

머신러닝 기반 스마트 거짓말 탐지기

최재은, 박상우, 김건주, 박규동

Smart Lie Detector

Jaeun Choi, SangWoo Park, GunJoo Kim, GyuDong Park

요약

본 논문에서는 거짓말 탐지기과 머신러닝을 접목한 기술에 대한 시스템을 제안한다. 이 머신러닝 알고리즘과 거짓말탐지기를 결합한 시스템은 기존의 거짓말탐지기의 결과보다 조금 더 정확성을 제공한다. 제안된 시스템은 크게 데이터 입력, 머신러닝, 데이터 출력으로 나누어진다. 원리는 센서를 사용자에게 부착하여 데이터 값을 입력받고, 해당 데이터 값을 RNN알고리즘을 통해 데이터 분석과 머신러닝을 통해 학습시키고, 학습된 데이터를 다시 사용자에게 맞추어 보다 정확한 값을 통해 나타내게 한다. 즉 기존 거짓말 탐지기보다 조금 더 정확한 결과 값을 얻기 위하여 위와 같은 기술을 접목시켜 개발하였다.

1. 서론

최근, 4차 산업혁명 시대에 들어서면서 인공지능, 사물 인터넷, 빅데이터, 클라우드 컴퓨팅 등 첨단 정보통신 기술이 경제, 사회 전반에 융합되어 혁신적인 변화가 나타나고 있다. 이렇게 빠르게 변하는 추세에 인공지능 또한 그 부각을 드러내고 있는데, 이에 따른 인공지능 기술중 하나로 머신러닝을 꼽을 수 있다. 머신러닝(Machine-Learning)이란, 기계학습의 다른말로, 경험을 통해 자동으로 개선하는 컴퓨터 알고리즘의 연구라고 할 수 있다. 이에 따라 미래에는 빅데이터를 통한 예측 프로그램, 머신러닝을 접목한 웨어러블 기기 등, 인공지능을 활용한 미래성이 예측되는 바이다. 현재는 단순히 하드웨어적인 부분과 소프트웨어적인 부분이 나뉘는데, 이를 접목시킨 기술이 시장에 등장하고, 점유율이 높아지기 시작함에 따라 기존의 제품들 역시 한층 더 업그레이드가 되어야 하는 시점이라고 생각한다. 스마트 거짓말 탐지기(Smart Lie Detector)는 기존 거짓말탐지기와는 다르게, 머신러닝 기술을 접목한 거짓말 탐지기로써, 센서를 이용한 사용자들의 데이터를 수집하고, 그에 따른 데이터 분석을 마치고 다시 사용자에게 데이터를 제공하는식의 결합품이 될 것이다. 이는 단순히 머신러닝과 거짓말 탐지기의 합작품으로 볼 수 있지만, 넓게 보면 머신러닝과 하드웨어의 결합으로 보다 폭넓은 기기들의 개발과 필요성을 알려줄 수 있는 계기가 될 것이다.

본 논문에서는 GSR센서와 심장박동 센서 그리고 야두이노를 활용한 거짓말 탐지기과, 머신러닝을 이용하여 얻은 정보를 거짓말 탐지기라는 물리적인 요소에 장착시켜 보다 정확한 결과값을 도출해 낼 수 있는 거짓말 탐지기를 개발하고자 한다.

2. 관련 연구

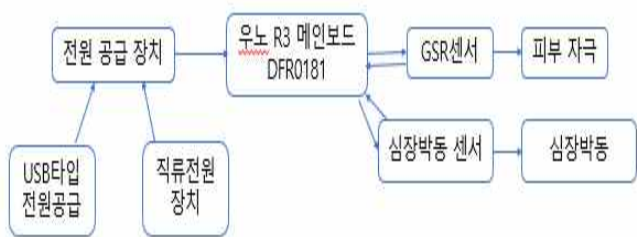
머신러닝은 다양한 속성을 포함한 충분한 데이터가 확보되어야 안정적이며, 신뢰적인 성능을 보인다. 다양한 목적 및 상황에 맞는 추천을 제공하기 위해 데이터 속성 확장 메커니즘과 머신러닝 기반 음식점 추천 시스템을

제안하는 바이다. 제안하는 시스템은 동적인 특징 추출로 대상의 특성을 유연하게 반영할 수 있으며, 보다 상세한 사용자의 취향 구분이 가능하다. 또한 주문량 예측 결과 20% 미만의 오차를 보여주어 추세 파악이 가능할 것으로 생각된다.[1].

