AUToSen: Deep Learning-based Implicit Continuous Authentication Using Smartphone Sensors

ABSTRACT:

Accelerometer(가속도계), gyroscope(상하좌우 동작 감지), magnetometer(자기장 감지) 3개의 센서 각각 0.5초 마다 사용 시 정확하게 작동.

센서 데이터를 1초 동안 사용 시:

F1-score: 이항분류(성공 or 실패)에서의 테스트 정확성 → 98%

오인식률(FAR:False acceptance rate): 본인의 것이 아닌 생체인식 정보를 본인의 것으로 잘못 판단할 확률을 의미한다. → 0.95%

오거부률(FRR:False Rejection Rate): 본인의 생체정보를 본인이 아닌 것으로 잘못 판단할 확률을 말한다. → 6.67%

동일 오류율(EER:Equal Error Rate): 오인식률과 오거부률이 같아지는 비율을 말한다. EER의 수치는 ROC곡선으로부터 쉽게 얻을 수 있다.EER은 다른 ROC곡선을 가지는 장치의 정확도를 비교하기 위한 빠른 방법이다.일반적으로, 가장 낮은 EER을 가지는 장치가 가장 정확하다. → 0.41%

센서 데이터를 0.5초 동안 사용 시:

F1-score: 이항분류(성공 or 실패)에서의 테스트 정확성 → 97.52%

오인식률(FAR:False acceptance rate) → 0.96%

오거부률(FRR:False Rejection Rate) → 8.08%

동일 오류율(EER:Equal Error Rate) → 0.09%

INTRODUCTION:

현재 스마트폰에 적용된 보안방식에는 Knowledge-based 와 Physiological biometrics-based 방법이 있음. 전자는 비밀번호와 패턴인식 등이고, 후자는 지문인식과 얼굴인식이다. 하지만 두 가지 모두 보안취약점이 있을 뿐만 아니라, 일단 한 번 통과하면 그 후에 계속적인 보안을 보장할 수 없다는 단점이 있다.

→ 그렇기 때문에 계속적으로 보안인증을 할 수 있는 biometrics-based 방법이 필요하다. 스마트폰에는 다양한 센서들이 장착되어 있기 때문에 센서들을 이용하여 스마트폰 평소 사용패턴을 이용하여 스마트폰 소유주가 맞는지 확인할 수 있다. 이러한 방법은 transparent, continuous, implicit(etc..)한 인증방식으로 불린다. 이러한 인증방식은 기존의 인증방식에 추가로 사용되어, 스마트폰 사용 중에 스마트폰의 소유주가 아니라고 판단 된다면 주요인증(primary authentication)을 다시 요구할 수 있다.

AUToSen 은 다음과 같은 장점이 있다: 백그라운드에서 유효한 유저인지 계속적으로 확인한다. 민감한 소프트웨어 및 하드웨어 권한을 필요로 하지 않아 유저의 개인정보를 침범하지 않는다. ABSTRACT 에서 언급한 단 3 개의 센서만으로도 충분히 유저의 행동패턴을 모델링 할 수 있다.

0.5 초 또는 1 초 같이 짧은 시간 동안 인증이 작동하기 때문에 높은 frequency 의 작동이 필요하다.

AUToSen: OVERVIEW

인증모델은 LSTM(장단기 메모리 모델)을 이용하여 순차적 센서 데이터를 처리하여 사용자의 행동패턴을 분석한다. 유저가 하고 있는 행동(웹서핑, 전화, 동영상 감상) 등에 상관 없이 작동한다. data preprocessing(데이터 전처리), temporal alignment(싱크 맞추기), feature extraction(특징 추출), and sequential modeling 등이 필요하다.

메인 프로세스:

센서 데이터 수집 → 데이터 전처리 → 인증모델에 데이터 feeding

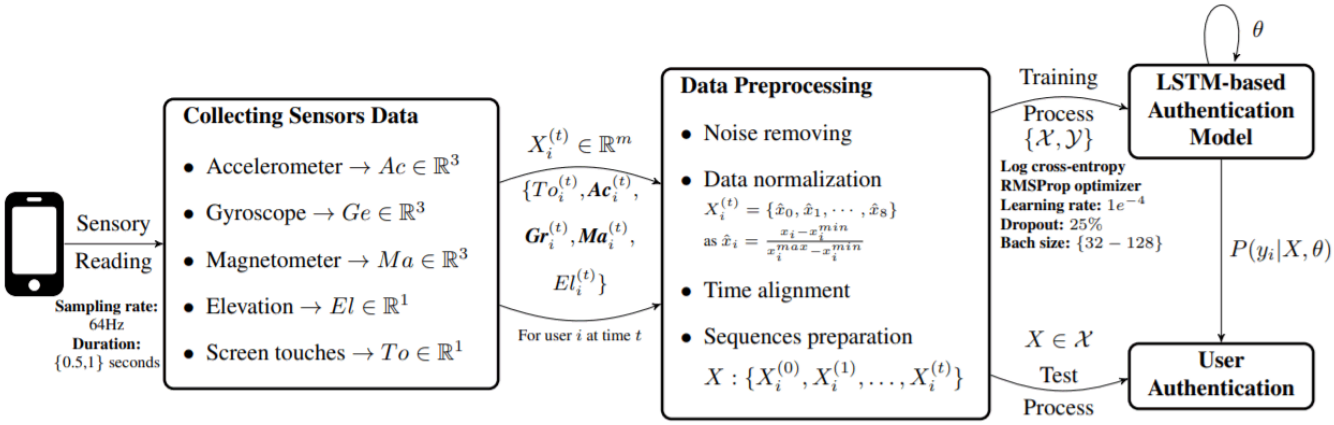


Fig. 1. AUTo*Sen*: an implicit authentication system overview.

A. Data Collection

안드로이드 데이터 수집 어플리케이션을 사용해 스크린 터치, 가속도계, 자이로스코프, 자기장계, 고도를 측정.

피실험군은 평균나이 25세, 표준편차 4.49의 84명. 5일 간 데이터 수집 어플리케이션을 실행해 데이터 수집.

Accelerometer(가속도계): 중력가속도를 수집하며, 3차원 벡터Ac(x,y,z)로 표현됨. 단위는 m/s^2.

Gyroscope: x,y,z 축에서부터의 각 회전각을 측정. 3차원 벡터 $Gr(\theta_x, \theta_y, \theta_z)$로 표현됨. 단위는 θ/s.

Magnetometer: 자기장 측정. 3차원 벡터 $Ma(T_x,T_y,T_z)$. 단위는 T(테슬라).

실험에 사용된 DataSet 4가지

1) Five-Sensor Dataset (ToAcGrMaEl)
2) Four-Sensor Dataset (AcGrMaEl)
3) Three-Sensor Dataset (AcGrMa)
4) Two-Sensor Dataset (AcGr)

B. Data Processing

노이즈 핸들링, 시간 정렬이 필요.

수집된 데이터는 $X_i^{(t)} \in R^m$ 으로 표현.

i는 사용자, t는 시간, m은 전체 데이터의 차원(ex. 5개의 데이터셋을 사용하면 11차원).

측정되지 않은 값(Missing values) 처리:

Screen Touch: 0으로 대체

고도: 마지막으로 측정된 고도로 대체

나머지 3개의 센서측정값: 기존에 측정된 값의 평균값으로 대체. 사이즈 5의 윈도우를 사용.

Data Normalization: 노이즈의 영향을 최소화 하기 위해 정규화. 대부분의 센서측정값은 [0,1] 안에 있음. 데이터 정규화를 위한 time frame을 5초로 사용. Sampling period를 1초로 할 때는 5번의 0.5초로 할 때는 10번의 최댓값과 최솟값을 측정하는 구간이 생김.

$X_i^{(t)} = \hat{x}_0, \hat{x}_1, \cdots, \hat{x}_{m-1}$ where $\hat{x}_i = \frac{x_i - x_i^{min}}{x_i^{max} - x_i^{min}} \in R \mid \forall x_i \in [0, 1]$

Sequence Generation: 센서 데이터는 64Hz의 rate로 수집됨. 데이터는 $X_i^{(t)} = To_i^{(t)}, Ac_i^{(t)}, Gr_i^{(t)}, Ma_i^{(t)}, El_i^{(t)}$ 형식으로 Align 됨. Sampling period를 0.5초로 하면 길이 32의 시퀀스들을, 1초로 하면 길이 64의 시퀀스들을 생성한다. Accelerometer와 Gyroscope의 값 둘 중 하나라도 5초 이상 바뀌지 않으면 Inactive 상태로 판단하여 모델링 데이터에서 제거한다. 즉 Active 상태만 모델링 데이터로 사용한다.

