



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

가속도센서와 자이로센서를 활용한
기계학습 기반 고위험군 종사자의
활동성 모니터링



2023년 8월

부 경 대 학 교 대 학 원

인 공 지 능 융 합 학 과

우 종 석

공 학 석 사 학 위 논 문

가속도센서와 자이로센서를 활용한
기계학습 기반 고위험군 종사자의
활동성 모니터링

지도교수 정 완 영

이 논문을 공학석사 학위논문으로 제출함.

2023년 8월

부 경 대 학 교 대 학 원

인 공 지 능 융 합 학 과

우 종 석

우종석의 공학석사 학위논문을
인준함.

2023년 8월 18일



위원장	공학박사	정 상 중 (인)
위원	공학박사	Lionel Nkenyereye (인)
위원	공학박사	정 완 영 (인)

목 차

그림 목차	iii
표 목차	v
Abstract	vi
 I. 서론	 1
1.1. 연구 배경과 필요성	1
1.2. 연구 목적	3
1.3. 연구의 구성	4
 II. 배경 이론	 5
2.1. 낙상 감지	5
2.2. 학습과 모델	7
2.2.1. 기계학습과 딥러닝	7
2.2.2. 알고리즘과 모델의 생성	8
2.2.3. 모델 평가의 선행 연구	14
 III. 연구 방법	 16
3.1. 시스템 구성	16
3.1.1. PCB 설계	16
3.1.2. 가속도센서와 자이로센서	18
3.2. 데이터 수집 및 처리	20
3.2.1. 실험 환경	20
3.2.2. 각도 측정	25
3.2.3. 데이터 증강	26
 IV. 결과 분석	 30
4.1. 모델 설계와 평가 지표	30
4.1.1. 예측 모델 설계	30

4.1.2. 평가 지표	31
4.2. 모델 평가	33
4.2.1. 환경 구성 및 하이퍼 파라미터의 설정	33
4.2.2. 모델별 설정과 성능 평가	35
V. 결론	45
참고문헌	47



그림 목차

그림 1.1. Status of Disaster Victims by Type of Disaster in 2022	1
그림 1.2. Fireground Injuries by Cause: 2021	2
그림 2.1. Machine Learning and Model Creation	7
그림 2.2. Support Vector in Two-Dimensional Space	9
그림 2.3. Structure of CNN	10
그림 2.4. Basic structure of LSTM unit	11
그림 2.5. Architectural elements in a TCN	13
그림 3.1. Structure of the System	16
그림 3.2. Wearable sensor module PCB	17
그림 3.3. Equipment Attached Status and Location	21
그림 3.4. Overall View of the Belt and Attached Device	21
그림 3.5. Falling movements	23
그림 3.6. Falling data	24
그림 3.7. Defining Fall Range by Comparing Changes in Angle and Magnitude of Acceleration	25
그림 3.8. Comparison of Augmented Data	29
그림 4.1. Creation and Evaluation of the Model	30
그림 4.2. Confusion matrix	32
그림 4.3. Composition of the Dataset	35
그림 4.4. Confusion matrix of SVM	36
그림 4.5. Structure of 1D CNN model	37
그림 4.6. Result of 1D CNN	38
그림 4.7. Structure of LSTM model	39
그림 4.8. Result of LSTM	40
그림 4.9. Structure of CNN-LSTM model	41

그림 4.10. Result of CNN - LSTM	42
그림 4.11. Structure of TCN model	42
그림 4.12. Result of TCN	43



표 목차

표 3.1. Parts specification	18
표 3.2. Action Classification	22
표 4.1. Development Environment	33
표 4.2. Setting Common Parameters	34
표 4.3. SVM result according to C	35
표 4.4. CNN result according to learning rate	37
표 4.5. LSTM result according to learning rate.....	39
표 4.6. CNN-LSTM result according to learning rate	40
표 4.7. TCN result according to learning rate	42
표 4.8. Performance evaluation score using Confusion matrix based on Recall	43
표 4.9. Performance evaluation score using Confusion matrix based on F1-score	44

Abstract

Falls are unexpected occurrences during daily activities that lead to significant difficulties in life. Not only do falls frequently happen among the elderly, but they also account for a high percentage of accidents in industries and hazardous occupations. For people who working in high-risk jobs such as firefighters or even working in industrial site, the rate of accident realated with falls are more than 20%.

To address these incidents caused by falls, research related to fall detection is actively being conducted. This study aimed to develop and evaluate various models that classify whether a fall has occurred when given measured data using a device attached to body which is equipped with an accelerometer and a gyroscope to measure people's movements. The goal was to ascertain which model performed best through performance evaluations.

To gather data, an experiment was conducted where devices were attached to the waists of participants who repeatedly performed actions defined as falls and non-falls.

Collated data tested by 5 models which were Support Vector Machine (SVM), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Temporal Convolutional Networks (TCN), and CNN-LSTM model. Performance evaluations were carried out while maintaining some of the hyperparameters the same for the deep learning models, to make the appropriate model selection. Among the implemented models, the LSTM model showed the most superior performance on the configured dataset.

I. 서론

1.1. 연구 배경과 필요성

낙상은 일상의 활동 중에 예기치 않게 발생하여 생활에 많은 어려움을 초래한다. 따라서 이러한 불의의 사고를 예방할 수 있어야 하겠지만, 실제로 그러한 사고가 발생했다면 상황을 빠르고 정확하게 판단함으로써 적절히 대처할 수 있어야 한다.

낙상은 산업재해에서 큰 비율을 차지한다. 다음 그림 1.1.의 2022년 우리나라의 산업재해 발생 현황을 보면, 전체 재해자 130,348명 중에서 넘어짐(25,084명, 19.2%)과 떨어짐(14,387명, 11.0%)의 비율이 여타 부분보다 높게 나타났음을 알 수 있다[1].

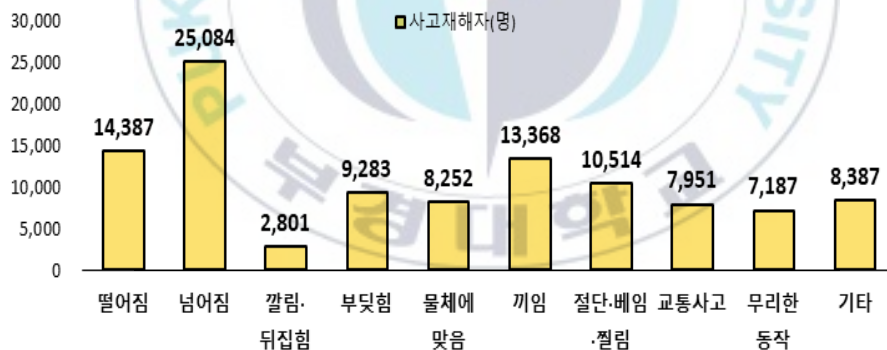


그림 1.1. Status of Disaster Victims by Type of Disaster in 2022.

그리고 같은 자료에 따르면, 이러한 낙상과 관련되는 사고에 의한 산업재해는 건설업(53%)과 제조업(24%) 등의 고위험 직종에서 주로 발생하는 것으로 나타났다.

또한 소방관들의 방재 작업 안전성 개선을 위해서도 낙상의 문제가 검토

되어야 한다. 우리나라의 경우, 2021년도에 36,267건의 화재가 발생하였는데, 인명 피해로는 사망자 276명, 부상자 1,854명이었으며, 장소별로는 사망자 190명(68.8%)과 부상자 945명(51.0%)이 주거 시설에서 발생하였다. 그리고 화재에 대처했던 소방공무원들의 피해에서 최근 5년(2017-2021)의 소방공무원 순직·공상자 현황을 보면 순직 23명, 공상 4,235명에 이르렀다 [2].

미국의 경우로 보면, 다음 그림 1.2.에서처럼 소방관들의 부상 중에서 낙상과 관련한 사고(fall, jump, slip, trip)가 차지하는 비율이 24%로, 다른 요인들에 비해 높은 것으로 나타났다[3].

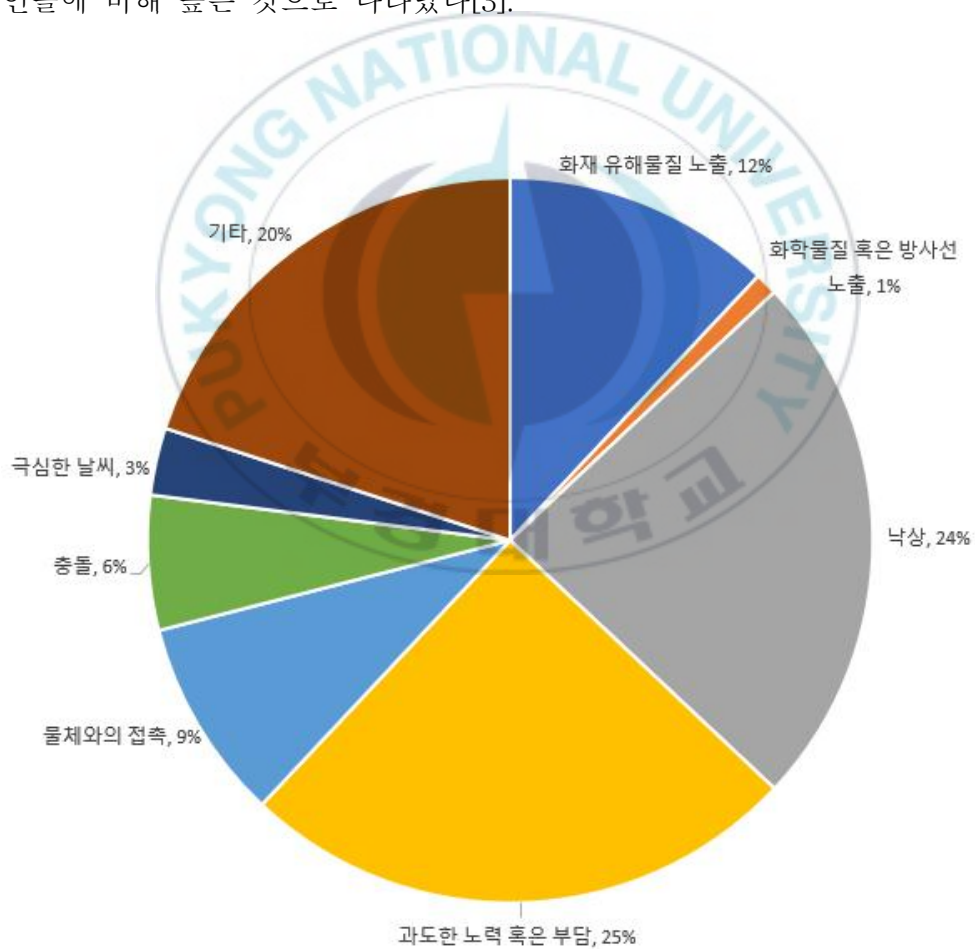


그림 1.2. Fireground Injuries by Cause: 2021.

최근의 낙상과 관련한 연구는 노인들의 의료 사고를 중심으로 진행되어 왔다. 물론 낙상은 노인에게서 발생률이 높은 것은 사실이지만[4], 모든 연령에서 발생할 수 있는 것이기도 하다. 특히 건설업이나 제조업 등의 산업 현장 종사자를 비롯하여 소방관과 같이 고위험 직종에 종사하는 사람들은 낙상 사고의 위험에 크게 노출되어 있다. 이에 따라 이들을 대상으로 하는 낙상 사고를 미연에 방지하고, 불의의 사고로 인한 여러 문제를 최소화하기 위한 연구의 필요성이 증가하고 있다[5].

1.2. 연구 목적

낙상으로 인한 사고를 예방하거나 사고 발생에 대처하기 위해서는 낙상을 유효하게 감지할 수 있어야 한다. 낙상을 감지하기 위해서는 행동과 신체적 균형을 지속적으로 관찰해야 하는데, 이를 위해 선행 연구에서는 영상 기기를 이용하거나 낙상 이후 발생하는 바닥의 진동을 측정하여 분석하기도 하였다. 그런데 이 방법은 감지 기기를 설치하는데 공간상의 제약 등의 문제들이 지적되었다. 최근에는 낙상에 대한 연구가 심화되면서 인공지능에 기반을 두고 웨어러블(Wearable) 기기와 같은 센서를 통해 추출한 데이터를 기반으로 낙상을 감지하는 시스템들이 개발되고 있다. 이러한 시스템들은 대부분 웨어러블 기기 안에 포함된 가속도센서나 자이로센서를 이용하여 낙상을 검출하는 것이 일반적이다.

기계학습(Machine Learning) 기반의 낙상에 관한 선행 연구에서는 의사결정나무, 랜덤 포레스트, SVM, 로지스틱 회귀분석, 배깅 등이 다양하게 활용되었으며, CNN이나 RNN 등과 같은 딥러닝(Deep Learning) 기반의 연구도 진행되어 왔다. 물론 이들에서 많은 성과가 있었으나, 연구에 활용된 해당 데이터 집합에 한정하여 적용될 수 있는 등의 환경과 대상에 대한 제약이 있었다.