머신러닝은 데이터를 기반으로 예측 모델을 생성하는 방법으로서, 높은 수준의 성능을 보이는 방법일수록 내부 알고리즘이 복잡하여 입력데이터에 대한 예측결과 해석이 어려운 것으로 알려져 있다. 선형회귀나 의사결정트리(Decision Tree)와 같은 알고리즘에서는 어느 정도 예측과정에 대한 설명이 가능하다. 머신러닝의 예측 과정 해석에 대한 연구는 시스템의 안정성 및 신뢰성 확보 등을 위해서 점차 중요한 분야로 인식 되고 있다.[2].

본 논문에서 첫 번째로 다루고자 하는 것은 거짓말 탐지기의 하드웨어인 아두이노이다. 이용하고자 하는 센서는 두가지인데 그 중 첫 번째인 GSR센서는 감정, 정서등에 관련된 문제를 연구할 때에 많이 쓰이고 있는데, 본 논문에서 쓰고자 하는 거짓말 탐지기의 센서 종류로 GSR센서가 활용되었다. [3].

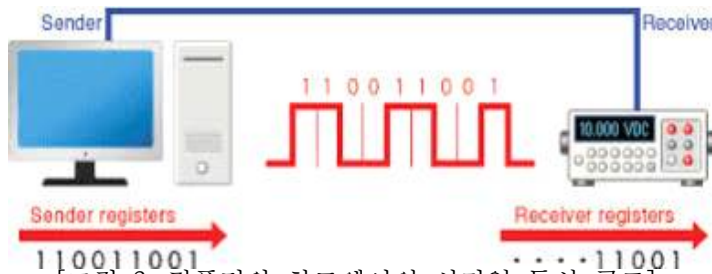
전 세계인구 72억명 중 활성화 된 인터넷 사용자는 약 30억여 명, 소셜 미디어 이용자는 약 20억명 등의 숫자가 정리되어 있다. 또한, 4가지 지표 모두 지난 몇 년간 빠르게 성장하고 있고, 인터넷 인구는 전년대비 21%, 모바일 소셜미디어 계정은 23%로 증가하는 것과 같다. 이렇게 많은 이용자들은 모두 익명제로 모든 표현을 아무 제약없이 자유롭게 표현하고 있다. 이로 인해 윤리 의식과 올바른 에티켓이 결여된 악성 댓글로 피해사례가 자주 보이고, 진실과 거짓 사이에 정확하지 않은 정보로 선동되는 일도 빈번하다. 이를 잡기 위해 거짓말 탐지기를 머신러닝과 접목하여 개발하고자 한다.[3].



[그림1. 아두이노를 이용한 거짓말탐지기의 구성도]

3. 거짓말 탐지기와 서버의 시리얼 통신

거짓말탐지기의 휴대성을 이용한 아두이노와, 머신러닝을 접목하기 위하여 시리얼 통신을 채택하였다. 이는 일반적으로 아두이노와 컴퓨터간의 일반적인 통신방식이다.



