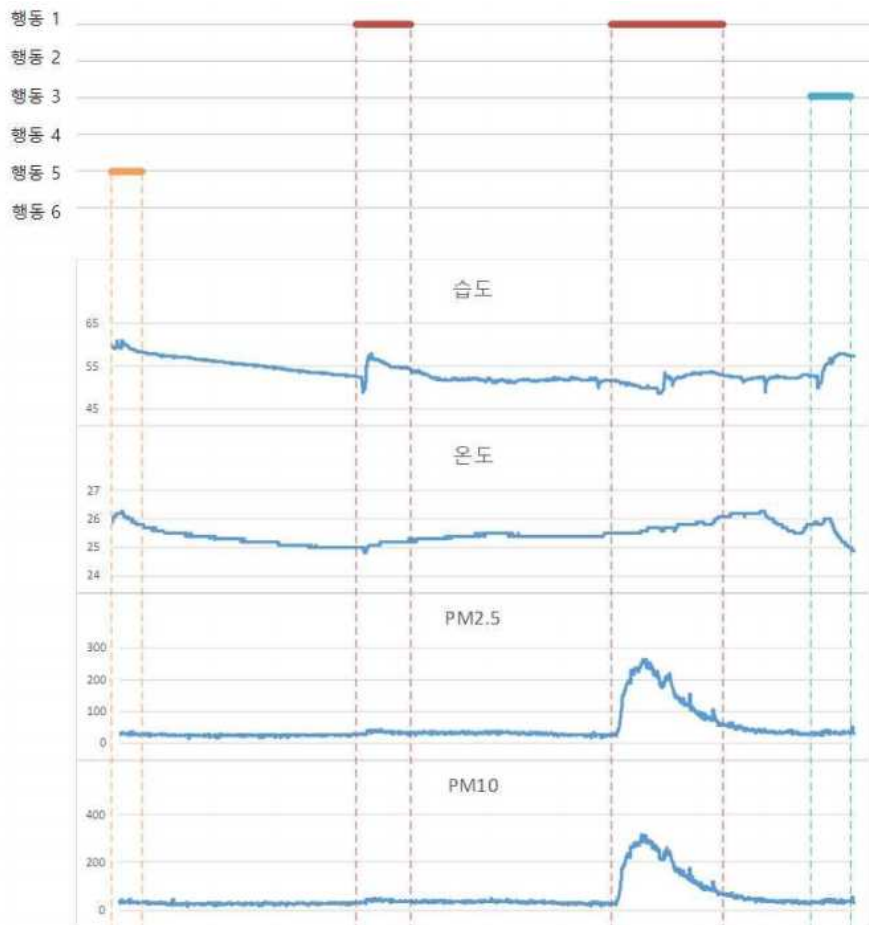
기존 오픈 데이터가 실험 환경에서 수집되었다는 점에서 실생활의 적용력이 떨어지기 때문에 이 연구에서는 직접 센서로부터 위와 같이 행동을 정의하여 데이터를 수집했다. 각각의 수치는 사용자의 행동 유형에 따라 다른 양상을 보였는데, 예를 들어 같은 요리를 하는 상황에서도 요리를 하는 시간이 길어져 음식을 태우는 경우 미세먼지 농도가 급격하게 높아졌다.



Overcooking



3.2.2 행동 발생 기반 슬라이딩 윈도우



각 데이터는 1분단위로 수치들이 기록되는데, 사람이 방에 없거나 수면하는 시간과 같이 행동이 발생하지 않는 구간이 있다. 이러한 구간까지 분류/학습 대상에 포함하는 건 비효율적이며, 동일한 지속 시간에 대해서 분석하는 것이 적절하므로, 슬라이딩 윈도우(Sliding Window)를 적용하여 일정한 크기로 데이터셋을 분할해 labeling한다.

