

<http://dx.doi.org/10.17703/JCCT.2022.8.6.741>

JCCT 2022-11-91

## 음성신호를 이용한 기계학습 기반 피로도 분류 모델

# Fatigue Classification Model Based On Machine Learning Using Speech Signals

이수화\*, 권철홍\*\*

Soo Hwa Lee\*, Chul Hong Kwon\*\*

**요약** 피로는 개인의 능력을 저하되게 하여 업무 수행을 어렵게 하며, 피로가 누적되면 집중력이 저하되어 안전사고를 초래할 가능성이 증가하게 된다. 피로에 대한 자각은 주관적이거나, 실제 현장에서는 피로의 수준을 정량적으로 측정할 필요가 있다. 기존 연구에서 피로 수준은 다원적 피로 척도와 같은 주관적 평가에, 생체신호 분석 등의 객관적 지표를 추가하여 전문가의 판단으로 측정하는 방식이 제안되었으나, 이러한 방법은 일상생활에서 실시간으로 피로도를 평가하기 어렵다. 본 논문은 현장에서 녹음한 음성 데이터를 이용하여 실시간으로 작업자의 피로 수준을 판정하는 피로도 분류 모델에 관한 연구이다. 현장에서 수집한 음성 데이터를 이용하여 로지스틱 분류, 서포트 벡터 머신, 랜덤 포레스트 등의 기계학습 모델을 학습시킨다. 성능을 평가한 결과, 정확도가 0.677 ~ 0.758로 우수한 성능을 보여주었고, 이 중에서 로지스틱 분류가 가장 우수한 성능을 나타냈다. 실험 결과로부터 음성신호를 이용하여 피로도를 분류하는 것이 가능하다는 것을 알 수 있다.

**주요어** : 피로도 분류 모델, 음성신호, 기계학습

**Abstract** Fatigue lowers an individual's ability and makes it difficult to perform work. As fatigue accumulates, concentration decreases and thus the possibility of causing a safety accident increases. Awareness of fatigue is subjective, but it is necessary to quantitatively measure the level of fatigue in the actual field. In previous studies, it was proposed to measure the level of fatigue by expert judgment by adding objective indicators such as bio-signal analysis to subjective evaluations such as multidisciplinary fatigue scales. However this method is difficult to evaluate fatigue in real time in daily life. This paper is a study on the fatigue classification model that determines the fatigue level of workers in real time using speech data recorded in the field. Machine learning models such as logistic classification, support vector machine, and random forest are trained using speech data collected in the field. The performance evaluation showed good performance with accuracy of 0.677 to 0.758, of which logistic classification showed the best performance. From the experimental results, it can be seen that it is possible to classify the fatigue level using speech signals.

**Key words** : Fatigue Classification Model, Speech Signals, Machine Learning

\*정회원, (주)코그넷나인 전문위원 (제1저자)

\*\*정회원, 대전대학교 정보통신·전자공학과 교수 (교신저자)

접수일: 2022년 9월 27일, 수정완료일: 2022년 10월 25일

게재확정일: 2022년 11월 1일

Received: September 27, 2022 / Revised: October 25, 2022

Accepted: November 1, 2022

\*\*Corresponding Author: [chkwon@dju.ac.kr](mailto:chkwon@dju.ac.kr)

Dept. of Information, Communication, Electronics Engineering,  
Daejeon Univ, Korea

## I. 서론

피로는 일상생활에서 겪게 되는 보편적인 정신적이며 신체적인 현상으로 개인의 능력을 저하되게 하여 업무 수행을 어렵게 한다. 피로는 과로로 정신이나 몸이 지쳐 힘든 상태를 말한다[1]. 즉 일을 과도하게 하여 정신뿐만 아니라 육체적으로 탈진한 상태를 말한다. 피로에 대한 자각은 주관적이며 심리적, 생리적, 인지적 변화와 관련된 여러 요인의 영향을 받으며, 이러한 측면들이 통합되어 하나의 피로로 자각된다. 이러한 피로의 주관적인 특징으로 인해 피로 수준을 객관적으로 측정하는 데는 어려움이 따른다. 그러나 실제 현장에서는 피로의 수준을 정량적으로 측정할 필요가 생긴다. 예를 들어, 트럭이나 버스 등을 운행하는 운전기사의 피로는 안전 운행과 밀접한 관계가 있으며, 피로가 누적되면 운전기사의 집중력이 저하되므로 안전사고를 초래할 가능성이 증가하게 된다.