[그림 2. 컴퓨터와 하드웨어의 시리얼 통신 구조]

1) 하드웨어

GSR센서와 심장박동 센서 사용을 위해 비교적 간편한 아두이노를 선택했다. 아두이노는 컴퓨터와 시리얼 통신을 통해 컴퓨터의 파이썬 코드로 제어가 가능하게 구현하였으며, Arduino를 통해 GSR센서와 (HRT)심장박동 센서를 제어 하였다. 시리얼 통신을 위한 별도의 pyserial 라이브러리를 설치하여 시리얼 통신간의 정보를 컴퓨터에게 제공한다. GSR센서 부분은 아두이노 코드만 작성되어 있으며 HRT(심장박동센서)는 HeartSpeed-master 라이브러리가 적용되어 있다. 이 라이브러리 소스에는 자체 Kalman filter를 사용해 측정되는 잡음이 실린 데이터로 부터 동적시스템의 미지의 상태변수를 최적으로 추정하기 위한 선형,불편 최소오차분산 반복적 알고리즘이 적용되어 있다.

2) 서버

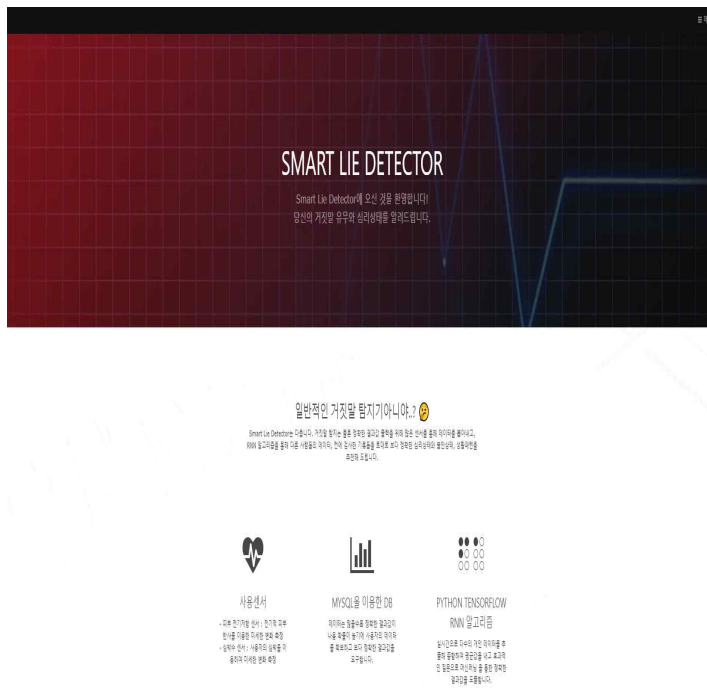
REST API라는 Representational State Transfer라는 용어의 약자로서, 웹의 장점을 최대한 활용할 수 있는 아키텍처로 특징은 무상태성 성격을 가지고 캐시가 가능한 이유로 사용하였다. Client와 Server의 구조를 갖고 있다. 또한 WAS(Web Application Server)를 이용하여 DB조회나 다양한 로직 처리를 요구하는 동적인 콘텐츠를 사용하기 위함이다. WAS를 이용하여 분산 트랜잭션, 보안, 메시징, 쓰레드 등의 처리 기능을 DB서버와 같이 수행하기 위함이다. [그림 3]과 같이 html과 css를 이용한 highcharts spline차트 라이브러리를 사용한다. 그래프는

클라이언트 측에서 비동기통신으로 데이터를 요청하면 서버측에서 저장하고, 서버측에서 다시 데이터를 받기 위해 serial 통신 모듈을 멀티스레드로 구현한다.



[그림 3. html과 highcharts 라이브러리를 이용한 그래프]

3) UI



[그림 4. UI의 MAIN화면]

UI는 HTML과 CSS, JAVASCRIPT를 이용하여 웹으로 제작되었다. 이는 실행을 위해 따로 라이브러리를 설치하는 것을 배제하고 접속 도메인만 알고 있다면 바로 접속이 가능하며, 즉 일반 사용자들이 보다 쉽게 접근할 수 있도록 한다. 접근성이 쉬운 UI 임과 동시에 기기만 준비된다면 언제 어디서든 거짓말 탐지기 테스트를 받을 수 있도록 사이트로 제작되었다.