본 연구는 건설업과 제조업, 소방 등과 같은 고위험 직종 종사자들의 낙

상 감지를 위한 예측 모델을 생성하고 그 유효성을 검증하는 것을 목적으로 하였다. 이를 위해 가속도센서와 자이로센서를 통해 추출한 데이터를 기반으로 가속도 신호와 방위각을 산출하여 낙상 여부를 감지하는 시스템을 구성하였다. 그리고 사용자에게 낙상 감지 장치를 복부에 착용하고 주어진 동작을 수행하게 하고 그로부터 사용자의 활동성을 모니터링하여 데이터를 수집하였으며, 수집된 데이터를 처리하여 대표적인 기계학습으로서의 SVM과 딥러닝 모델로서의 1D CNN, LSTM, CNN-LSTM, TCN 등에 근거한 낙상 감지 예측 모델들을 생성하고 그 성능을 측정하고 비교하여 유효성을 평가하였다. 그리하여 본 연구는 기계학습과 딥러닝 기반의 낙상 감지에 대해 포괄적으로 이해하며, 특히 고위험군 직종의 종사자들이 겪을 수 있는 낙상을 감지하기 위한 최적의 모델을 설정하는 데 기여할 것으로 기대된다.

1.3. 연구의 구성

본 연구는 다음과 같이 구성하였다. I 장에서는 본 연구에 대한 배경과 필요성을 서술하고 연구의 목표를 설정하였다. II 장에서는 낙상의 기본 개념을 이해하고 기계학습과 딥러닝 기반의 낙상 감지 시스템 구현을 위한 관련 이론에 대해 간략히 서술하였다. III 장에서는 3축 가속도센서와 자이로센서를 활용하여 사용자의 신체 활동을 운동량 정보로 변환하고 낙상 여부를 판단하는 시스템을 구현하였다. 그리고 수집한 데이터를 정량화하고 예측 모델에서 활성화되도록 처리하였다. IV 장에서는 낙상 감지를 위한 SVM과 1D CNN, LSTM, CNN-LSTM, TCN 모델을 생성하여 이들의 유효성을 검증하고 성능을 비교하였다. 마지막으로 V 장에서는 본 연구의 결론과 추후 과제에 대한 의견을 제시하고 연구를 마무리하였다.

II. 배경 이론

2.1. 낙상 감지

낙상은 본인의 의사와 상관없이 신체가 조절되지 않아 높은 곳에서 떨어지거나 미끄러져 넘어지는 동작을 말한다[4]. 이것을 신체의 움직임과 관련해서 보면, 낙상은 균형을 잃고 넘어져서 발바닥 이외에 손이나 둔부 등의 신체 일부가 지면 또는 바닥에 닿는 것이라 할 수 있다.

과거에는 낙상이 발생하면 위험 상황을 알리기 위해 본인이 버튼을 누르는 기기를 사용하기도 하였다. 그런데 이것은 넘어진 사람이 버튼을 누를 수 없는 상황에서는 의미가 없었다. 그리하여 다른 방식으로 낙상을 감지하는 시스템의 구현과 그에 따른 알고리즘의 개선을 통해 감지 정도를 높여주는 다양한 연구들이 이어져 왔다[6].

낙상 감지 시스템은 크게 상황 인식(Context-awareness) 시스템과 웨어러블 장치(Wearable device) 시스템으로 나뉜다[7]. 상황 인식 시스템은 주변 센서를 이용하여 낙상을 감지하는 방법으로 카메라 등의 영상 정보를 활용하는 방식이 많으며[8], 적외선 센서를 이용하거나[9], 바닥에 설치된 센서를 통한 진동이나 소리를 이용하는 방식이 있다[10,11].

영상 정보를 활용하는 것은 주변 환경에 장비를 설치하여 사용자의 상황을 파악하는 방법으로, 사용자를 신체적으로 구속하지 않아도 된다는 장점이 있다[12]. 그러나 이 방법은 실내로 제한되거나 측정의 사각지대에서 발생하는 것은 감지하지 못하고 사생활이 노출될 수도 있으며, 경험을 통해 임계점을 설정하고 이를 통해 낙상을 감지하는 것이기 때문에 규칙 분석이나 특징 추출이 존재하지 않아 일반화되기 어렵다는 한계가 있다[13]. 진동에 기반한 감지는 일상 활동에서 발생하는 진동과 크게 다른 패턴을 감지하는 것을 기반으로 한다. 이 방법은 개인의 사생활을 보호하지만, 바닥의 역학적 상황에 따라 효과가 다를 수 있다는 한계가 있다.

웨어러블 장치 시스템은 신체에 착용한 장치를 통해 신체 상황을 검증하는 것으로, 기술적 관점에 따라 크게 휴대형(Portable), 부착형(Attachable), 이식/복용형(Eatable)으로 분류된다[14]. 이 시스템은 실내외의 공간적인 제약이 없으며, 신체의 구속 정도와 밀착 정도는 다양하지만 비교적 자유로운 신체 활동이 가능하다는 장점이 있다.

웨어러블 장치 시스템에서는 일반적으로 가속도센서와 자이로센서를 이용하여 낙상을 판별한다. 가속도센서를 이용하는 것이 대표적인데, 선행 연구에서는 허리에 가속도센서를 부착하고 신체의 운동 패턴을 분석하여 낙상을 감지하는 알고리즘을 제안하거나[15,16], 휴대 전화에 가속도센서를 연결하는 방법이 제안되기도 하였다[17]. 그리고 가속도센서를 귀에 부착하는 형태가 제안되기도 하였는데[18], 이것은 안경을 쓰고 있는 사람의 경우 부착이 어려우며 이질감이 생길 수 있다는 단점이 있다. 한편, 자이로센서를 가슴 부위에 부착하여 각도의 변화값을 알아낸 뒤 임계값과 비교하여 낙상을 감지하는 기법이 활용되기도 하였는데[19], 이도 역시 대상자에 따라서는 착용에 어려움이 제기될 수 있다. 물론 가속도센서와 자이로센서를 함께 착용하는 경우도 있는데, 이것은 가속도센서에 각속도의 측정이 가능한 자이로센서를 보완하여 낙상 감지의 정확도를 높이는 효과가 있다.

한편, 이런 방식들을 활용하여 구성된 삼성 갤럭시 워치와 노인용 낙상 감지기 등과 같은 시제품들이 다양한 방면에서 실생활에 쓰이고 있다. 그런데 이런 제품들의 경우 다목적으로 만들어졌거나 행동에 장애를 초래하기도 하기 때문에, 특정한 동작을 수행하게 되는 사용자들에게 적합하도록 특수한 목적으로 구성되는 낙상감지 장치가 요구된다.

2.2. 학습과 모델

2.2.1. 기계학습과 딥러닝

기계학습(Machine Learning)은 ‘컴퓨터에 명시적인 프로그램 없이 배울 수 있는 능력을 부여하는 연구 분야’로[20], 여기서는 인간이 학습하듯이 컴퓨터에도 데이터들을 제공하여 학습하게 하고 그로부터 규칙이나 새로운 지식을 추출해 내는 것을 연구한다. 이에 따르면, 어떤 사물의 구조적 특징이나 원형 패턴을 정의하지 않고 데이터 인식이 가능한 알고리즘에 일련의 데이터를 제공하면, 알고리즘이 주어진 데이터로부터 학습을 통해 새로운 지식과 규칙을 추출하여 예측 모델을 생성하는데, 이 모델에 새로운 데이터를 입력하면 학습한 지식과 규칙을 적용하여 새로운 출력을 구성해 내는 것으로 해석한다. 이것은 다음 그림 2.1과 같이 정리된다.

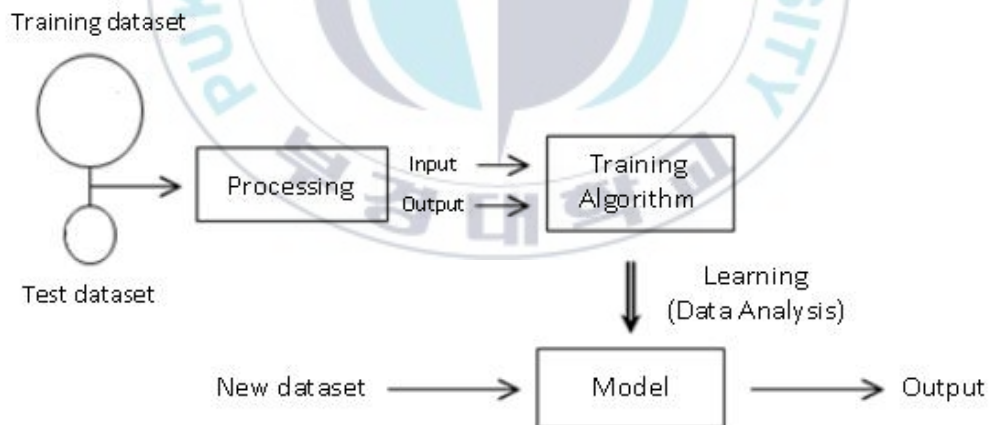


그림 2.1. Machine Learning and Model Creation.

여기서 학습 알고리즘(Algorithm)은 어떤 문제를 해결하기 위해 정해진 일련의 절차나 방법을 모아 놓은 것을 뜻한다. 기계학습에서 알고리즘은 주어진 데이터를 분석하고 정보를 축적하는 과정으로 학습을 수행하며, 그

를 바탕으로 새롭게 주어지는 데이터에 대한 예측이나 결정을 도출하는 특정한 모델(Model)로 생성된다. 따라서 모델은 알고리즘이 데이터를 통해 학습한 결과로 구현된 객체를 의미하는데, 때로는 서로 통용되기도 한다.

기계학습에서 적용될 수 있는 알고리즘은 학습의 유형에 따라 다를 수 있다. 학습 유형에는 지도 학습과 비지도 학습, 강화 학습이 있으며, 낙상과 관련하여 연구의 대상이 되었던 대표적인 알고리즘에는 k-NN, 선형회귀, 의사결정나무, 랜덤 포레스트, 나이브 베이즈 분류, SVM 등이 있다. 최근에는 컴퓨터가 사람의 뇌처럼 사물을 분류하도록 훈련시키는 이른바 인공신경망에 대한 연구가 심화되면서 그에 따라 다양한 알고리즘으로 발전하고 있다. 여기서는 입력층과 출력층 사이에 은닉층(Hidden layer)을 두며, 각 계층의 공간들을 왜곡과 합을 반복하는 과정으로 데이터를 처리한다. 이들은 낙상 감지에도 적용되고 있는데, 대체로 DNN과 CNN, RNN, AE, GAN, UNET 등으로 분류될 수 있다[21].

2.2.2. 알고리즘과 모델의 생성

앞에서와 같이 낙상을 감지하기 위해 여러 알고리즘들이 적용되었다. 본 연구에서는 대표적인 분류 알고리즘인 SVM과 영상 처리 모델인 1D CNN, 시계열 데이터 처리 모델인 LSTM, 이들의 결합형인 CNN-LSTM, 그리고 CNN의 일종인 TCN을 다루었는데, 이들에 대해 간략히 살펴본다.

SVM(Support Vector Machine)에서는 유클리드 거리를 최소화하여 매개변수 벡터를 최적화하고 이를 통하여 클래스를 분류하는 초평면(Hyperplane)을 찾는다. 이 과정에서 각 분류 클래스에 해당하는 변수 중 초평면과 가장 인접한 지점이 서포트 벡터(Support vector)이고, 서포트 벡터 사이에 해당하는 마진(Margin)을 통해 결정경계(Decision boundary)를 구하게 된다.

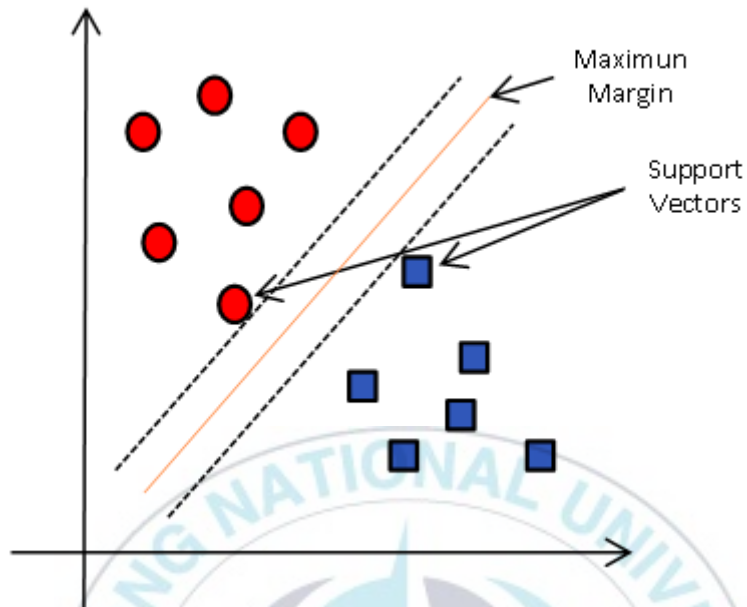


그림 2.2. Support Vector in Two-Dimensional Space.