C. LSTM-based User Authentication

LSTM을 이용하여 Binary Classification을 함. 아웃풋은 정당한 사용자인가 vs 침입자인가로 2가지 클래스이다. 인풋 데이터 $\{X_i^{(0)}, X_i^{(1)}, \dots, X_i^{(n-1)}\}$에 대하여 확률 $P(y_i \mid X : \{ X_i^{(0)}, X_i^{(1)}, \dots, X_i^{(n-1)}\}, \theta)$ where θ는 LSTM 파라미터, $y_i = \{0,1\}$을 계산한다. LSTM은 RNN에서 긴 시퀀스를 처리할 때 기울기의 vanishing과 exploding을 극복하기 위한 RNN의 변종이다. LSTM은 gating mechanism과 memory cells $C_i^{(t)}$를 사용한다. 인풋 데이터 $X_i^{(t)}$, hidden state $h_i^{(t-1)}$, 메모리 셀 $C_i^{(t-1)}$이 주어졌을 때 LSTM 유닛은 $h_i^{(t)}$, $C_i^{(t)}$를 계산한다.

첫 번째로 LSTM은 4가지 gate를 계산한다: input 게이트($\mathfrak{i}$), forget 게이트($\mathfrak{f}$), output 게이트($\mathfrak{o}$), input modulation 게이트($\mathfrak{g}$).

$\mathfrak{i} = \text{sigmoid}(W_{x\mathfrak{i}}Xi^{(t)} + W_{h\mathfrak{i}}h_i^{(t-1)})$,

$\mathfrak{f} = \text{sigmoid}(W_{x\mathfrak{f}}X_i^{(t)} + W_{h\mathfrak{f}}h_i^{(t-1)})$,

$\mathfrak{o} = \text{sigmoid}(W_{x\mathfrak{o}}X_i^{(t)} + W_{h\mathfrak{o}}h_i^{(t-1)})$,

$\mathfrak{g} = \tanh(W_{x\mathfrak{g}}X_i^{(t)} + W_{h\mathfrak{g}}h_i^{(t-1)})$

Where $\text{sigmoid}(x) = (1+e^{-x})^{-1}$, $\tanh(x) = (e^{2x}-1)(e^{2x}+1)^{-1}$. Then,

$C_i^{(t)} = \mathfrak{f} \odot C_i^{(t-1)} \odot \mathfrak{g}$

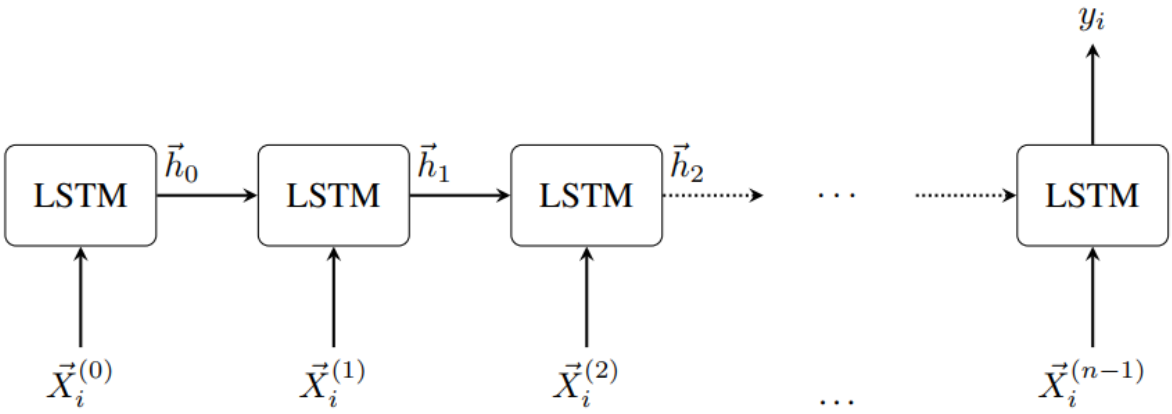
$h_i^{(t)} = \mathfrak{o} \odot \tanh(C_i^{(t)})$

where $\odot$ 는 원소별 곱셈.

마지막 time step, t = n-1에서 아웃풋 확률은 $P(y_i \mid X, \theta) = \text{sigmoid}(W_{hy}h_i^{(n-1)})$로 계산되고, 계산된 값이 0.5 이상이면 아웃풋 $y_i = 1$이다.