※ **센서 부착 전 방해요소가 없는지 확인!**

I. **센서 착용순서**

- ① 아래 사진처럼 GSR 센서를 장착합니다!
- ② 아래 사진처럼 심박수 센서를 장착합니다!
- ③ 잘 부착이 되었는지 확인해 주세요!



II. **홍분하지 말고 차분을 유지하세요.**

자분한 음악을 준비했습니다. 🎵



자분한 사진을 준비했습니다. 📷



III. **검사할 준비가 되셨습니까?**

네! 검사하겠습니다! 🏃‍♂️

[그림 5. 시작하기 전의 유의사항 UI]

거짓말 탐지기는 말 그대로 감정상태의 분석을 통한 결과 값을 도출해 내는 프로그램이기 때문에, 보다 차분한 감정 상태를 요구한다. 이는 거짓말 탐지기 테스트를 받기 전, 보다 높은 수준의 정확도를 얻기 위해 사용자의 감정을 최우선적으로 생각하고, 노래와 사진들로 감정을 차분하게 만들 것을 요구한다.

I. **피부 전기저항 그래프 값**



II. **심박수 그래프 값**



III. **진단표**

빈 농도	거짓말 농도	바로 달하는 순발력	아두이노 측정값
78%	87%	매우빠름	75.5

True False

결과값 보기

그림 6. 사용자의 데이터를 센싱중인 화면

[그림 6]은 다음과 같이 사용자의 데이터를 센싱하는 화면인데, 이 화면을 통해 감정을 분석하고 결과 값을 도출해 낸다. 사전에 학습된 데이터를 통해 그래프의 높낮이의 변화에 따른 감정 상태를 보다 정확하게 분석할 수 있으며, 이는 궁극적 목적인 거짓말 탐지기의 결과 정확도에 영향을 끼치는 요소이다.

4) **데이터 베이스**

데이터베이스(이하 DB)는 MySQL을 사용하였다. MySQL을 사용한 이유는 전 세계에서 가장 널리 사용되는 오픈소스형 DB이고, 그만큼 많은 기능들과 수행능력을 가지고 있어 채택되었다. MySQL을 DBeaver와 연동하여 MySQL에서는 가질 수 없었던 보기 쉬운 UI를 DBeaver를 통해 쉬운 DB관리를 할 수 있다.

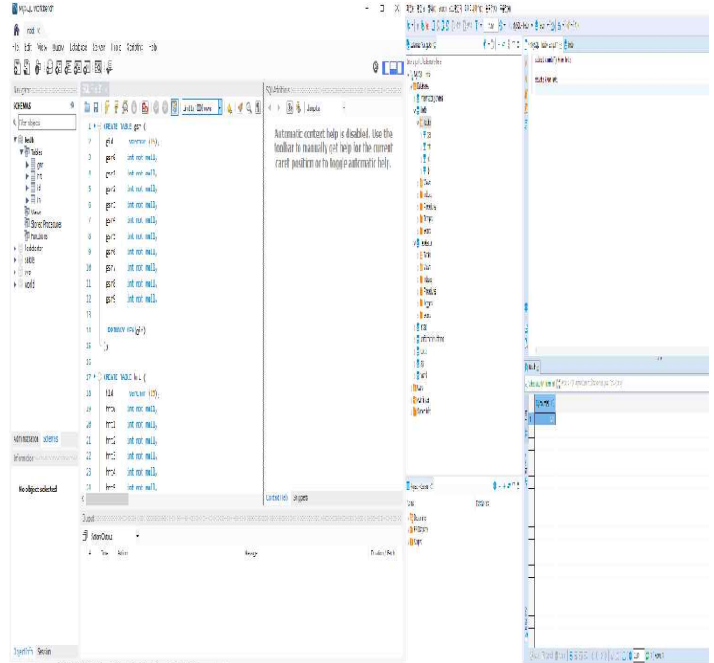
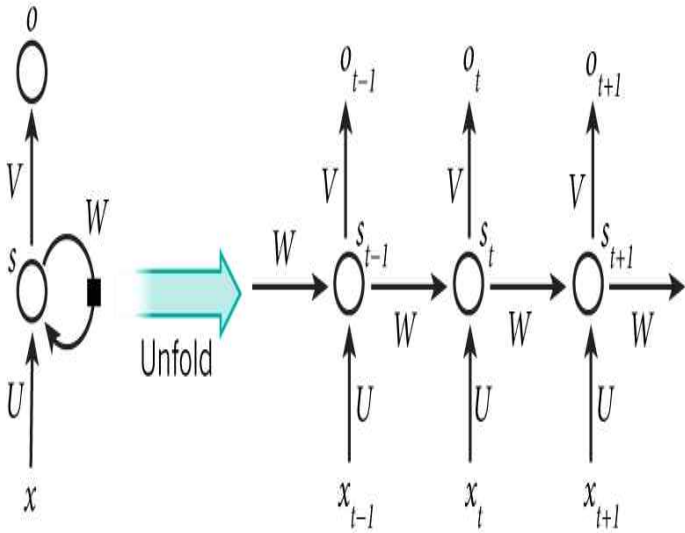