SVM에서 마진이 넓을수록 새로운 데이터가 잘 구별될 확률이 높기 때문에 SVM의 목표는 마진을 최대화하는 것이며, 아무리 많은 데이터가 있다고 해도 SVM은 서포트 벡터만을 기준으로 결정경계를 설정하고 분류가 진행되므로 고차원 데이터의 분류 문제에 좋은 성능을 보이며 계산량이 다른 모델에 비해 적다는 장점이 있다.

CNN(Convolutional Neural Network)은 인공신경망의 한 종류로 다양한 합성곱 필터(convolution filter)를 사용하여 이미지의 특징(feature)을 추출한다. CNN은 입력층과 출력층 사이에 다수의 은닉층으로 구성되어 있으며, 이들은 합성곱층(Convolution layer)과 풀링층(Pooling layer)으로 구분된다. 합성곱층은 필터와 입력 데이터의 합성곱 연산을 통해 특정 맵을 생성하고, 데이터가 줄지 않도록 데이터의 가장자리를 채워준다. 풀링층은 일정 영역의 정보를 축약하는 역할을 하며, 전체 연결층(Fully connected network layer)에서 입력으로 사용될 유용한 정보를 추출한다. 여기서 각

층을 통하여 이미지의 특징을 추출해냄으로써 정확도를 높여 나간다. 다음 그림 2.3.은 CNN의 기본적인 구조를 보여 준다.

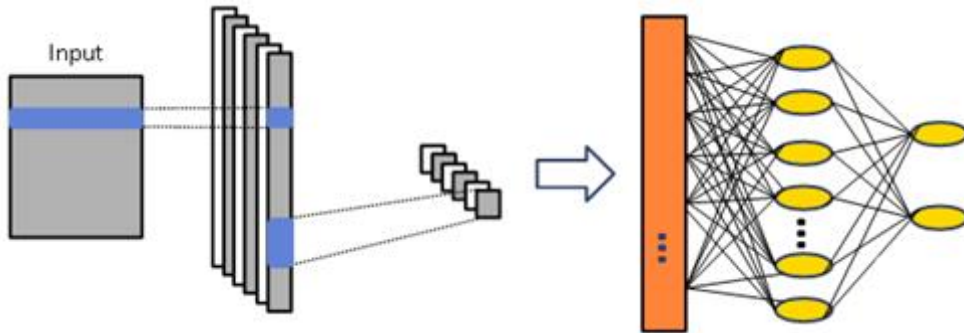


그림 2.3. Structure of CNN.

CNN은 이미지 분석을 위한 딥러닝에서 널리 사용되는데, 다양한 필터를 통해 이미지의 특징을 추출하고 추출된 특징들을 사용하여 객체 탐지 (Object detection), 이미지 분할(Semantic segmentation), 이미지 분류 (Classification)를 수행한다. CNN의 성능을 향상시키는 방법으로는 신경망을 더 깊게 만들거나 입력 이미지의 크기를 키우는 것이 있다.

LSTM은 시간적인 순서가 있는 RNN(Recurrent Neural Network)의 일종으로 시계열 데이터와 같은 연속적인 정보를 처리하는 데 특화된 인공신경망 구조이다. 기존의 RNN은 장기 의존성 문제로 인해 긴 시계열 정보를 처리하는 데 어려움이 있었는데, LSTM은 입력 게이트(Input gate)와 망각 게이트(Forget gate), 출력 게이트(Output gate)를 통해 정보의 흐름을 조절하고 불필요한 정보를 삭제하며 필요한 정보를 유지한다. 다음 그림 2.4.는 LSTM 구조를 보여 준다.

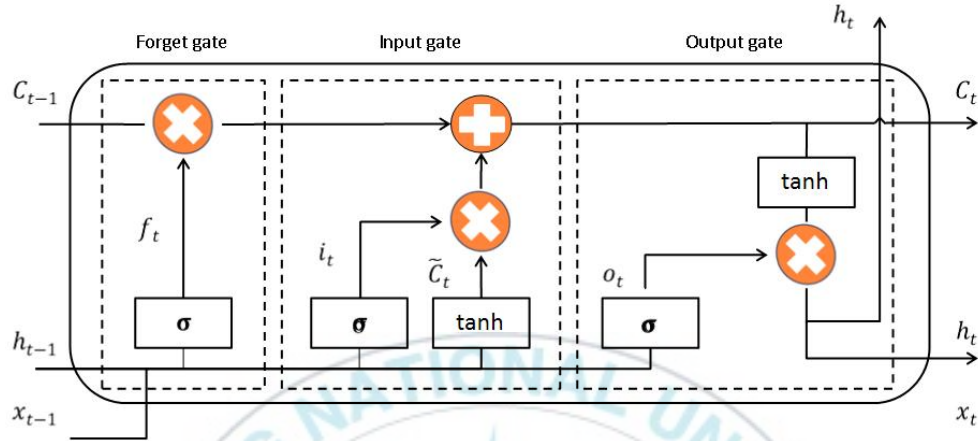


그림 2.4. Basic structure of LSTM unit.

그리고 LSTM의 작동 과정은 다음 수식으로 설명된다.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.4)$$

$$h_t = o_t * \tanh(C_t) \quad (2.5)$$

위 그림 2.4.에서 왼쪽 첫 번째가 망각 게이트인데, 여기서는 이전 상태 셀에서 어떤 정보를 기억할지 결정하기 위해 시그모이드 함수 σ 가 사용된다. 이 절차는 식 (2.1)과 같다. 여기서 W_f 와 b_f 는 훈련 후에 결정될 파라미터이며, h_{t-1} 은 시간 epoch t-1에서 숨겨진 상태이고, x_t 는 시간 epoch에서의 입력 벡터이다.

다음의 입력 게이트는 입력을 제어하고 새로운 정보의 어떤 부분이 현재 셀 상태에서 저장될 것인지를 결정하는데, 이 절차에는 식 (2.2, 2.3)이 사

용된다. 여기서 W_i 와 b_i , W_c , b_c 는 훈련 과정을 통해 결정되는 파라미터이고, h_{t-1} 은 시간 epoch $t-1$ 에서 숨겨진 상태이며, x_t 는 입력 벡터이다.

그리고 출력 게이트는 출력을 결정하고 제어하며, 식 (2.4, 2.5)와 같이 sigmoid 함수와 tanh 함수로 구성된다. 여기서 W_o 와 b_o 는 훈련 중 결정되는 파라미터이고, C_t 는 시간 t 에서의 셀 상태이다. o_t 는 sigmoid 함수의 출력으로 숨겨진 상태의 출력을 결정한다. 그리고 셀 상태는 tanh 함수를 통해 주어지는데 벡터 o_t 와 곱해져서 LSTM의 출력을 결정한다.

CNN-LSTM은 수집된 데이터를 1D CNN 알고리즘의 입력값으로 사용하여 나온 출력값을 LSTM 알고리즘의 입력값으로 취하는 결합형이다. CNN은 합성곱층(Convolution layer)을 적용하여 특징의 자동 추출과 대규모 병렬 처리가 가능하고, LSTM은 대규모 병렬 처리가 가능하지 않지만 원하는 시기에 진행하거나 제어할 수 있어서 CNN의 장기 의존성 문제를 해결할 수 있다. 따라서 CNN-LSTM의 조합 모델을 이용하게 되면 양자의 장점을 활용하고 단점을 보완하여 시공간적인 특징을 모두 학습할 수 있다.

TCN(Temporal Convolutional Network)은 CNN의 일종으로 인과적 합성곱(Causal convolution)과 팽창 합성곱(Dilated convolution)을 사용하여 시계열 데이터의 시간적 맥락(Temporal context) 정보를 효과적으로 추출할 수 있는 것으로 평가받는다. 이 두 개의 변형된 합성곱을 함께 사용하여 과거의 정보를 추출하고 그 정보의 인과관계를 분석하여 결과를 얻을 수 있다. 특히 팽창 합성곱은 각 은닉층의 추출 간격을 조절하여 더 넓은 입력 수용 범위를 갖게 함으로써 더 많은 과거의 정보를 추출할 수 있다. TCN의 구성을 보면 그림 2.5.과 같다.

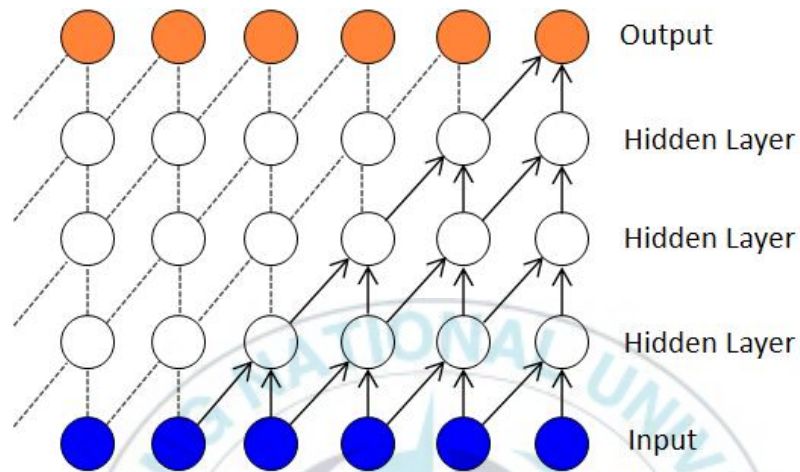


그림 2.5. Architectural elements in a TCN.

각 TCN 레이어는 여러 컨볼루션 레이어로 구성되며, 이 컨볼루션 레이어들은 팽창 합성곱을 사용합니다. 이 팽창 합성곱은 컨볼루션의 범위나 receptive field를 확장하면서 따로 파라미터 수를 늘리지 않는 방법입니다. 즉, 각 레이어는 점점 더 넓은 범위의 입력 시퀀스를 고려할 수 있게 됩니다.

TCN은 RNN 계열의 모델들과 다르게 병렬화가 가능하고, 비교적 안정적인 gradient를 제공한다. 그리고 필터의 크기와 층의 깊이뿐만 아니라, dilation factor의 제어와 receptive field의 크기 조절이 가능하여 자유롭게 메모리 크기를 조절하고 수용 범위를 유연하게 늘릴 수 있어서 특징을 더 유용하게 추출할 수 있다.

2.2.3. 모델 평가의 선행 연구

낙상 감지와 관련하여 기계학습과 딥러닝 기반의 모델들의 성능을 비교하는 선행 연구가 있었다. 이들은 대부분 기계학습 모델을 대상으로 하여 정확도와 정밀도, 재현율, F1-score 등을 기준으로 하는 분류 모델의 평가 방식이 적용되었다. 우선 참고문헌 [22]에서는 실측 데이터에 낙상 감지 학습 능력을 정량화한 결과 F1-score에서 의사결정나무 0.77, 랜덤 포레스트 0.85, SVM 0.86으로 나타나 SVM이 제일 좋은 성능을 보였다고 하였다. 그리고 [23]에서는 다중선형회귀분석, 의사결정나무, 랜덤 포레스트, 딥러닝을 대상으로 하였으며 성능 향상을 위해 k-fold 교차 검증 방식을 적용하였는데, 분류 정확도는 각각 0.58, 0.56, 0.78, 0.84로 딥러닝이 가장 높게 나타났다. 이에 비해서 [24]에서는 정확도의 측면에서 보면, 로지스틱 회귀분석 94.20%, SVM 93.30%, 의사결정나무 87.50%, 랜덤 포레스트 95.30%로 나타나서 랜덤 포레스트가 가장 우수한 성능을 보였다고 하였다.

또한 참고문헌 [13]에서는 기계학습 모델인 의사결정 나무, 랜덤 포레스트, SVM 등과 함께 1D CNN을 적용하여 낙상 감지를 검증하였는데, SVM의 성능이 다른 기계학습 모델에 비해 가장 높았으나 1D CNN의 검출 성능이 SVM에 비해 모든 지표에서 평균 7.5% 더 높았다고 하였다. 그리고 [25]에서는 CNN-LSTM과 이것에서 나타나는 degradation 현상을 해결하여 새롭게 구성한 ResNetCNN-LSTM 모델을 구현하고 이들의 정확도를 비교하였으며, 이들은 각각 99.81%, 98.17%로 약간의 차이는 있으나 모두 성능이 우수한 것으로 평가하였다. 또한 [26]에서는 전이 학습으로 학습했을 때, GRU(Gated Recurrent Unit)는 LSTM보다 학습 속도가 빨랐으며, 정밀도와 재현율, F1-score, 정확도 등 모든 면에서 우수한 성능을 보였으며, 특히 정확도에서 LSTM 0.9670, GRU 0.9696으로 나타났다고 하였다.