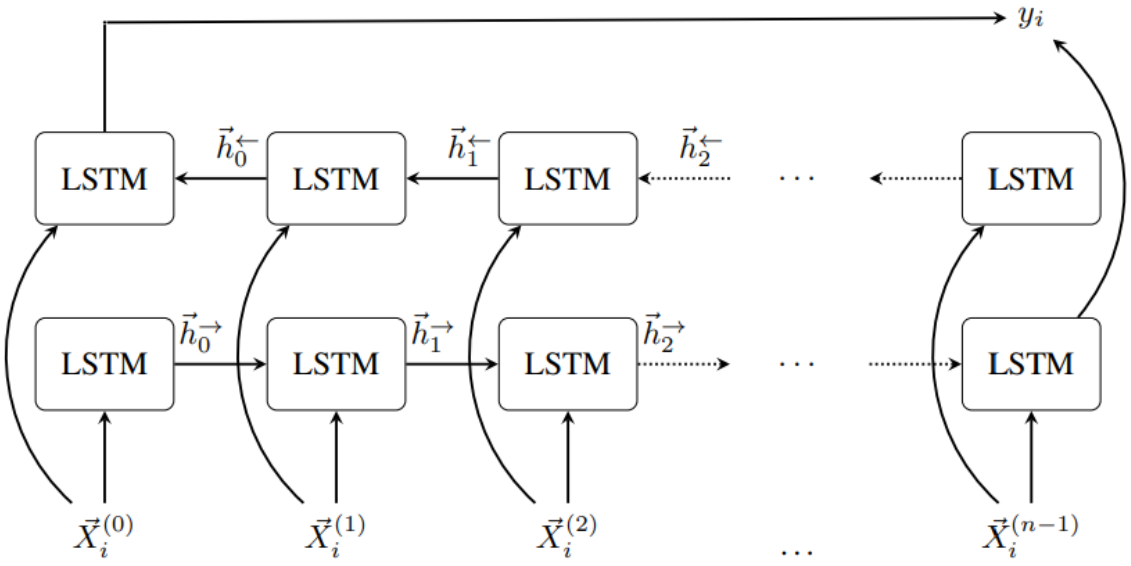
퍼포먼스 측정을 위한 3가지 종류의 LSTM:

1. Simple LSTM: RNN unit i는 RNN unit i-1의 정보만을 사용하여 i의 상태를 만들고, 이 정보를 다음 상태에 전파함.



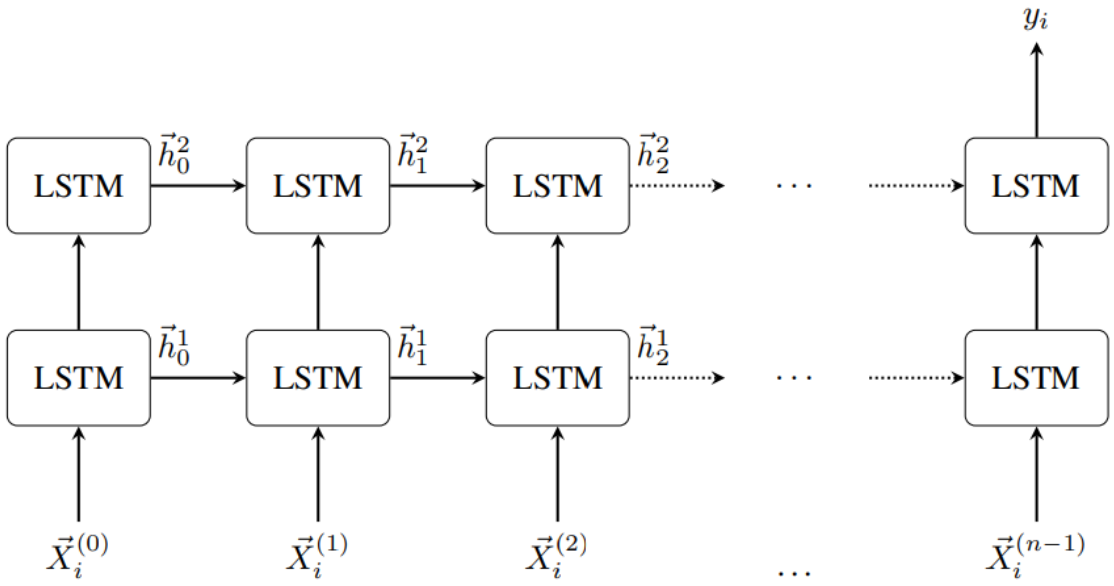
(a) Simple LSTM model.