슬라이딩 윈도우 기반 분할 알고리즘의 수도 코드 (pseudo code)는 다음과 같다.

```
Initialize Window_size(s) and Window_start
While R doesn't have the following data do
    If i-th data in R detects an behavior level
        H_i:i+s = {H_i,1, ..., H_i,s}
        T_i:i+s = {T_i,1, ..., T_i,s}
        PM25_i:i+s = {PM25_i,1, ..., PM25_i,s}
        PM10_i:i+s = {PM10_i,1,..., PM10_i,s}
        Append each windowing data to dataset S
        Append behavior label B_n = behavior level
    Shift window(window_size, window_sart)
End While

Return S, B_n
```

3.2.3 데이터 구조 변환

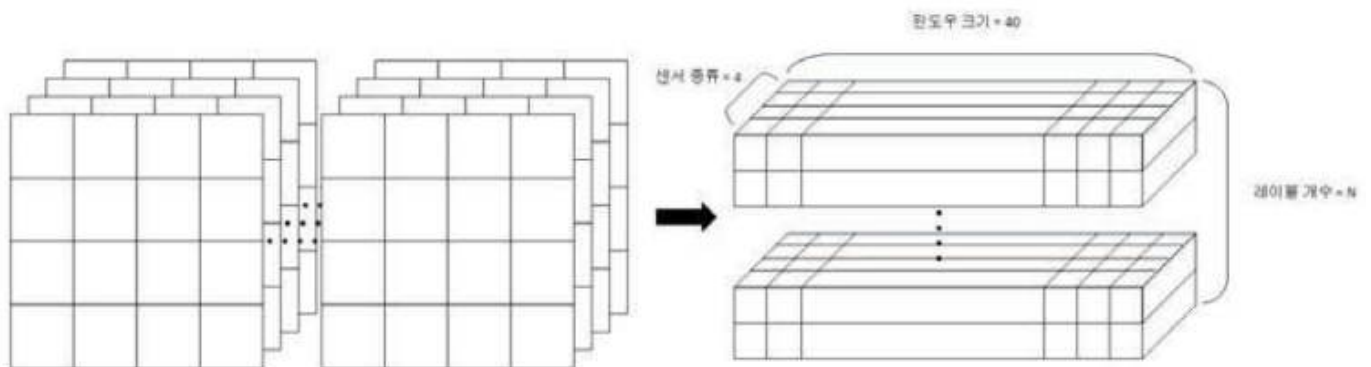


그림 9 2차원 배열 데이터를 3차원 데이터로 구조 변환

슬라이딩 윈도우로 데이터 분할을 마친 2차원 데이터셋 4개가 준비되었다. 이때, Convolutional filter는 3차원 입력값을 받으므로, 제3의 축인 깊이(depth) 방향으로 배열을 합치며 "행동레이블(N) x 윈도우크기(40) x 센서 종류(4)" 구조의 3차원 데이터셋으로 변환해준다.

이러한 변환을 위한 수도코드는 다음과 같다.

```
print a shape of S_2D (input)
S_2D = {[[H_1], ..., [H_n]], [[T_1], ..., [T_n]], [[PM25_1], ..., [PM25_n]], [[PM10_1], ..., [PM10_n]]}
# n개의 윈도우로 분할된 모습
Data Converting(S_2D) # 2차원 -> 3차원 변환
print a shape of S_3D
S_3D = {[[H_11, T_11, PM25_11, PM10_11],
        [H_12, T_12, PM25_12, PM10_12],
        ...,
        [H_140, T_140, PM25_140, PM10_140]]},
```

```

[[H_21, T_21, PM25_21, PM10_21],
...
[H_240, T_240, PM25_240, PM10_240]],
...
[[H_n1, T_n1, PM25_n1, PM10_n1],
...
[H_n40, T_n40, PM25_n40, PM10_n40]]}

```

3.2.4 데이터 정규화

각 데이터의 측정 범위가 다를 때, 비용함수의 분산이 커짐에 따라 경사 하강법 적용이 힘들다는 점에서 데이터의 정규화 및 표준화 작업이 필요하다. 이런 전처리를 해주지 않으면, 입력값의 분포가 고르지 않아 학습률을 매우 작게 해야 하므로 최적값을 찾는 데 시간이 오래 소요될 것이다. 이 연구에서는 정규화 공식으로 다음을 사용했다.