주관적 피로를 측정하는 방법으로 개인이 느끼는 피로를 설문지를 통해 평가하는 다원적 피로 척도가 널리 사용되고 있다. 객관적으로 피로를 측정하는 방법에는 생체신호인 근전도, 혈중 호르몬 수준, 호흡, 심박수 변동, 체온, 혈압, 맥파, 중추 신경계의 반응, 뇌파, 동공의 변화를 평가하는 생리적 피로도 측정 방법이 연구되고 있다. 따라서 피로 수준은 다원적 피로 척도와 생체신호 분석을 종합해야 정확한 판정이 가능하다. 주관적 평가에 객관적 지표를 추가하여 피로 수준을 평가하기 위해서는 관련 전문가의 판단이 필요하여 시간이 소요되므로, 일상생활에서 실시간으로 피로 수준을 평가하는 것은 어려운 일이다.

본 논문은 피로로 인해 발생할 수 있는 안전사고 예방을 위하여 현장에서 음성 데이터를 녹음하여 실시간으로 작업자의 피로 수준을 판정하는 피로도 분류 모델에 관한 연구이다. 스마트폰 등을 이용하여 수집하기 쉬운 음성신호를 이용하고, 피로도 분류를 위해 기계학습 기반 모델을 연구한다.

본 논문에서는 1장 서론에 이어 2장에서 본 연구에서 적용한 기계학습 모델을 설명한다. 3장에서 실험에 사용한 음성 DB와 피로도 참값을 도출하는 방법을 기술하고, 4장에서 각 모델을 학습하는 방법을 설명한다. 5장에서 본 논문에서 제안한 기계학습 모델의 성능 평가 결과를 제시하며, 6장에서 결론을 맺는다.

## II. 기계학습 모델

이 절에서는 본 연구에서 적용한 기계학습 모델을 설명한다. 사용한 모델은 로지스틱 분류(Logistic Classification), 서포트 벡터 머신(Support Vector Machine), 의사결정나무(Decision Tree)와 랜덤 포레스트(Random Forest) 등이다.

### 1. 로지스틱 분류

$p$  차원 입력 벡터  $\vec{x} = [x_1, x_2, \dots, x_p]^T$ 와 가중치 (Weight) 벡터  $\vec{w} = [w_1, w_2, \dots, w_p]^T$ 의 선형 조합은 다음과 같이 나타낼 수 있다.

$$z = \vec{w}^T \vec{x} \quad (1)$$

Odds ratio는 특정 이벤트가 발생할 확률( $P$ )과 그 이벤트가 발생하지 않을 확률( $1-P$ )의 비율이다. Logit 함수는 Odds에 자연로그를 취한 것이다.

$$\text{logit}(P) = \ln \frac{P}{1-P} \quad (2)$$

로지스틱 분류에서는 입력 샘플  $x$ 가 주어졌을 때 출력 클래스  $y$ 가  $y=1$ 에 속할 확률의 Logit 함수를  $z$ 와 등가로 놓는다.

$$\text{logit}(P(y=1|x)) = z \quad (3)$$

따라서 로지스틱 분류에서는 어떤 샘플이 특정 클래스에 속할 확률  $P$ 를 다음과 같이 구할 수 있는데, 이를 로지스틱 가정(Hypothesis)이라고 한다.

$$P = \frac{1}{1+e^{-z}} = H(\vec{x}) \quad (4)$$

$1/(1+e^{-z})$ 는 로지스틱 시그모이드(Logistic Sigmoid) 또는 시그모이드 함수이다. 이 과정을 살펴보면, 입력과 가중치를 선형 조합한 다음에 시그모이드 함수를 취한 것을 알 수 있다.