한편, 낙상 이외의 다른 상황을 대상으로 모델을 평가한 연구도 있었다. 참고문헌 [27]에서는 영화 리뷰의 감성분석을 딥러닝 기법 알고리즘인

CNN과 LSTM의 결합을 제안하고 성능을 평가하였는데, 정확도에서 CNN 89.04, LSTM 86.88, CNN-LSTM 90.33%로 나타나 CNN-LSTM이 가장 안정적이고 정확하였다고 하였다. 또한 RMSE 등의 회귀 모델 평가 방법을 활용한 경우를 보면, [28]에서는 미국항공우주국에서 베어링의 성능 저하 예측 연구를 목적으로 제공하는 데이터 세트를 사용하여 모델들을 평가하였는데, 정확성에서 TCN이 CNN, LSTM, CNN+LSTM보다 평균적으로 각각 30.8%, 47.5%, 23.3% 좋았다고 하였다. [29]에서는 금 가격 예측을 위한 모델을 비교하였는데, SVM, TCN, LSTM, CNN-LSTM 중에서 TCN이 가장 우수한 성능을 나타내었다고 하였다. 그리고 [30]에서는 홍수 관리를 위한 예측에서 LSTM-GRU가 가장 우수하였고 다음으로 Multi-LSTM, Multi-GRU 순으로 나타났다고 하였다.

본 연구는 고위험군 직종의 종사자들이 겪을 수 있는 낙상을 감지하기 위한 최적의 모델을 설정하는 것을 목적으로 하였다. 이에 따라 선행 연구에서 비교적 우수한 것으로 평가된 예측 모델들 중에서 분야에 따라 특정 모델을 선정하고, 이들의 낙상 감지 능력을 평가하였다. 구체적으로는 전통적 기계학습 분야에서 대표적인 분류 알고리즘인 SVM을 선정하였고, 딥러닝 분야에서 영상 처리를 특징으로 하는 1D CNN과 시계열 데이터 처리에 우수한 LSTM, 이 둘을 결합한 CNN-LSTM, 그리고 CNN의 일종인 TCN을 성능의 평가와 비교의 대상으로 선정하였다.

Ⅲ. 연구 방법

3.1. 시스템 구성

3.1.1. PCB 설계

사람의 동작에 대한 인식 및 인지하는 방식은 다양하게 도출할 수 있는데 크게 카메라를 이용하는 방식과 자이로센서와 가속도센서를 활용하는 방식으로 나뉜다. 자이로센서와 가속도센서를 활용하는 방식에는 웨어러블 장치에 포함된 센서를 통해 각각의 동작에 따라 발생하는 3차원 공간상의 위치에 대한 값을 수집한다.

본 연구에서는 고위험 직종 종사자들의 활동 현장에서 낙상 사고가 발생했을 때, 신속하게 사고를 인지하고 필요한 조치를 취하는 데 유용하도록 현장에 적용할 수 있는 낙상 감지 시스템을 구현하였다. 이것은 그림 3.1.과 같다.

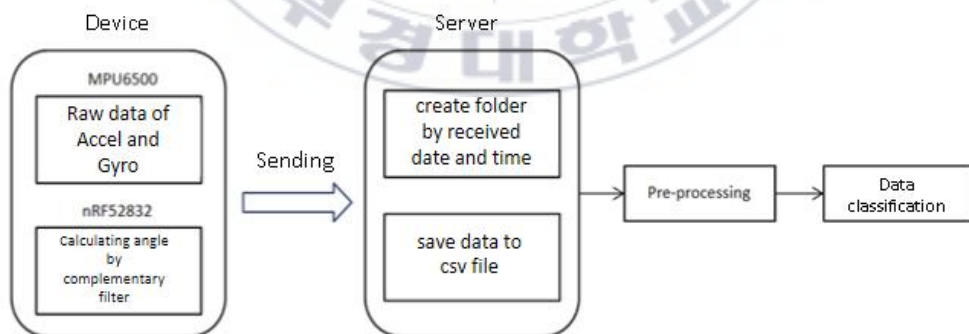


그림 3.1. Structure of the System.

이에 따라 연구 과정에서는 크게 데이터 측정과 무선통신, 데이터 가공의 3가지의 단계를 수행하게 되는데, 이를 위해 아래의 그림 3.2.와 같이

PCB를 설계 제작하였다.

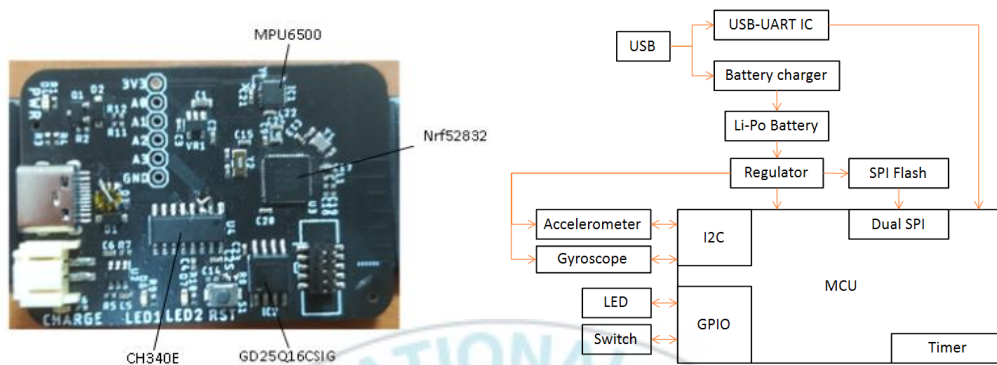


그림 3.2. Wearable sensor module PCB.

이 장치에는 3축 가속도와 3축 각속도를 측정할 수 있는 IMU 센서인 MPU6500(InvenSense Inc., U.S.A.)이 장착되어 있으며, 이를 통해 사용자의 동작에 따른 가속도와 각속도 데이터를 측정하였다. MPU6500은 6축 측정 센서이며 가속도의 측정범위를 $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ 중 하나로 정할 수 있고, 각속도의 측정 범위 역시 $\pm 250\text{deg/sec}$, $\pm 500\text{deg/sec}$, $\pm 1000\text{deg/sec}$, $\pm 2000\text{deg/sec}$ 중 하나를 선택할 수 있다. 본 연구에서는 가속도는 $\pm 16g$ 로, 각속도는 $\pm 2000\text{deg/sec}$ 로 설정하여 실험을 진행하였다. 그리고 이 장치에는 저전력으로 작동이 가능한 MCU이면서 Bluetooth Low Energy 통신을 지원하는 NRF52832를 사용하였다. 이 장치는 3.7V, 600mA에 해당하는 Lithium polymer battery를 연결하여 추가적인 전원이 없어도 작동할 수 있게 되어 있으며, 전원을 연결할 때 배터리를 충전할 수 있는 충전회로가 존재한다. 사용된 주요 부품들을 정리하면 표 3.1과 같다.

표 3.1. Parts specification

	Name	Specification
MCU	nRF52832	Arm Cortex-M4
Memory	GD25Q16CSIG	SPI Flash, 16MB
Accelerometer	MPU6500	$\pm 16g$
Gyroscope		$\pm 2000\text{deg/sec}$
RF	Bluetooth Low Energy	1 ~ 2000kbps
Battery	Li-Polymer	3.7V

3.1.2. 가속도센서와 자이로센서

관성(慣性, Inertia)은 물체가 운동 상태를 그대로 유지하려는 성질을 말하는데, 사람의 동작에서 관성을 측정하는 것이 관성측정장치(Inertial Measurement Unit: IMU)이다[31]. 이것은 가속도센서와 자이로센서로 구성되며, 때로는 지자기센서를 포함하기도 한다. 종류에 따라 가속도센서 3축과 자이로센서 3축의 6축 센서, 지자기센서까지 포함된 9축 센서가 있다. 가속도센서는 선형 가속도를 측정하고 자이로센서는 회전 운동을 감지하는 각속도를 측정하는데, 두 센서를 결합하여 3차원 공간에서의 물체의 움직임을 추적한다. 가속도는 물체가 움직일 때, 물체가 진행하는 속도의 변화를 뜻하고, 각속도는 기울어지는 각도의 속도를 의미한다.

가속도센서는 물체에 작용하는 가속력과 진동력, 충격력 등 동적 힘이 발생했을 때, 움직임의 변화에 따른 가속도의 변화를 순간적으로 감지해낸다. 일반적으로 물체는 속력이나 운동 방향이 바뀌면서 속도가 변하는데, 가속도(Acceleration)는 시간에 따라 속도가 변하는 정도를 나타내는 물리량을 뜻한다. 가속도는 a 로 표기되며, m/sec^2 가 단위가 된다. 가속도센서가 측정하는 가속도는 3축 방향으로 속도가 변화하는 데 걸리는 시간이다. 즉, 물체가 3축으로 진행하는 정도를 측정하는 것으로, 이렇게 구해진 가속도

를 통해 물체의 운동 상태를 상세하게 감지할 수 있다.

일반적으로 모든 물체는 지구의 중력을 적용 받고 있어서 외부 힘이 없는 경우 세 축의 크기 합은 중력가속도(g)로 측정된다. 가속도센서에서 받는 가속도는 x , y , z 각 축의 합성벡터로 표현할 수 있으며, Acc 로 자주 표현되는 가속도의 크기는 각 축의 요소를 제공하여 합한 후 제곱근을 구한 값이 된다. 이는 다음 식 (3.1)과 같이 표현할 수 있다.

$$Acc = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (3.1)$$

자이로센서는 운동량의 원리에 기초하여 물체의 방향을 측정할 수 있으며, 따라서 낙상 감지를 위해 가속도센서와 함께 사용된다. 자이로센서는 중력을 이용하여 코리올리 힘을 검출하는데, 중력이 가해질 때 진동 속도가 변하는 것을 각속도로 계산해 낸다.

각속도는 물체가 특정 축을 기준으로 회전하는 속도로, 시간당 회전하는 각도에 해당한다. 이것은 ω 로 표기하며, 단위는 rad/s 이고, 회전각도 θ 의 시간적 변화율 $\omega = d\theta/dt$ 로 나타낸다. 각속도는 어떤 물체가 회전 운동할 때 생기는 코리올리 힘을 전기적 신호로 변환하여 계산하는 방식으로 측정한다. 가속도를 적분하면 속도가 나오고, 속도를 적분하면 거리가 구해지므로 각속도 값을 시간에 대해 적분하면 축이 회전한 거리, 즉 각도를 구할 수 있다. x , y , z 의 축에 작용하는 힘을 이용하여 물체가 회전한 각도를 계산할 수 있다.

자이로센서의 측정값(A_x , A_y , A_z)을 이용해 $\text{roll}(\phi)$ 과 $\text{pitch}(\rho)$ 를 계산하면 다음과 같이 나타난다.

$$\Phi = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (3.2)$$

$$\rho = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (3.3)$$

그런데 정지 상태의 긴 시간의 관점에서 보면, 가속도센서에 의해 계산된 기울어진 각도는 올바른 값을 보여 주지만 자이로센서에서는 시간이 지날수록 틀린 값을 보여 준다. 반대로 움직이는 짧은 시간의 관점에서 보면, 자이로센서는 올바른 값을 보여 주지만 가속도센서는 기울어진 각도와는 다른 계산 값을 보여 주기도 한다. 자이로센서와 가속도센서의 값을 함께 보정함으로써 물체가 지면의 수평면과 어느 정도의 각도로 틀어져서 어느 정도의 속도로 이동하는지를 정확히 계산할 수 있다.

3.2. 데이터 수집 및 처리

3.2.1. 실험 환경

낙상 감지를 위한 동작을 수행하는 데에는 위험이 동반된다. 그리하여 실험 참가자들은 낙상 동작의 수행에서 상대적으로 불편을 덜 느끼도록 해주어야 하며, 또한 낙상 실험 시에는 충격을 흡수해 줄 장비가 필요하다. 본 연구에서는 이러한 조건에 맞추어 총 10명의 참가자를 선정하고, 반드시 매트리스 위에서 안전하게 동작을 수행하도록 하였다. 또한, 참가자들은 가속도 데이터의 크기 변화가 두드러지지 않고 특정 동작의 영향을 덜 받으며, 착용에 따른 불편함이 비교적 적다고 판단되는 복부에 측정 장치를 부착하도록 하였다.



그림 3.3. Equipment Attached Status and Location.