그림 7. mysql의 인터페이스와 DBeaver의 인터페이스 [그림 7]과 같이 mysql과 DBeaver의 기능적인 면에서는 차이가 없지만, 인터페이스 요소와 단순 스크립트 실행을 위해서 DBeaver를 연동하였다.

4. **거짓말 탐지기의 머신러닝**

단순 아두이노를 이용한 거짓말 탐지기는 시중에도 현재 많은 프로젝트 제작물 혹은 상품성으로도 나오고 있는 추세이다, 하지만 앞서 서론에서 설명한 것처럼 4차 산업혁명시대에 맞는 AI의 기술 중 일부인 머신러닝을 접목하여 사용하고자 한다. 기계학습을 통해 데이터를 산출하여 컴퓨터에게 보다 정확한 결과 값 예측을 요구한다. 머신러닝 방식은 먼저 아두이노에서 시리얼 통신으로 받아온 값을 DB에 저장하고, 비동기통신으로 클라이언트 측에서 데이터를 요청하면 서버 측에서 저장하여 다시 데이터를 뿌려주는 형식인데, 최근 9개의 센싱된 데이터들이 그래프에 저장된다. 이 때 True와 False라는 라벨로 구분을 지어놓은 것으로 나누고, 데이터를 DB에 저장하고, 추후 해당 저장된 데이터 값으로 거짓말 탐지

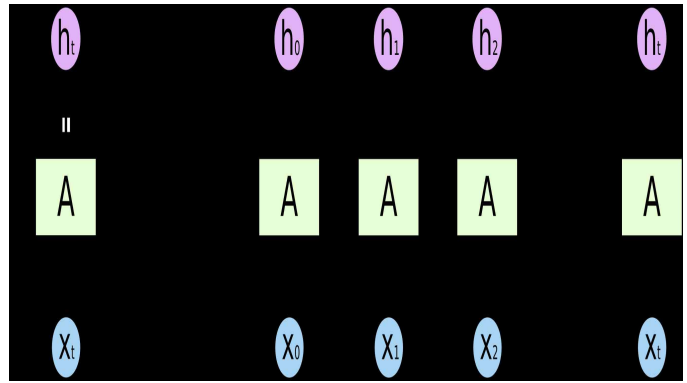
기를 작동시킬 때, 그래프의 변화의 주기가 크지 않도록 한다. 그리고 Tensorflow(머신 러닝 시스템)으로서 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리인데, 이 Tensorflow 중 하나인 KERAS를 채택하여 머신러닝을 구동하는데 활용하였다. 가장 흔하게 사용되는 모델임과 동시에 모델은 일반적으로 층의 그래프를



가지고 있기 때문이다.