로지스틱 분류에서는 손실(Loss) 또는 비용(Cost) 함수로 다음과 같은 함수를 사용한다.

$$Loss(H(x), y) = \begin{cases} -\ln(H(x)), & y = 1 \\ -\ln(1-H(x)), & y = 0 \end{cases} \quad (5)$$

$$Loss(H(x), y) = -y \ln(H(x)) - (1-y) \ln(1-H(x))$$

이 손실 함수는 이진 크로스 엔트로피(Binary Cross Entropy)와 같은 형태를 보인다. 로지스틱 분류에서는 경사 하강법(Gradient Descent) 등과 같은 최적화 기법을 사용하여 이 손실 함수를 최소화하는 과정을 통해 가중치의 최적값을 구한다.

기계학습에서 자주 발생하는 문제 중에 과적합(Overfitting)이 있다. 이것은 훈련 데이터에 과도하게 학습이 되어 훈련 데이터에는 성능이 좋으나, 별도의 테스트 데이터에서는 성능이 나쁜 경우를 말한다. 과적합을 방지하기 위해 규제(Regularization)를 사용하는데, 규제는 가중치 값을 제한하는 방법이다. 식 (5)의 손실 함수를 다음과 같이 변경하여 사용한다.

$$Loss(H(x), y) = \text{식 (5)} + \frac{\lambda}{2} \|w\|^2 \quad (6)$$

$\lambda$ 는 규제의 정도를 나타내며, 손실 함수를 최소화하는 과정에서 가중치는 작은 값을 갖게 된다.

## 2. 서포트 벡터 머신

서포트 벡터 머신은 클래스를 구분하는 마진(Margin)을 최대로 하는 초평면(Hyperplane)을 구하는 기계학습 방법이다. 초평면에 가장 가까운 결정 경계에 있는 샘플을 서포트 벡터라고 하며, 각 클래스 서포트 벡터 사이의 거리를 마진이라고 한다.

두 클래스의 서포트 벡터  $\{\vec{x}_p, \vec{x}_n\}$ 를 지나는 결정 경계 초평면은 다음과 같이 표현할 수 있다.

$$\vec{w}^T \vec{x}_p - b = +1 \quad (7)$$

$$\vec{w}^T \vec{x}_n - b = -1$$

두 식을 빼서 정규화하면 다음과 같다.

$$\frac{\vec{w}^T (\vec{x}_p - \vec{x}_n)}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|} \quad (8)$$

좌측 항은 서포트 벡터 사이의 거리 즉 마진을 나타내고, 서포트 벡터 머신은 우측 항을 최대화하여 마진을 최대화하는 방법이다.

실제로는 Quadratic 프로그래밍을 이용한 최적화 방법을 이용하여, 식 (8) 우측항의 역수인  $\|\vec{w}\|^2/2$ 을 최소로 하는  $\vec{w}$ 를 구한다. 즉 최소화해야 하는 목적 함수는 다음과 같다.

$$\arg \min_{(w,b)} \frac{1}{2} \|\vec{w}\|^2 \quad (9)$$

모든 훈련 샘플을 두 가지 클래스로 분류할 수 있는 초평면을 구하기는 쉽지 않다. 따라서 슬랙(Slack) 변수  $\xi$ 를 도입하여 일부 샘플이 다른 클래스에 포함되거나, 결정 경계 사이에 존재하는 것을 허용하는 방법을 사용한다. 이것을 소프트 마진 분류라고 하며, 식 (9)의 목적 함수는 다음과 같이 변경된다.

$$\arg \min_{(w,b,\xi)} \left\{ \frac{1}{2} \|\vec{w}\|^2 + C \sum \xi_i \right\} \quad (10)$$

$C$ 는 분류 오류에 대한 비용의 정도를 나타낸다.