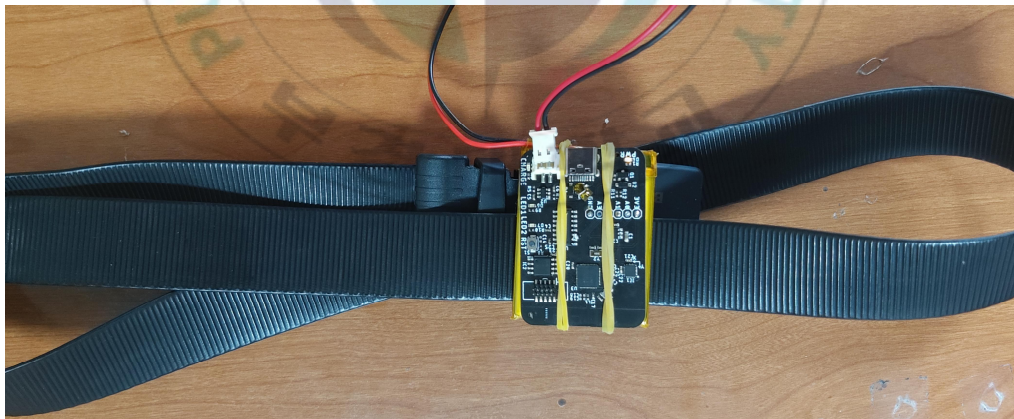


그림 3.4. Overall View of the Belt and Attached Device.

실생활에서 주로 행해지는 동작들은 연속적이어서 쉽게 구분되지 않는다. 본 연구에서는 낙상과 관련하여 일상적인 동작을 ADL(activity of daily life)과 낙상으로 구분하고, ADL은 서있기, 걷기, 달리기, 앉기, 일어나기, 눕기, 계단 오르기를 선정하였다. 그리고 낙상은 앞으로 넘어지기, 뒤로 넘어지기, 옆으로 넘어지기로 구분하였다. 그리고 이들은 각각 다음과 같이 정리되는 것으로, 실험 참가자들에게 설명하고 각각의 동작을 수행하게 하여 데이터를 수집하였다.

표 3.2. Action Classification

Task	Description	분류
서있기	60초 동안 움직이지 않고 서있기	ADL
걷기	평평한 바닥에서 30m 정도 걷기	
달리기	빠른 속도로 30m 정도 달리기	
앉기	서있는 상태에서 의자에 앉기 20회	
일어나기	의자에 앉아있는 상태에서 일어서는 동작 20회	
눕기	서있는 상태에서 매트 위로 눕는 동작 20회	
계단오르기	60초 동안 계단 오르기	
앞으로 넘어지기	서있는 상태에서 전방 낙상 10회	낙상
뒤로 넘어지기	서있는 상태에서 후방 낙상 10회	
옆으로 넘어지기	서있는 상태에서 측방 낙상 10회	

그리고 원활한 낙상 감지를 위해 3축 가속도의 값과 roll, pitch 값 이외에 가속도의 크기를 수신단계에서 함께 측정하였으며, 그 결과로 나타난 가속도의 크기와 roll, pitch 값을 기반으로 낙상 데이터를 데이터 윈도우로 분류하였다. 각 윈도우는 낙상과 낙상 전 후의 데이터가 함께 들어갈 수 있도록 2초간 200개의 데이터가 들어가도록 설정되었다.



그림 3.5. Falling movements.

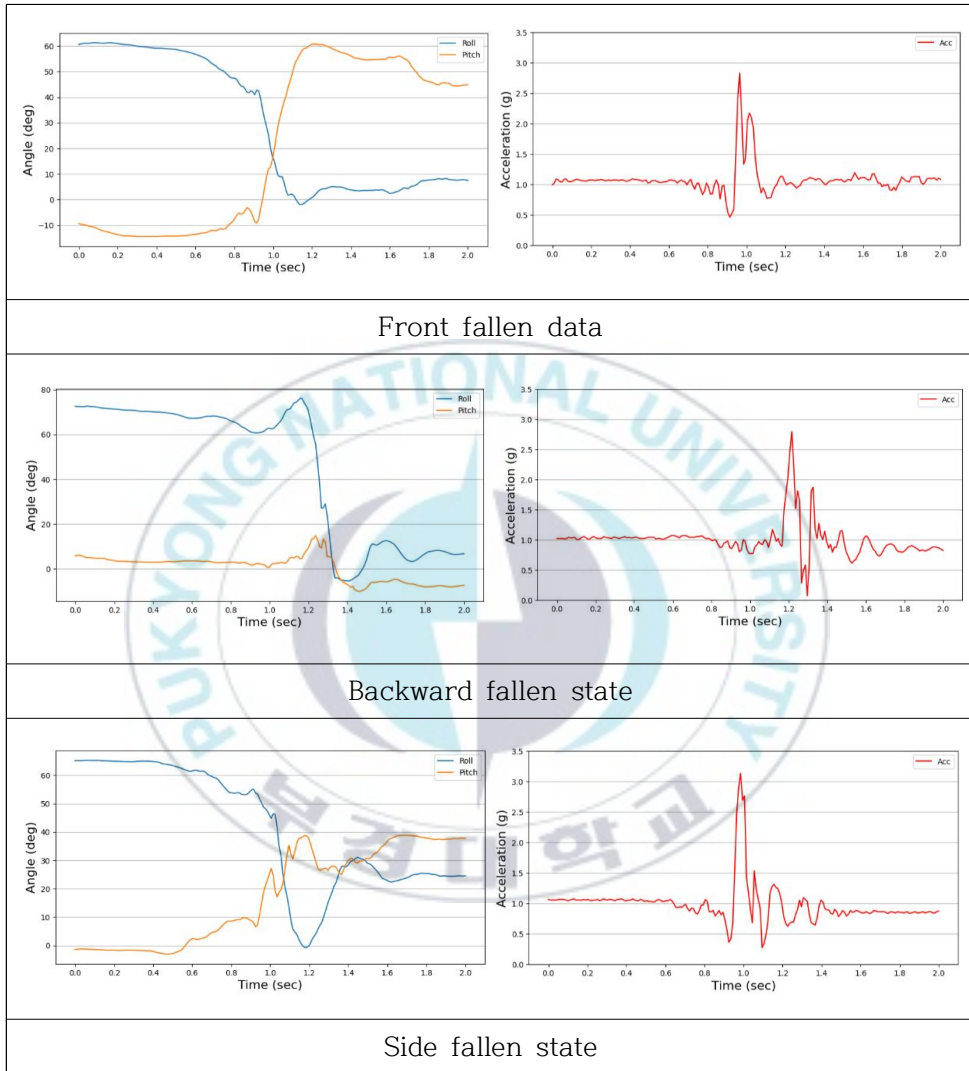


그림 3.6. Falling data.

낙상이 발생하면 사람의 몸이 기울어지면서 지면에 부딪히는 상황까지에 가장 큰 각도와 변화가 있으며, 지면에 몸이 닿은 이후에도 신체의 상태가 안정화될 때까지 시간이 걸리게 된다. 본 연구는 roll과 pitch 값과 가속도의 크기값을 비교하여 낙상의 시작과 끝을 정하고, 그에 기반해 라벨링을 진행하였다.

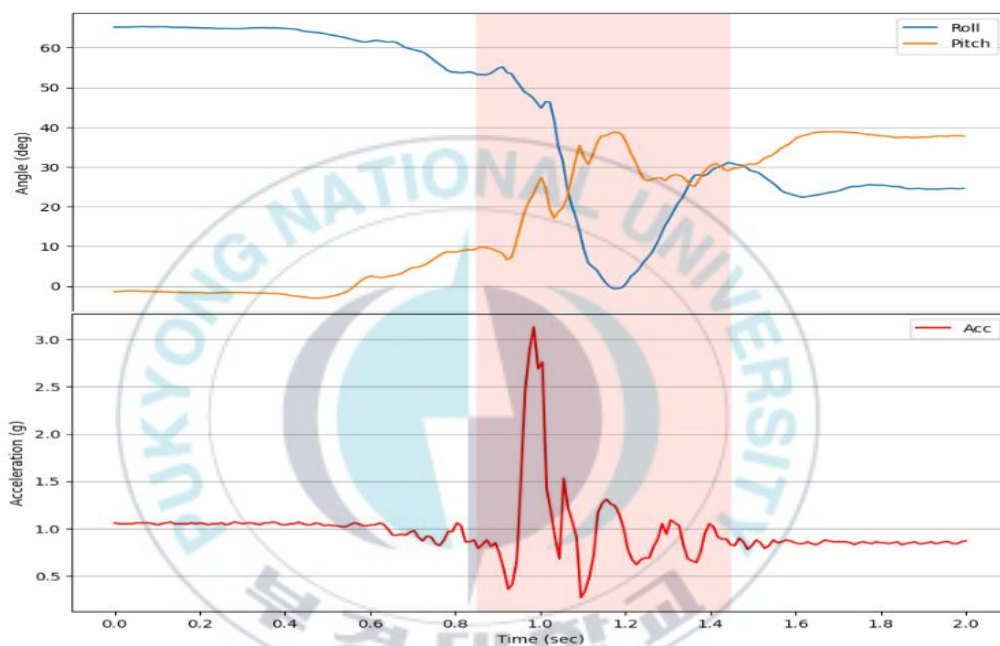


그림 3.7. Defining Fall Range by Comparing Changes in Angle and Magnitude of Acceleration.

3.2.2. 각도 측정

IMU 센서를 활용하여 각도를 구할 때 센서를 통합하고 노이즈를 줄이기 위해 필터링이 요구된다. 이를 위해 주로 사용되는 방법으론 상보필터와 칼만필터를 적용시키는 방법이 있는데, 이 중에서 상보필터가 상대적으로 더 빠르게 계산이 이루어진다. 본 연구에서는 상대적으로 더 빠르게 값을

출력할 수 있는 상보필터(Complementary filter)를 활용하였다.

가속도센서는 진동 등의 고주파에 취약하며, 자이로센서는 드리프트와 같은 저주파에 영향을 받게 된다. 상보필터는 이런 가속도센서와 자이로센서에서 나타나는 단점들을 서로 보완해 주는 역할을 한다. 이 상보필터를 수식으로 나타내면 식 (3.4)와 같다.

$$Angle = a(pre_Angle + Angle_{gyr}) + (1 - a)Angle_{acc} \quad (3.4)$$

a 는 가중치로, 일반적으로 자이로센서값을 더 신뢰해서 1에 가까운 값을 가지게 된다. 본 실험에서는 가중치를 0.95로 설정하였으며, 따라서 최종적으로 측정하고자 하는 각도인 roll과 pitch 값은 식 (3.5)와 식 (3.6)으로 나타나게 된다.

$$roll = 0.95(gx*dt + Angle_{gyr}) + 0.05*atan2(-ax, az)*180/\pi \quad (3.5)$$

$$pitch = 0.95(gy*dt + Angle_{gyr}) + 0.05*atan2(-ax, \sqrt{(ay)^2 + (az)^2})*180/\pi \quad (3.6)$$

이 계산은 측정 장치의 MCU 내에서 수행되며 가속도 데이터값과 함께 서버로 전송된다.

3.2.3. 데이터 증강

낙상은 그 위험성으로 인해 데이터 수집에 어느 정도 한계가 있다. 그런데 데이터가 부족하면 과적합(Overfitting)과 같은 문제가 생기면서 학습이 제대로 이루어지지 않을 수 있다. 이런 문제를 해결하기 위해 데이터 증강을 통해 개선할 필요가 있다.

데이터 증강은 기존에 존재하는 데이터를 변형해서 새로운 데이터를 만드는 방식을 뜻한다. 본 연구에서는 상대적으로 측정이 원활한 ADL 데이

터에 비해서 데이터의 수가 부족한 낙상 데이터에 적용을 하여 ADL과 낙상 데이터 간의 비율 차이를 완화하였다. 데이터 증강 방식으로는 시계열 데이터에 적용이 용이한 가우시안 노이즈(Gaussian Noise) 추가와 시간 왜곡(Time Warping)을 적용하였다.

다만 가속도의 크기를 나타내는 Acc 값은 3축 가속도 값들을 통해서 얻어내는 값이기 때문에 노이즈 추가를 직접적으로 하지 않고 증강을 적용한 후 다시 수식을 통해서 Acc 값을 도출하도록 설정하였다. 다음 그림 3.8.은 기존의 데이터와 증강을 통해 변형된 데이터의 예시를 보여 준다.





그림 3.8. Comparison of Augmented Data.

그림 3.8.은 낙상에 해당하는 특정 데이터의 데이터 윈도우에서 가속도 센서로 측정된 값과 그 값에 증강을 적용하여 변형된 상태를 보여 준다. 두 번째 그래프는 노이즈를 추가해서 데이터가 변형된 모습을 보여 준다. 세 번째 그래프는 시간 축을 늘리거나 줄여서 데이터를 변형하는 기법인 시간 왜곡 증강 기법이 적용되어 전반적인 패턴은 원 데이터와 동일하게 유지되지만 형태가 변형된 모습을 보여 준다.