$$X = \frac{(X - X_{min})}{(X_{max} - X_{min})}$$

3.3 Convolutional-LSTM을 사용한 행동 인식

3.3.1 시계열 데이터의 행동분류

다음은 입력값으로 수집한 시계열 데이터의 분할에 적용되는 수식이다. 이 수식에 의하면, 과거 시점부터 현재 시점까지의 데이터를 얻을 때 발생할 행동의 조건부 확률을 구하고, 가장 높은 값을 선택함으로써 가장 분류를 잘 하는 값을 선택할 수 있다. 이때, 입력 데이터의 채널이 여러 개라는 점에서 1차원의 정보만 받는 LSTM 대신 CNN을 사용한다.

$$B_t, \dots, B_{t+k} = \underset{S_{t+1}, \dots, S_{t+w}}{\operatorname{argmax}} p(b_1, \dots, b_6 \mid S_{t-w+1}, S_{t-w+2}, \dots, S_t)$$

3.3.2 Convolutional-LSTM 신경망

종합해보면, 다양한 채널을 가지고 시간에 따라 집계된 입력 데이터에 특징에 따라 CNN 모델과 LSTM 모델을 결합한 모델이 적합하다. 더불어, 이전에 FC-LSTM (Fully Connected LSTM)이 개발되었으나, 셀 내부의 상태 전이에 채널 정보를 반영하지 못한다는 한계에 의해 셀 내/외부의 상태 전이가 모두 가능하여 데이터 손실이 없는 Convolutional-LSTM

모델을 선택한다. 아래와 같이 Convolutional-LSTM 모델은 내부/외부 상태전이 모두 합성곱 연산(\odot)이 적용되어 3차원 데이터의 정보를 보존하는 모습이다.