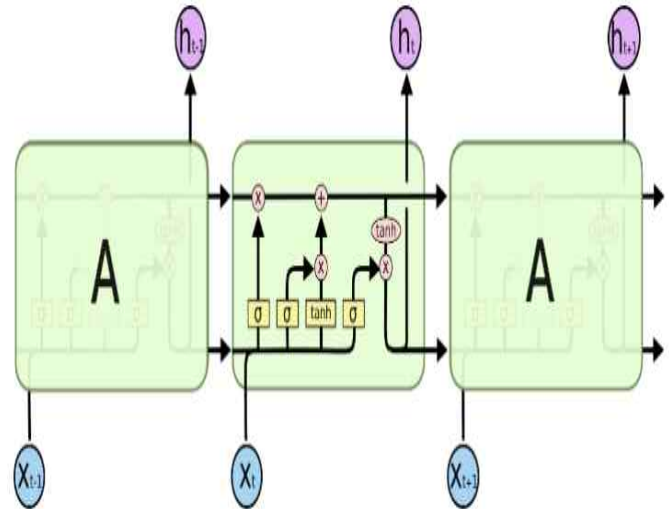
[그림 8. RNN 알고리즘의 구조]

RNN이란 Recurrent Neural Networks의 줄임말로써, 한글로 말하면 신경망이라고 할 수 있다. 다른 머신러닝의 신경망인 FFNets는 데이터를 입력하면 입력층에서 은닉층까지 차근차근 연산이 진행되고 출력이 나가게 되는데, 이 과정에서 입력 데이터는 모든 노드를 딱 한 번씩만 지나가게 되는 구조이다. 데이터가 노드를 한 번만 지나가게 된다는 것은 데이터의 순서, 즉 시간적인 측면을 전혀 고려하지 않는 구조라는 것이다. 이는 데이터들의 시간 순서를 무시하고 현재 주어진 데이터를 통해서 독립적으로 학습을 하는 것이다. 하지만 RNN의 경우는 은닉층의 결과가 다시 같은 은닉층의 입력으로 들어가게 되어있으므로 이런 특성이 순서 또는 시간이라는 측면을 고려할 수 있는 특징을 가져다 주게 되는데, 그래프의 값을 순서대로, 실시간으로 센싱하여 데이터를 학습해야 하기 때문에 무작위적인 방법을 사용하기 보다는 RNN의 알고리즘을 이용하여 데이터를 학습시키는 방법을 채택하였다.



[그림 9. LSTM 네트워크의 구조]

LSTM이란 Long Short Term Memory라고 불리는 특별한 종류의 RNN이다. LSTM은 장기 의존성 문제를 해결할 수 있는데 오랜 기간 동안 정보를 기억하는 일은 LSTM에 있어 특별한 작업 없이도 기본적으로 취하게 되는 기본 특성이다.



[그림 10. LSTM에 들어있는 4개의 상호작용하는 레이어가 있는 반복되는 모듈의 구조도]

[그림 10]과 같이 LSTM도 RNN과 마찬가지로 체인구조를 가지고 있는데 반복되는 모듈은 다른 구조를 가지고 있다. 이는 단일 뉴럴네트워크 레이어를 가지는 것 대신에, 4개의 상호작용 가능한 특별한 방식의 구조를 가지고 있다.

5. 결 과

데이터 학습을 위하여 트레이닝 데이터를 집어넣었다. 시스템을 구현하고 학습시킨 데이터는 약 개이며, 사용자의 정보를 센서로 인식받았을 때 정확도를 높이기 위해 더 많은 데이터를 집어넣고자 하였다. 학습 후에 얻은 데이터를 바탕으로 각각 정확도를 확인한 결과 [그림]와 같은 정확도를 보여주었다. 하지만 데이터를 다양한 사람에게서 얻을 수 없었던 점을 고려하여 사람에 따라 정도의 차이가 발생할 수 있으며, 이를 줄이기 위하여 보다 많은 학습 데이터가 필요할 것으로 요구된다.