IV. 결과 분석

4.1. 모델 설계와 평가 지표

4.1.1. 예측 모델 설계

낙상 감지 알고리즘은 주어진 데이터에 대해 패턴을 인식하고 자율적으로 학습하여 최종적인 예측을 수행하는 모델이 설정된다. 즉, 데이터(data) → 모델(model) → 실행(action)의 과정을 수반하여 가정을 조정하고 오류에 대응하면서 예측 모델이 설정되는 것이다. 이 과정은 특정 환경에서 사용 가능한 모든 정보를 고려할 수 있고 관찰 데이터가 부족하거나 예측 변수가 많은 경우에도 적용될 수 있다.

선행 연구에서는 데이터 수집과 처리를 통해 데이터 세트를 구축하고 학습 모델을 구성하였으며, 이에 따른 결과를 분석하여 데이터와 모델을 수정하는 과정을 거쳐 학습 모델을 향상시키고자 하였다. 본 연구에서는 데이터 수집과 처리를 통해 구성된 데이터 세트를 활용하여 다음과 같은 과정으로 낙상 감지 모델을 생성하고 그 효율성을 평가하였다.

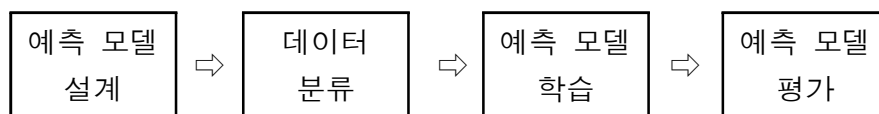


그림 4.1. Creation and Evaluation of the Model.

우선 예측 모델을 설계하고, 앞에서 수집된 데이터를 학습 데이터와 평가 데이터로 분류한 다음, 학습 데이터로 모델을 학습시키고, 평가 데이터로 모델을 검증하여 평가하였다.

본 연구에서 예측 모델은 앞의 2장에 언급한 바와 같이 대표적인 기계학습으로서의 SVM과 딥러닝 모델로서의 1D CNN, LSTM, CNN-LSTM, TCN 등을 바탕으로 설계하였다. SVM은 Python용 기계학습 라이브러리인 scikit-learn을 활용하고, 파라미터는 scikit-learn에서 제공하는 grid search를 통해 최적의 것으로 설계하였다[32]. 그리고 딥러닝 모델에서 LSTM과 CNN, TCN은 google에서 제공하는 기계학습 라이브러리인 TensorFlow를 사용하였으며, 모델을 최적화하고자 하였다[33].

4.1.2. 평가 지표

모델의 성능을 평가하는 것은 모델이 주어진 작업을 얼마나 잘 수행하는지를 확인하는 과정이다. 이를 위해서 여러 지표들을 사용할 수 있는데, 본 연구에서는 scikit-learn에서 제공하는 분류 예측 리포트인 Classification_report에 의거하여 Accuracy(정확도), Precision(정밀도), Recall(재현율)을 계측하였다. 이것을 Confusion matrix(오차 행렬)로 보면 다음 표 4.2와 같다[34].

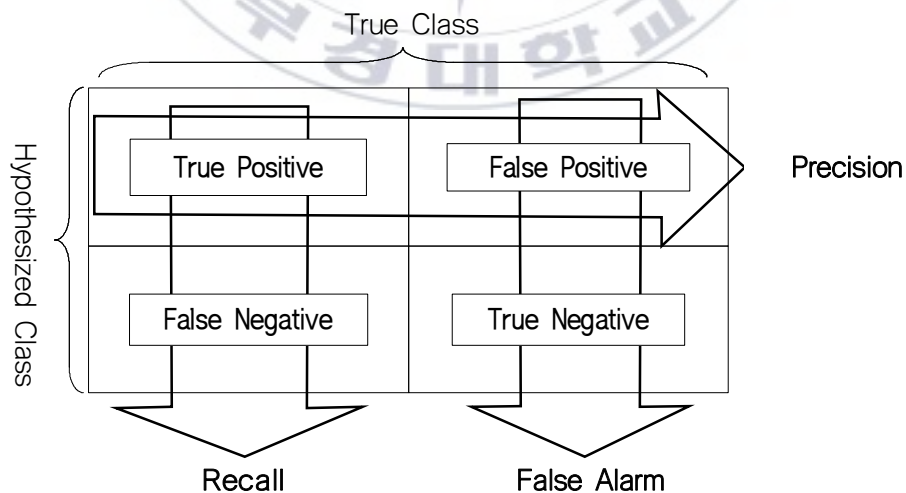


그림 4.2. Confusion matrix.

Accuracy(정확도)는 측정된 값을 표준이거나 이미 알려진 값에 어울리게 측정된 값으로 이해되는 것을 의미하며, 전체 데이터 중에서 차지하는 정확한 예측의 총수로 정의할 수 있다. 이것은 기계학습 모델의 성능을 결정하는 데 중요한 기준이 된다. 하지만 정확도는 모든 모델의 성능을 모두 반영하는 것이 아니며, 특히 클래스에 속한 샘플이 그 수에서 불균형이 클 경우에는 적절한 기준이 되지 못한다.

Precision(정밀도)와 Recall(재현율)은 이진 분류(binary classification)에서 모델이 양성 클래스를 얼마나 정확하게 예측하는지에 관한 지표이다. Precision은 긍정적인 것으로 예측된 것 중에서 실제로 긍정적인 것의 비율인데, 잘못된 알람(False Alarm)의 여부를 알려 준다. 그리고 Recall은 긍정적 표본을 탐지하는 모델의 능력을 측정하며, 이것이 높을수록 샘플은 더 긍정적으로 평가된다.

그리고 이들은 다음과 같은 수식으로 정의된다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

Precision과 Recall은 상충 관계에 있기 때문에 어느 한쪽의 점수가 높으면 다른 쪽은 점수가 낮은 경향을 보여 준다. 이에 대해 정밀도와 재현율을 함께 참고할 수 있는 F1-score가 활용된다.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

F1-score는 Precision과 Recall의 조화 평균값으로, 이 두 가지를 같이 고려하는 지표이다. 이것은 특별히 Precision과 Recall 중 어느 것이 높아야 한다는 목적이 없는 경우 많이 사용되며, 값이 클수록 성능이 좋은 것으로 평가된다.

4.2. 모델 평가

4.2.1. 환경 구성 및 하이퍼 파라미터 설정

본 연구에서는 딥러닝 모델의 학습과 테스트는 클라우드 기반의 딥러닝 개발 및 실행 플랫폼인 google의 Colab을 사용하였다. Colab은 Python을 기반으로 하며 V100 GPU가 사용 가능한 환경이다. 또한 인공지능망 모델은 google에서 제공하는 오픈소스 학습 라이브러리인 TensorFlow를 통해 진행하였다.

표 4.1. Development Environment

Tool	Specification
플랫폼	google Colaboratory
언어	Python 3
모듈	TensorFlow
GPU	V100

딥러닝 모델을 학습하고 테스트하기 위해서는 정형화된 데이터 세트를 구성해야 한다. 그리하여 가속도 3축과 roll, pitch로 구성된 총 5개의 축과 200 프레임을 가지는 200x5 형태의 데이터 세트를 구성하였다. 그림 4.3.은

본 연구에서 사용된 데이터 세트의 구조를 나타낸다.

표 4.2. Setting Common Parameters

하이퍼 파라미터 이름	공통 파라미터 설정
정규화	MinMaxScaler
최적화 함수	adam
손실함수	crossentropy
드롭아웃	30%
활성함수	sigmoid
시퀀스 길이	200 프레임

딥러닝 모델들의 경우, 다양한 하이퍼 파라미터를 통해 다양한 설정을 할 수 있다. 그래서 모델들 간에 공통된 파라미터 설정을 부여해서 좀 더 원활한 비교를 진행하였다. 공통 파라미터로는 특성들 사이의 스케일 차이를 줄이는 데 도움이 되는 MinMaxScaler 함수를 적용하였고, 기계학습 모델을 학습시킬 때는 도움이 되는 경사 하강법을 기반으로 한 최적화 알고리즘인 adam을 적용하였다. 그리고 낙상과 비낙상의 구분과 같은 분류 문제에 사용 빈도가 높은 crossentropy 손실 함수와 sigmoid 함수를 적용하였다.

본 실험의 경우 각 딥러닝 모델들의 학습률을 0.001, 0.0005, 0.0001의 3가지로 설정하여 학습률 별로 어떤 결과가 나타나는지 각각의 결과값을 측정하였다. 또한 SVM의 경우, 다른 모델과 달리 딥러닝 모델이 아닌 머신러닝 모델로 분류되어야 하며, 이 모델들의 경우 위에서 설정한 학습률과 같은 파라미터 설정을 사용하기 힘들다. 따라서 SVM 모델은 임의로 조정이 가능한 양의 매개 변수인 C를 1.0, 10.0, 100.0의 3가지로 설정하여 측정하였다.

4.2.2. 모델별 설정과 성능 평가

우선 SVM의 경우, 딥러닝 모델들에 사용이 되는 하이퍼 파라미터 설정이 아닌 scikit-learn을 통해 SVM을 구현할 때 몇 가지 인자들의 설정을 할 수 있다. 이에 따라 매개 변수인 C를 1.0, 10.0, 100.0로 설정하고 데이터가 고차원 공간에서 복잡하게 분리될 수 있어도 잘 작동되는 Radial Basis Function (RBF) 커널을 적용하였다. SVM의 평가 결과를 지표별로 나타내면 다음 표 4.3과 같다.

표 4.3. SVM result according to C

C	Accuracy	Precision	Recall	F1-score
1.0	0.917	0.819	0.473	0.600
10.0	0.913	0.750	0.509	0.606
100.0	0.910	0.716	0.524	0.605

C의 값이 1.0일 때가 Accuracy 값과 Precision 값이 가장 높게 나타났지만 Recall 가장 낮게 나타났고 이 값은 C가 100.0일 때 가장 높게 나타났다. 하지만 F1-score의 경우 C가 10.0일 때가 미세하게 앞서는 결과를 보여 준다.

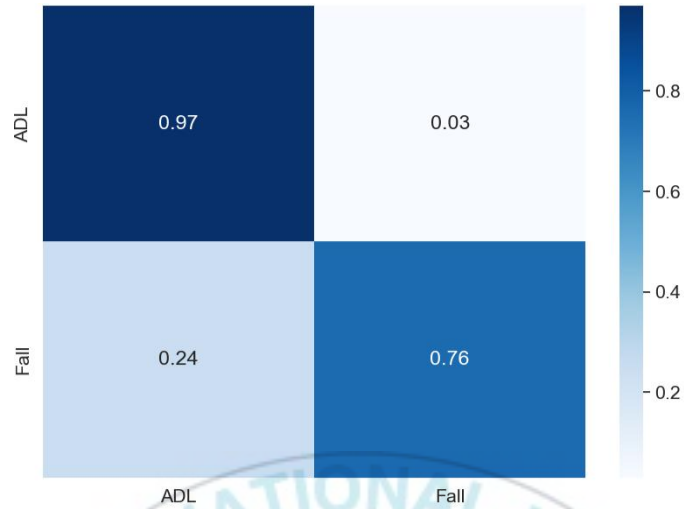


그림 4.4. Confusion matrix of SVM.

1D CNN의 경우, 공통 하이퍼 파라미터로 설정한 adam, sigmoid, crossentropy, 함수를 적용하였고 dropout은 30%로 설정하였다. 또한 정규화 방식으로 MinMaxScaler에 추가적으로 L2 가중치 감쇠(Regularization)와 배치 (Batch) 정규화를 적용하였다. 1D CNN 모델은 총 3개의 히든 레이어를 가지게 되었으며 각 레이어의 필터의 숫자는 64개로 동일하게 부여하였다. 이것은 다음 그림 4.5와 같이 보일 수 있다.

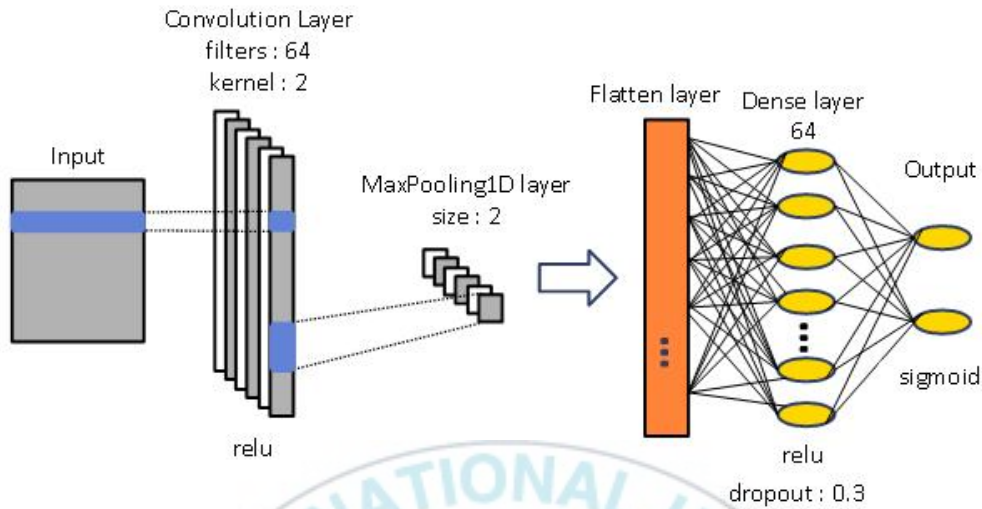


그림 4.5. Structure of 1D CNN model.