2. Bidirectional LSTM: 현재 유닛의 상태를 만들기 위해서 과거뿐만 아니라 미래의 정보까지 사용함. Simple LSTM보다 성능이 좋음. LSTM 두 개를 사용하는데 첫 번째 LSTM은 {$X_i^{(0)}$, $X_i^{(1)}$, ... , $X_i^{(n-1)}$} 순서로 작동하고, 다른 RNN은 { $X_i^{(n-1)}$, $X_i^{(n-2)}$, ... , $X_i^{(0)}$} 순으로 작동함.



(b) Bidirectional LSTM model.

3. Multilayers LSTM: 2레이어를 사용함. 복잡한 패턴을 캡쳐하는 데에 좋음.



(c) Multi-layers LSTM model.

모든 hidden recurrent layer는 16개에서 256개의 LSTM 유닛으로 구성. 아웃풋 레이어는 Sigmoid 레이어이며 진짜 사용자인지 확률을 계산.

인증모델은 소유주(legitimate user)의 데이터 뿐만 아니라 비소유주(imposters)의 데이터 또한 사용하여 학습함. 소유주의 데이터는 Positive로, 비소유주의 데이터는 Negative로 표시됨. 모델을 학습시킬 때 1명의 실소유주와 10명의 비소유주의 데이터를 사용하며 비소유주의 데이터 총합이 실소유주의 데이터보다 5배 많음. stratified 10-fold cross-validation을 사용: fold(겹) 마다 돌아가면서 1 fold는 검증하는 데에, 9 folds는 학습시키는 데에 데이터를 사용. 최종 결과는 10번의 실험결과의 평균임.

모델을 학습시킬 때 Weight은 랜덤한 값으로 시작한다. 최적화 과정은 log cross-entropy loss(binary corss-entropy)을 최소화 하는 방법으로 시행된다. Log cross-entroy는 다음과 같이 정의된다.

$\text{loss}(\theta) = \frac{-1}{N}\sum_{n=1}^{N}[y_i \times \log(P_n) + (1 - y_i) \times \log(1 - P_n)]$

where $P_n = P(y_i \mid X, \theta)$.

학습 시키기 위해 RMSProp 알고리즘을 사용. 학습률은 $1e^{-4}$. 또한 dropout을 사용함.

효율적인 학습을 위하여, 미니배치를 사용함. 수 많은 샘플들은 [batch_size, sequence_length, sample_length] 차원의 텐서로 packed 됨. Batch size는 32~128 샘플이며, sequence length는 데이터 샘플링이 0.5초일 때 32, 1초일 때 64이고, sample_length는 인풋에 사용하는 센서의 개수가 5개일 때 11, 4개일 때 10, 3개일 때 9, 2개일 때 6이다.

D. Authentication Evaluation Metrics

F1-score가 정확성의 척도가 됨.

$\text{F1-score} = 2 \times (Recall \times Precision) \div (Recall + Precision)$

Where Recall = $(TP) \div (TP + FN)$ → 실제 Positive였던 경우 중 Positive로 보고된 비율

Precision = $(TP) \div (TP + FP)$ → Positive로 보고된 경우 중 실제 Positive였던 비율

TP: True Positive

FP: False Positive

FN: False Negative

FAR(침입자를 소유주로 잘못 받아들이는 비율) = $(FP) \div (FP + TN)$

FRR(소유주를 침입자로 오인하여 거부하는 비율) = $(FN) \div (FN + TP)$

EER(FAR과 FRR이 같아지는 비율)

E. AUToSen's Operations

유저등록: 데이터 수집, 데이터 클리닝 및 전처리, 인증모델 학습 및 평가로 구성됨. 계산의 필요성 때문에 등록은 인증서버에서 처리됨. 인증서버는 학습과 유저모델을 지속적으로 업데이트 하기 위하여 필요함. 유저의 행동패턴이 바뀌어 일정 횟수 이상의 False Alarm이 보고되면, 모델을 재학습 시킬 수 있다.