2020_6_19_13_54_32_943175	458	469	469	469	481	481	481	470
2020_6_19_13_55_29_434557	531	531	288	288	288	234	234	234
2020_6_19_13_56_10_508870	464	454	454	454	486	486	486	448
2020_6_19_13_56_26_812623	415	415	441	441	441	444	444	444
2020_6_19_13_56_42_557447	432	432	432	453	453	453	466	466
2020_6_19_13_56_51_264175	466	470	470	470	466	466	466	453
2020_6_19_13_57_10_258788	558	558	558	499	499	499	621	621
2020_6_19_13_57_28_889571	448	503	503	503	427	427	427	436
2020_6_19_13_57_46_939367	280	280	280	411	411	411	444	444
2020_6_19_13_58_11_912674	529	529	454	454	642	642	642	524
2020_6_19_13_59_50_876397	448	552	552	552	457	457	457	459
2020_6_19_13_5_0_195088	415	231	231	231	423	423	418	418
2020_6_19_13_6_35_224897	447	445	445	445	477	477	466	466
2020_6_19_13_6_5_248992	462	462	462	304	304	304	424	424
2020_6_19_13_7_13_686431	445	445	445	447	447	447	254	254
2020_6_19_13_7_1_81415	432	316	285	285	368	412	412	421
2020_6_19_13_7_55_394439	447	447	456	454	443	443	443	453
2020_6_19_13_8_18_699541	463	463	486	486	503	503	503	485
2020_6_19_13_8_54_905560	447	447	467	467	467	454	454	454
2020_6_19_13_9_25_493132	415	415	415	448	448	460	460	460
2020_6_19_13_9_49_8932	442	442	486	486	486	464	464	464
2020_6_19_14_0_25_82388	506	506	506	447	447	447	410	410
2020_6_19_14_0_4_787971	453	453	453	442	442	442	445	445
2020_6_19_14_1_2_98215	541	541	541	461	461	461	446	446

[그림 11. ML을 위한 데이터 추출 과정]

[그림 11]과 같이 10개단위의 값을 끊어서 불러와 저장하고, 해당 데이터 값의 튀는 정도에 따라 True와 False라벨을 나누어 DB에 저장한다. True라벨은 일정한 주기의 그래프를, False라벨은 불안정한 그래프 주기를 보이며 해당 데이터들로 추후 ML을 진행하여 데이터의 정확도를 높여 결과적으로 신뢰성을 높이는 역할을 맡게 한다.

```

4/163 [.....] - ETA: 0s - loss: 0.0776 - acc: 1.0000
24/163 [====] - ETA: 0s - loss: 0.3450 - acc: 0.8750
44/163 [=====] - ETA: 0s - loss: 0.3267 - acc: 0.8489
64/163 [=====] - ETA: 0s - loss: 0.2741 - acc: 0.8750
84/163 [=====] - ETA: 0s - loss: 0.2698 - acc: 0.8810
104/163 [=====] - ETA: 0s - loss: 0.2963 - acc: 0.8846
124/163 [=====] - ETA: 0s - loss: 0.3065 - acc: 0.8871
144/163 [=====] - ETA: 0s - loss: 0.3176 - acc: 0.8819
Epoch 0015: saving model to best_model.h5

163/163 [=====] - 0s 3ms/sample - loss: 0.3224 - acc: 0.8650
123
123
123
127.0.0.1 - - [27/Jun/2020 17:50:53] "GET /request_data HTTP/1.1" 200 -
[[0.14732689]
[0.14761929]
[0.14761929]
[0.14425527]
[0.14425527]
[0.14425527]
[0.14425527]
[0.14532775]
[0.14532775]
[0.14532775]
[0.14562136]
[0.14655462]
[0.14423287]
[0.1437029 ]
[0.1437029 ]
[0.1437029 ]
[0.1437029 ]
[0.1437029 ]
[0.1437029 ]
[0.1450584 ]
[0.1450584 ]

```

[그림 12. RNN알고리즘을 통한 ML진행]

[그림 11]과 같이 추출된 데이터를 이용하여 ML을

진행한다. 학습된 데이터는 기존 학습하기 전 최근 데이터를 불러와 학습된 데이터와 비교하여 결과값을 제공한다.