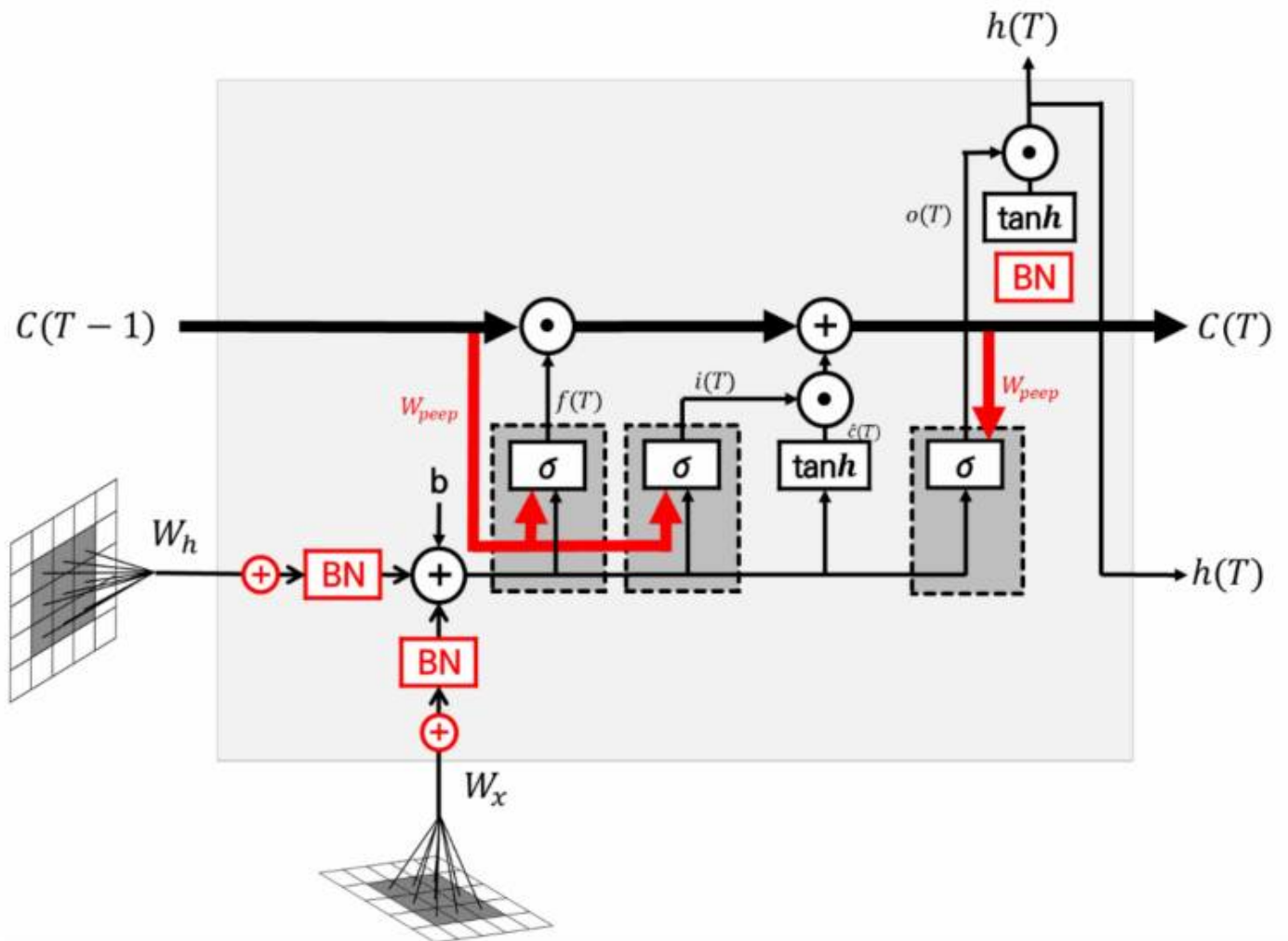


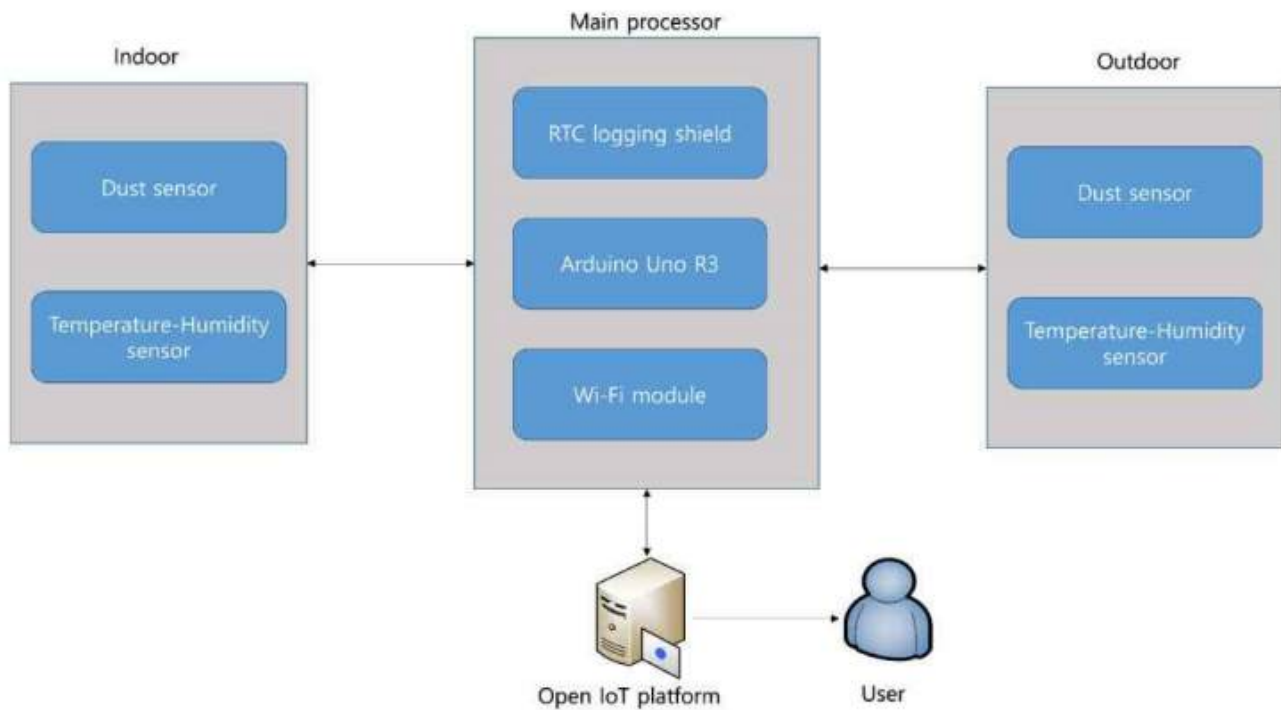
그림 11 LSTM 셀의 연산과정

i_t : 입력 게이트, f_t : 망각 게이트, o_t : 출력 게이트

4. 실험 결과

4.1 실험 환경

4.2.2 데이터 수집부



데이터 수집에 사용되는 하드웨어로는 아두이노, 온도/습도/미세먼지 측정 센서, 측정 시간 기록용 RTC logging shield, 사물인터넷 플랫폼과의 연동을 위한 와이파이 모듈이 있다. 더불어, Thingspeak 플랫폼을 데이터를 수집하고 실시간으로 시각화하는데 이용한다.

4.2 실험 분석 및 결과

행동 수준 정의에 의해 수집한 데이터는 2098개로, 7:3의 비율로 학습, 검증 데이터셋으로 나눈다. 이후 최적의 학습 정확도를 위해 하이퍼파라미터를 조정해보고 아래와 같은 최적 파라미터 수치를 도출한다. 추가로, 과적합을 방지하기 위해 가중치 학습 시 뉴런 간 신호전달을 막는 메서드인 드롭아웃 (Dropout)을 사용한다.

표 6 신경망 파라미터 설정

| Parameter | Value |
|-----------|-------|
| 입력 채널의 수 | 7 |
| 필터 크기 | 64 |
| 활성화 함수 | ReLU |
| 학습률 | 0.8 |
| 드롭아웃 | 0.5 |
| 미니배치 크기 | 64 |

결과적으로, Convolutional-LSTM 모델의 정확도가 가장 높았고, 내부 전이 시 채널 정보를 반영하지 못했던 LSTM 모델의 정확도도 비슷한 수준으로 높게 나왔다. 이로써, 채널을 나누는 것이 큰 효과가 없음을 알 수 있다. 따라서, 차원 변환으로 발생하는 추가적 연산을 고려하면 가성비가 떨어지는 작업일 수 있다.

표 7 학습 모델별 행동 인식 정확도

| Model | Accuracy (Epochs=25) | Accuracy (Epochs=50) | Accuracy (Epochs=75) | Accuracy (Epochs=100) |
|-----------------------|-------------------------|-------------------------|-------------------------|--------------------------|
| CNN | 50.127% | 53.531% | 59.201% | 66.092% |
| LSTM | 75.359% | 82.211% | 86.554% | 90.199% |
| FC-LSTM | 63.237% | 77.620% | 84.049% | 88.484% |
| Convolutional LSTM | 77.261% | 82.012% | 88.247% | 92.151% |

리뷰 후기

이전 논문 리뷰에 이어 Convolutional-LSTM을 IoT와 환경 데이터에 적용한 연구도 재미있었다. 내가 하고 있는 프로젝트에서도 논문에서처럼 파라미터, 모델 종류를 바꿔가며 성능 차이를 테스트해봐야겠다. 다수의 채널을 가진 시계열 데이터에 대해 차원을 변환하고, 슬라이딩 윈도우를 적용해 분할하는 방법도 재미있었다. 개인적으로는 사물인터넷 망 구축을 위해 데이터를 수집하고 정의하는 과정도 흥미로웠다.