1D CNN은 학습률을 감소시킴에 따라 전반적으로 결과값들이 향상되는 현상을 보였다. 그런데 Recall 값은 Precision-Recall 사이의 트레이드 오프의 영향으로 0.0005에 비해 Precision이 급격하게 상승하며 오히려 떨어지는 현상을 보였다. 이와 같은 1D CNN의 지표별 결과는 다음과 같이 나타났다.

표 4.4. CNN result according to learning rate

Learning Rate	Accuracy	Precision	Recall	F1-score
0.001	0.933	0.804	0.652	0.720
0.0005	0.940	0.792	0.731	0.760
0.0001	0.947	0.864	0.701	0.774

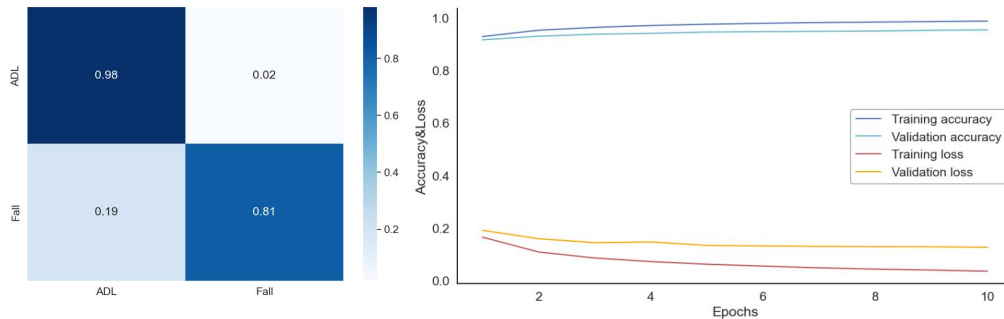


그림 4.6. Result of 1D CNN.

LSTM의 경우, 공통 하이퍼 파라미터로 설정한 adam, sigmoid, crossentropy, 함수를 적용하였고 dropout은 30%로 설정하였다. 또한 정규화 방식으로 1D CNN과 마찬가지로 MinMaxScaler에 추가적으로 L2 가중치 감소 (regularization)와 배치(Batch) 정규화를 적용하였다. LSTM은 위 구성에 따라 총 4개의 히든 레이어를 가지게 되었으며 입력 레이어에는 60개의 노드를 적용하였고, 히든 레이어에는 각 30개의 노드를 적용하였다. 구현된 모델의 구성은 다음 그림 4.7과 같다.

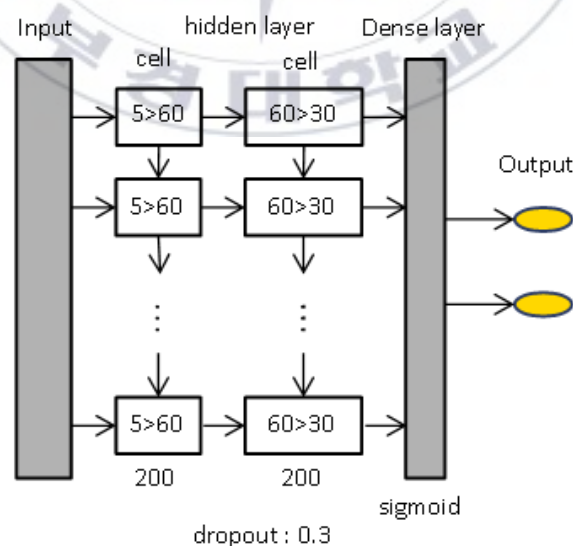


그림 4.7. Structure of LSTM model.

LSTM 역시 학습률이 낮을 때 더 높은 결과값을 도출하였다. 또한 학습률이 낮아짐에 따라 Recall 값이 Precision 값과의 차이가 줄어들다 0.0001 에선 오히려 앞서는 결과를 보여주었다. 이러한 LSTM의 지표별 평가 결과를 보면 다음과 같다.

표 4.5. LSTM result according to learning rate

Learning Rate	Accuracy	Precision	Recall	F1-score
0.001	0.942	0.810	0.721	0.763
0.0005	0.935	0.763	0.732	0.747
0.0001	0.952	0.812	0.823	0.817

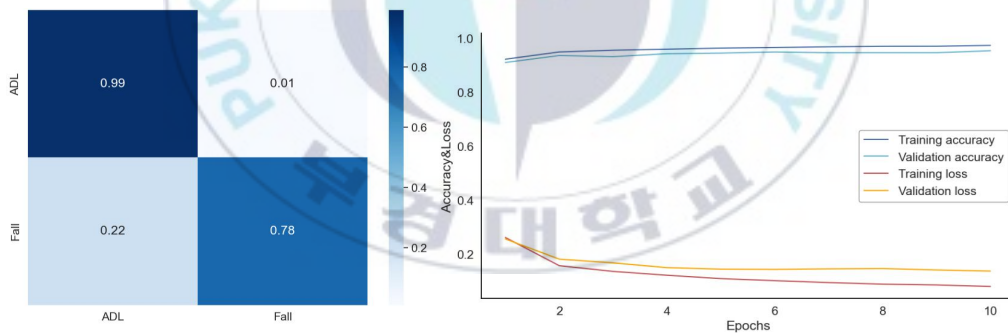


그림 4.8. Result of LSTM.

CNN-LSTM의 경우, 앞서 다룬 1D CNN과 LSTM을 결합한 것이었다. 공통 하이퍼 파라미터로 설정한 adam, sigmoid, crossentropy, 함수와 MinMaxScaler를 정규화 작업이 적용되었고 dropout은 30%로 설정하였다. Kernel이 2인 64개의 필터를 가진 레이어를 통과한 후 60개의 노드와 30개의 노드가 있는 레이어들을 통과시켰다. 구현된 모델의 구성은 다음 그림

4.9와 같다.

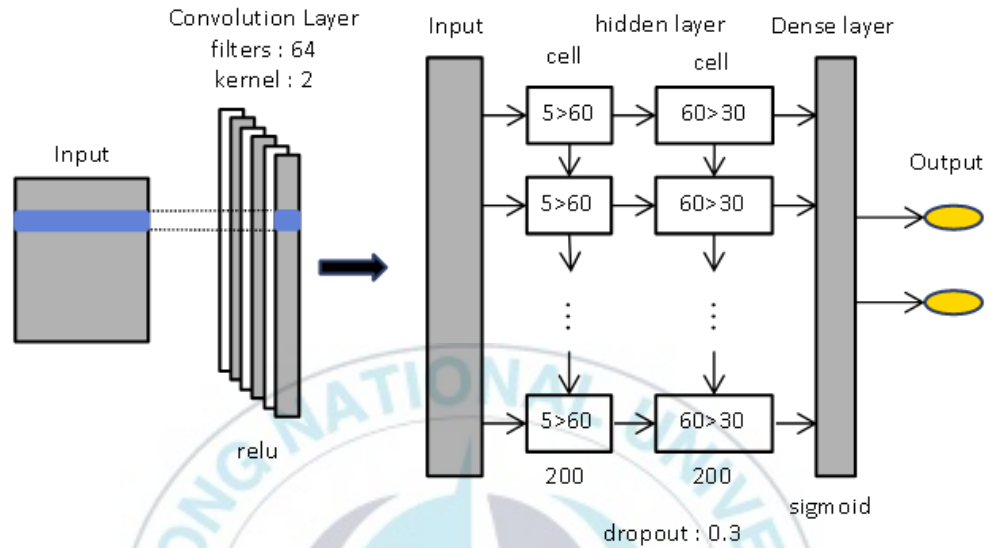


그림 4.9. Structure of CNN - LSTM model.

CNN-LSTM 모델은 0.0005의 학습률에서 가장 좋은 성능을 기록하였다. Precision은 0.0001의 학습률에서 0.9를 넘어섰지만 Recall이 심하게 떨어지는 결과를 보였다. CNN-LSTM 모델의 지표별 평가 결과는 다음과 같다.

표 4.6. CNN-LSTM result according to learning rate

Learning Rate	Accuracy	Precision	Recall	F1-score
0.001	0.940	0.837	0.670	0.745
0.0005	0.951	0.886	0.720	0.794
0.0001	0.944	0.910	0.634	0.747

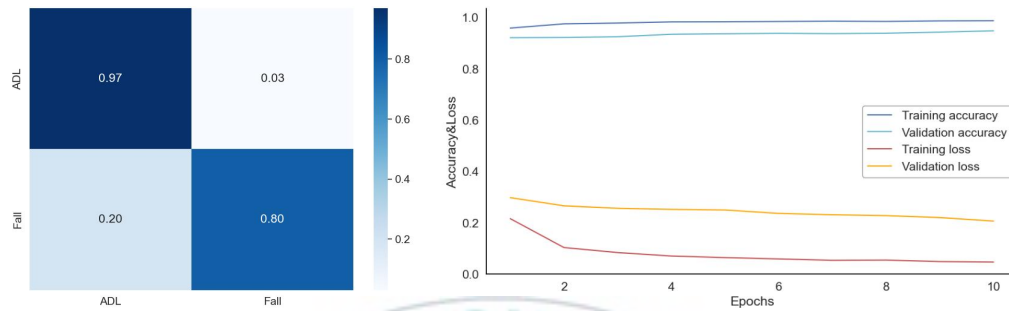


그림 4.10. Result of 1D CNN - LSTM.

TCN의 경우, 공통 하이퍼 파라미터로 설정한 adam, sigmoid, crossentropy 함수와 MinMaxScaler를 정규화 작업으로 적용하였고 dropout은 30%로 설정하였다. TCN 모델은 총 2개의 레이어 스택이 존재하는데, 각 레이어는 필터의 숫자는 64개로 정하였고, 커널의 크기는 2로 설정하였다. 그리고 각 레이어는 dilation factor를 4, 8로 설정하였다. 구현된 모델의 구성은 다음 그림 4.11과 같다.

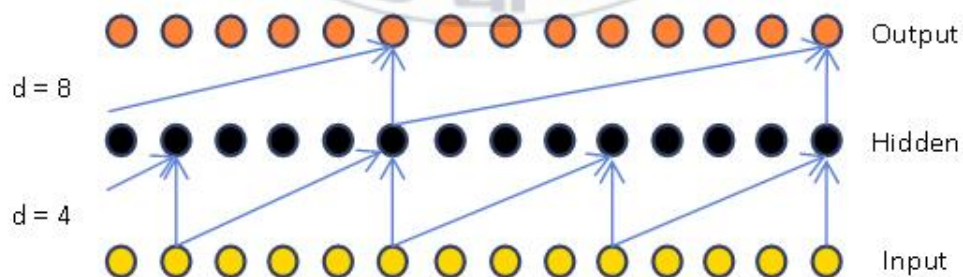


그림 4.11. Temporal Convolutional Network.

TCN은 학습률의 변화에 따라 결과값이 일정하게 변하는 모습이 전혀 보이지 않았다. 정밀도와 재현율을 함께 조합시킨 F1-score의 경우 0.001의 학습률이 가장 좋은 결과값을 보였다. TCN의 지표별 평가 결과는 다음과 같다.

표 4.7. TCN result according to learning rate

Learning Rate	Accuracy	Precision	Recall	F1-score
0.001	0.943	0.802	0.743	0.771
0.0005	0.932	0.712	0.805	0.755
0.0001	0.943	0.854	0.679	0.756

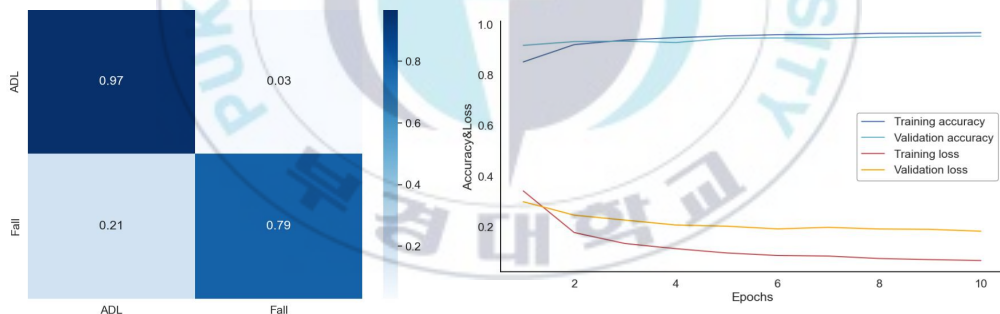


그림 4.12. Result of TCN.