유저 연속적 인증: 로컬인증모듈과 클라이언트/서버 디자인 방법 두 가지가 있다. 후자를 사용하면 모델 학습을 위해 서버를 이용할 수 있다. 이 방법을 사용해도 데이터 전송은 많지 않고, 실시간으로 인증을 처리하는 데에 충분하다. 하지만 단점으로는 인터넷 연결 문제등의 이유로 제대로 작동하지 않는 경우가 있다는 것이다. 이런 점 때문에 전자가 선호된다. 최근 스마트폰의 연산속도 및 저장공간 증가와 텐서플로우 라이트의 지원으로 가능하다. 하지만 이 연구에서는 후자를 사용

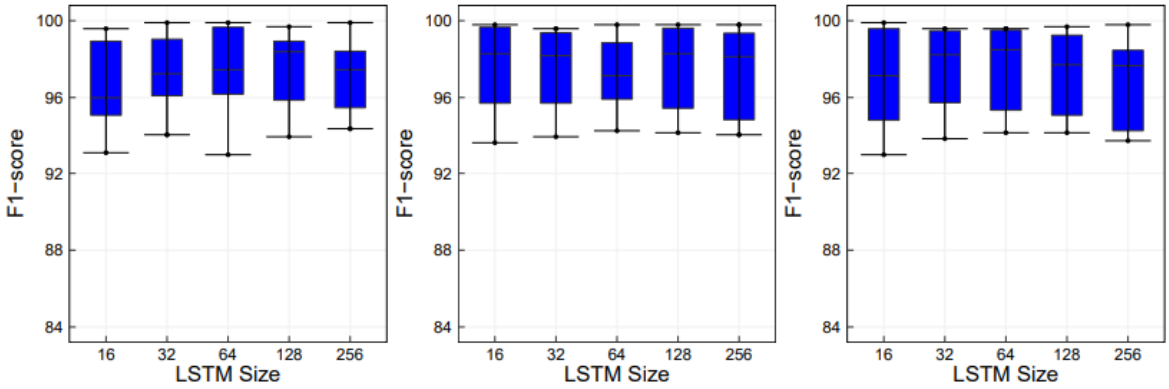
하기로 한다.

EXPERIMENTS AND EVALUATION

A. The Effects of Sensors Data

위에서 언급한 3가지 LSTM 기법을 사용했고, 샘플링 period는 1초, 샘플링 rate는 64Hz임.

Three Sensors(AcGrMa): 대부분의 경우 이 세 가지 센서만 사용하는 것이 가장 좋은 결과를 보였음. 그 이유는 스크린 터치와 고도는 이 세 가지 센서보다 더 민감하고, 상황에 따라 크게 다르기 때문.
Simple LSTM 모델에서 모델 사이즈를 16으로 한 경우 F1-score는 평균 95.59%가 나왔다. Bidirectional LSTM 모델의 경우 16개 모델 사이즈를 사용한 경우 평균 97.49%, 128개를 사용한 경우 97.59%가 나왔다.



(a) Simple LSTM (b) Bidirectional LSTM (c) Multi-layer LSTM

Fig. 5. The accuracy of different LSTM model architectures when we feed the authentication model with three sensors (AcGrMa) data sequences collected within a second sampling period.

앞으로 나의 계획 및 할 일:

이 논문을 읽어본 결과 어떤 방법을 썼는지는 언급되었지만 세부내용까지는 언급되어 있지 않다. 그러므로 직접 연구해야 할 부분이 많다. 또한 Accelerometer, Gyroscope, Magnetometer 센서만 이용하는 것으로 충분한 것이 이 연구에서 밝혀졌다. 그러므로 앞으로 만들 작품에서는 고도계와 스크린터치는 배제할 것이다. 또한 Simple LSTM 모델도 평균 95% 이상의 좋은 결과를 보여주었으므로, 간단화하기 위하여 이 모델을 사용할 것이다. 또한 이 논문에서는 서버/클라이언트 방법을 이용하였지만, 가능하다면 텐서플로우 라이트를 이용하여 로컬방법을 사용을 해볼 것이다. 이 연구를 위하여『인공지능을 위한 텐서플로우 애플리케이션 프로그래밍, 이종서등 저, 광문각, 2019』, 『Intelligent Mobile Projects with Tensorflow, Jeff Tang저, PacktPublishing, 2018』를 공부할 것이다.

REFERENCE

이 요약서의 모든 내용 및 사진자료는 인하대학교 Mohammed Abuhamad, Dae Hun Nyang, 성균관대학교 Tamer Abuhmed, University of Central Florida 의 David Mohaisen 교수님의 AUToSen: Deep Learning-based Implicit Continuous Authentication Using Smartphone Sensors 논문에 근거함.

논문 링크: https://ieeexplore.ieee.org/abstract/document/9007368/