I. 결과값

"당신이 진실을 말했을 확률은 14.57% 입니다!"
 "답변에 영혼이 없었네요.. 😊"

II. 상세 결과 보기

GSR 테이블

피부전기자극정보	측정 값	측정결과
GSR_1	258	[0.1450999528169632]
GSR_2	486	[0.14560484886169434]
GSR_3	486	[0.14560484886169434]
GSR_4	256	[0.14627550542354584]
GSR_5	256	[0.14627550542354584]
GSR_6	256	[0.14627550542354584]
GSR_7	498	[0.1460520476102829]
GSR_8	498	[0.1460520476102829]
GSR_9	498	[0.1460520476102829]
GSR_10	504	[0.1448694765567795]

HRT 테이블

심박정보	측정 값	측정결과
HRT_1	51	[0.14793285727500916]
HRT_2	83	[0.14552775025367737]
HRT_3	83	[0.14552775025367737]
HRT_4	83	[0.14552775025367737]
HRT_5	83	[0.14552775025367737]
HRT_6	83	[0.14552775025367737]
HRT_7	83	[0.14552775025367737]
HRT_8	63	[0.14505040645599365]
HRT_9	63	[0.14505040645599365]
HRT_10	63	[0.14505040645599365]

[그림 13. ML적용 후 데이터 값 비교에 따른 결과]

[그림 13]은 LSTM의 알고리즘에 따라 기존 학습된 데이터를 이용한 트레이닝 데이터와 최근 UI를 통해 수집된 사용자의 데이터를 비교하여 데이터를 나타내주고, 확률 계산까지 마친다. %의 수치는 학습된 데이터가 많을수록 더 정확한 결과 값에 근접할 것이다.

6. 결론

본 논문은 거짓말 탐지기라는 주제로 4차 산업혁명 시대에 걸맞는 기술인 인공지능기술을 탑재하여 제작된 작품이다. 단순 센서로 감정을 판단하여 나타내어진 그래프로 거짓말을 판단하는 기존의 방식과 비슷하지만, 보다 정확한 결과 값을 위한 머신러닝 기술을 등재하였고, 현재는 단순한 하드웨어에 머신러닝 기술을 접목한 프로그램이지만, 추후 해당 프로그램, 작품과 같이 머신러닝을 결합한 기기들이 등장할 수 있는 발판 또는 계기가 되었으면 한다. 사람들의 편의성을 위하여 개발되고 있는 로봇 또한 초창기에는 단순 작은 병뚜껑을 따주거나 맥주를 대신 따라주는 등 단순한 행동에서 이제는 사람들을 대신하여 위험한 일을 해주는 등, 일상생활에서 많은 편의를 제공하고 있다. 이처럼 인공지능 기술 또한 무수한 실패를 거듭하여 발전하고 있는 상황에서 미래

기술의 작은 발판이 되었으면 하는 바이다.

참고 문헌

- [1] Hoan-Suk Choi, Oian Peng, Woo-Seop Rhee
“Design and Implementation of the Machine
Learning-based Restaurant Recommendation
System” 디지털콘텐츠학회논문지 Vol. 21, No. 2, pp.
259-268, Feb. 2020

- [2] Yong-Geon Lee, Jae-Young Oh, Gibak
Kim “Interpretation of Load Forecasting Using
Explainable Artificial Intelligence Techniques” , The
Transactions of the Korean Institute of Electrical
Engineers, Vol. 69, no. 3, pp. 480~485, 2020

- [3] DongMin park, GyengTae Kim, GiHo Lee, Hwang
Gi-Hyun “A study on the Development of A Lie
Detector Using Arduion System” , 2015년
대학전자공학회 하계학술대회 논문집 2015.6,
1600-1601(2pages)