위에서와 같은 평가를 바탕으로 각 모델의 성능을 평가할 수 있다. 그런데 낙상을 확실하게 감지하기 위해서는 모델이 낙상을 낙상으로 판단하는 Recall 값이 가장 큰 영향을 미치게 된다. 따라서 모델의 성능을 Recall 값을 기준으로 비교해 볼 수 있는데, 각 모델에서 Recall 값이 가장 높게 나타난 학습률 조건에서의 평가 지표별 측정값을 정리하면 다음 표 4.8과 같

다.

표 4.8. Performance evaluation score using Confusion matrix based on Recall

	SVM	1D CNN	LSTM	CNN-LSTM	TCN
Accuracy	0.910	0.940	0.952	0.951	0.932
Precision	0.716	0.792	0.812	0.886	0.712
Recall	0.524	0.731	0.823	0.720	0.805
F1-score	0.605	0.760	0.817	0.794	0.755

위 표 4.8에서 보면, Recall이 가장 높게 측정된 모델은 LSTM에 해당한다. Recall이 낙상을 낙상으로 감지하는 성능을 나타내는 척도이므로, 이 결과는 LSTM이 가장 낙상을 잘 감지한다는 것을 의미한다. 한편, 다른 평가 척도에서 Accuracy의 경우 마찬가지로 LSTM이 가장 높게 나타났고 CNN-LSTM이 LSTM과 비슷하게 측정되었다. 그리고 Precision은 CNN-LSTM이 가장 높은 측정값을 기록하였다.

그런데 Recall만을 기준으로 판단을 하게 되면 Precision과 Recall 사이의 trade-off로 인하여 Precision을 낮추는 경향이 나타난다. 높은 Precision은 True 케이스들을 식별하려는 경향이 있지만, 이는 다르게 말하면 False 케이스를 True로 판단할 수도 있다는 의미가 된다. 따라서 Recall과 Precision을 조합한 F1-score를 기준으로도 성능 평가를 진행하였는데, 표 4.9는 각 모델에서 F1-score가 가장 높게 나타난 학습률 조건에서의 평가 지표별 측정값을 정리한 것이다.

표 4.9. Performance evaluation score using Confusion matrix based on F1-score

	SVM	1D CNN	LSTM	CNN -LSTM	TCN
Accuracy	0.913	0.947	0.952	0.951	0.943
Precision	0.750	0.864	0.812	0.886	0.802
Recall	0.509	0.701	0.823	0.720	0.743
F1-score	0.606	0.774	0.817	0.794	0.771

위 표 4.9에서 보면, F1-score를 기준으로 평가하였을 때도 가장 높은 점수를 기록한 모델은 LSTM으로 나타났다. Recall과 Accuracy는 순위에 큰 차이가 없었지만, Precision의 경우 Recall을 기준으로 측정했을 상황과 동일하게 CNN-LSTM이 가장 높은 결과값을 기록하였고, 그 밖의 모델들에서는 순위의 변동이 있었다.

결국 Confusion matrix를 활용하여 Recall과 F1-score를 기준으로 비교를 한 결과, LSTM이 다른 모델들에 비해 가장 높은 성능을 보이는 것으로 확인되었다. CNN 모델과 LSTM 모델을 조합한 CNN-LSTM 모델이 LSTM과 가장 비슷한 결과를 보여 주었고, 1D-CNN과 TCN이 서로 비슷한 결과를 보여 주었다. 이러한 결과는 선행 연구와 유사하였는데[13, 25-33], SVM은 전반적으로 다른 모델들에 비해서 낮은 특성을 보였다.

V. 결론

낙상에 의한 재해는 건설업이나 제조업 등의 산업 현장뿐만 아니라 화재 진압과 같은 고위험 활동에서 큰 비율을 차지한다. 본 연구에서는 이러한 고위험군 직종 종사자들의 낙상에 의한 사고에 대응하기 위해 현장 적용이 가능한 IMU 센서 기반의 낙상 감지 시스템을 구현하고 사용자의 움직임에 따른 데이터를 수집하였으며, 이를 바탕으로 낙상 감지 예측 모델의 성능을 평가하고 비교하였다.

낙상 감지 시스템은 3축 가속도센서와 자이로센서를 활용하였으며, 실험 참가자에게 이것을 복부에 착용하고 일상적 동작과 낙상의 동작을 하게 하고, 그들의 신체 활동을 모니터링하여 운동량 정보로 변환하고 데이터를 수집하였다. 수집된 데이터는 상보필터를 활용하고 증강하는 등의 처리 과정을 거쳤다. 그리고 처리된 데이터를 바탕으로 대표적인 기계학습 모델인 SVM, 그리고 딥러닝 모델인 1D CNN, LSTM, CNN-LSTM, TCN을 생성하고 유효성을 검증하였다.

본 연구에서는 Accuracy(정확도), Precision(정밀도), Recall(재현율), 그리고 F1-score를 계측하여 각 모델의 유효도를 비교하였다. 그 결과를 평가 지표별로 보면, Accuracy에서는 LSTM이 가장 높은 점수를 기록하였고, Precision에서는 CNN-LSTM이 가장 높았으며, Recall에서는 LSTM이 제일 좋게 나타났다. 그리고 F1-score에서는 Recall이 가장 좋았고 Precision 역시 준수한 LSTM이 가장 우수한 것으로 나타났다. 전체적으로 보면, LSTM 가장 높게 나타났으며, 다음으로 CNN-LSTM 이 좋은 모델로 평가되었다. 1D CNN과 LSTM의 유효성이 높게 나타나는 것은 선행 연구와 유사하였다.

그런데 각각의 모델은 어느 부분에서는 높은 성능을 보이지만 또 다른 부분에서는 한계가 나타난다. 낙상 감지 예측 모델이 효율적으로 작동하도록 하기 위해서는 보다 더 다양한 데이터를 확보하고 이것이 적절히 작동

되도록 하는 새로운 접근 방법이 모색되어야 할 것이다. 이를 위해서는 낙상과 비낙상 데이터의 양을 확대하고 그 비율을 적절히 조절되어야 하며, 모델을 다양하게 구성하여 적용할 필요가 있다. 또한 모델의 정확성을 높이고 최적화하기 위한 명확한 기준 설정에 대한 연구가 필요할 것이다. 특히 낙상을 유효하게 판별해 내기 위해서는 광범위한 인체의 행동 패턴의 연구와 함께 다양한 센서의 활용에 대한 연구가 병행되어야 할 것이며, 그에 따라 낙상 감지 시스템의 상용화를 통해 위험 상황이 발생했을 때 신속히 인지하여 대응하는 데 기여할 수 있게 될 것이다.



참고문헌

- [1] 고용노동부 정책자료실, ‘2022년 산업재해현황분석’,
<https://www.moel.go.kr/policy>
- [2] 소방청, ‘2022 소방청 통계연보’, <https://www.nfa.go.kr>
- [3] ‘United States Firefighter Injuries in 2021’,
<https://www.nfpa.org/-/media/Files/News-and-Research/Fire-statistics-and-reports/Emergency-responders/osffinjuries.pdf>
- [4] 질병관리청 국가건강정보포털 <https://health.kdca.go.kr>
- [5] 김재정, 박민호, 김신호, 서지윤, 정도운, “산업현장에 적용 가능한 IMU 센서 기반의 사용자 낙상감지 시스템 구현”. 한국정보통신학회 여성ICT 학술대회논문집, pp. 33-36. 2021.
- [6] 백운성, 김동민, 변재영, “낙상 감지시스템에 대한 최신 개발 동향 분석”, 정보기술융합공학논문지 2-1, pp. 30-40, 2012.
- [7] 정필성, 조양현, “사물인터넷 기반의 낙상 감지 시스템”, 한국정보통신학회논문지 19-11, pp. 2546-2553, 2015.
- [8] 김선기, 안종수, 김원호, “영상처리 기반 낙상 감지 알고리즘의 구현”, 한국위성정보통신학회논문지 12-2, pp. 56-60, 2017.
- [9] X. Luo, T. Liu, J. Liu, X. Guo and G. Wang, “Design and implementation of a distributed fall detection system based on wireless sensor networks”, EURASIP Journal on Wireless Communications and Networking, PP. 118, 2012.
- [10] 김동완, 유중현, 백승화, “바닥 진동을 통한 노인 낙상 검출”, 전기전자학회논문지 18-1, pp. 134-139, 2014.
- [11] M. Popescu, Y. Li, M. Skubic and M. Rantz, “An Acoustic Fall Detector System that Uses Sound Height Information to Reduce the False Alarm Rate”, 30th Annual International IEEE EMBS Conference

- Vancouver, British Columbia, Canada, pp. 20-24, 2008.
- [12] Lin, C. W., Z. H. Ling, Y. C. Chang, C. J. Kuo, "Compressed-domain Fall Incident Detection for Intelligent Homecare", The Journal of VLSI Signal Processing, Vol.49, No.3, pp. 393-408, 2007.
- [13] 김인경, 김대희, 노송, 이재구, "1D CNN과 기계 학습을 사용한 낙상 검출", 정보처리학회논문지 소프트웨어 및 데이터 공학 10-3, pp. 85-90, 2021.
- [14] 양진숙, 김주연, "뉴미디어 시대의 웨어러블 디바이스 사례분석 연구", 한국디자인문화학회지 20-2, pp. 354-364, 2014.
- [15] Zhang, T., J. Wang, L. Xu, P. Liu, "Fall Detection by Wearable Sensor and One-Class SVM Algorithm", Lecture Notes in Control and Information Sciences, pp. 858-863, 2006.
- [16] 정도운, 정완영, "3축 가속도 센서를 이용한 자세 및 활동 모니터링", 센서학회지, 16-6, pp. 467-474, 2007.
- [17] Zhang, T., J. Wang, P. Liu, J. Hou, "Fall Detection by Embedding in Accelerometer in Cellphone and Using KDF Algorithm", IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.10, pp. 277-284, 2006.
- [18] Lindermann, U., A. Hock, M. Stuber, W. Keck, C. Beeker, "Evaluation of a fall detector based on accelerometers: A pilot study", Medical Biological Engineering and Computing, Vol.43 No.5, pp. 548-551, 2005.
- [19] Bourke, A. K., G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor", Medical Engineering and Physics, Vol.30, pp. 84-90, 2008.
- [20] Arthur Samuel, Some Studies in Machine Learning Using the Game of Checkers, IBM Journal of Research and Development, Vol. 3, Issue 3, pp. 206-227, 1959.
- [21] 김남호, "딥러닝 기반 낙상 인식 알고리즘", 한국정보통신학회 종합학술대회논문집 25-1, pp. 552-554, 2021.

- [22] 김인경, 김대희, 허성실, 이재구. "기계학습 기반의 낙상 검출". 한국정보처리학회 학술대회논문집, 27-1, pp. 547-550, 2020.
- [23] 김태민, 김동환, 천세학. "머신러닝기법들의 낙상사고 예측을 비교 연구." 한국지능정보시스템학회 학술대회논문집, pp. 91-92, 2018.
- [24] 김예나. "낙상 위험도 예측을 위한 머신러닝 기반 낙상 요인 분석", 순천향대학교 석사학위논문, 2023.
- [25] 이영욱, 박재한, 신수용, "CNN-LSTM 기반 낙상 감시 시스템 구현", The Journal of Korean Institute of Communications and Information Sciences 47-2. pp. 340-347, 2022.
- [26] 김인경, 허성실, 권선영, 이재구. "전이 학습을 통한 GRU와 LSTM의 낙상 검출 성능 향상", 2020년도 한국통신학회 학술대회논문집, pp. 597-598, 2020.
- [27] 박호연, 김경재, "CNN-LSTM 조합모델을 이용한 영화리뷰 감성분석", 지능정보연구 25-4, pp. 141 - 154, 2019.
- [28] 류동흠, 이용빈, 최동훈, "시간 합성곱 신경망을 이용한 베어링 성능 저하 예측 방법 개발", 대한기계학회 춘추학술대회, pp. 1138-1141, 2021.
- [29] 박경운, "SVM, LSTM, CNN-LSTM, TCN을 이용한 금 가격 예측", 이화여자대학교 석사학위논문, 2022.
- [30] 조민우, "LSTM 칩 GRU를 활용한 홍수 관리 시스템 설계 및 구현", 배재대학교 석사학위논문.
- [31] J.-J. Kim, et al., "Implementation of the fall detection system based on IMU sensor applicable to industrial sites", J. KIICE, pp. 33-36, 2021.
- [32] <https://scikit-learn.org>
- [33] <https://www.tensorflow.org>
- [34] 오상훈, "개정판 기계학습 -수학적 이해에서 알고리즘까지-", pp. 133-135, 도서출판 홍릉, 2020.