이와 같은 방법은 선형적으로 분류되는 데이터 집합에는 적절하지만, 비선형적인 분포를 띄는 데이터 집합인 경우는 커널 함수를 이용하여 데이터를 고차원 선형 공간으로 변환시킨 다음에 위의 과정을 수행한다. 커널 함수에는 radial basis function(RBF), polynomial, sigmoid, linear 등이 사용된다. 본 연구에서 사용한 RBF 커널은 가우시안 커널이라고 불리며 다음 식과 같다.

$$K(\vec{x}, \vec{x}') = \exp(-\gamma \|\vec{x} - \vec{x}'\|^2) \quad (11)$$

$\gamma$ 는 가우시안 분포에서 분산의 정도를 나타낸다.

## 3. 의사결정나무와 랜덤 포레스트

의사결정나무는 예/아니오 질문을 통해 데이터를 유사한 그룹으로 분할 하면서 이진 나무 구조를 형성하여 클래스를 추정하는 방법이다. 나무의 루트(Root)에서 훈련 데이터의 특성을 이용하여 임의로 분할 기준을

제시하여, 정보 이득이 최대가 되는 또는 불순도가 최소가 되는 분할 기준을 선정한다. 자식 마디에서 나뭇가지가 순수해질 때까지 또는 제시된 분할 정지 규칙을 만족할 때까지 이런 과정을 반복 수행하여 나뭇가지를 성장시키며 훈련 데이터를 더 작은 집합으로 분할한다. 불순도 지표는 지니, 엔트로피, 분류 오차 등을 사용한다. 정지 규칙은 나무의 최대 깊이를 설정하여 과적합을 방지한다. 의사결정나무는 모델의 동작을 설명하기에 적합한 방식이다.

랜덤 포레스트는 기본 모델로 여러 개의 의사결정나무를 생성하여 각 나무에서 예측한 클래스를 모아 다수결 투표로 최종 클래스를 선택하는 앙상블 기법이다. 각 나무를 학습시킬 때 전체 훈련 데이터에서 중복을 허용하여 부분 집합을 선택한다. 이를 부트스트랩(Bootstrap) 샘플링 방법이라고 한다, 그리고 나무의 각 마디에서 훈련 데이터의 모든 특성 대신 일부 특성만으로 분할 기준을 탐색하여 각 나무가 서로 독립성이 유지되도록 만든다. 개개의 의사결정나무는 과적합 할 가능성이 크지만, 랜덤 포레스트는 나무의 수가 많을수록 훈련 데이터에 강인하면서 과적합을 방지하는 방법으로, 일반적으로 의사결정나무보다 성능이 우수하다.

### III. 음성 DB와 피로도 참값 도출

#### 1. 음성 DB

본 연구에서 사용한 음성 DB는 다음과 같다. 피험자는 남성으로 평균 조종사, 일반 군인, 경찰관, 소방관, 트럭 운전기사 등이다. 녹음 콘텐츠는 피험자가 모음 ‘아’를 2초간 2회, 연속해서 단문 1개와 장문 1개를 한번씩 발화하여 총 녹음 시간은 40초 정도이다. 음성 데이터는 사무실과 같은 업무 환경에서 수집 장비와 피험자의 거리는 1미터 내외를 유지하며, USB 형태의 지향성 마이크를 사용하여 수집한다. 음성 데이터 포맷은 샘플링 주파수 16kHz, 해상도 16비트, 모노 형식으로, 보통 속도로 발화하며 음성 샘플 크기가 10,000 ~ 25,000 사이가 되도록 발화 크기를 조정한다.

수집한 음성 파일에서 자동으로 모음 아 a1, a2, 단문 s1과 장문 s2를 분할 하여 실험에 사용한다. 먼저 녹음한 음성 파일에서 시작과 끝부분의 묵음을 제거한다. 다음에 음성 파일과 이에 해당하는 한글 텍스트 파일을 입력으로 Kaldi 툴[2]을 이용하여 자동 정렬(Forced

Alignment) 과정[3]을 수행하여 TextGrid 파일을 생성한다. TextGrid 파일에는 음소와 단어의 시작과 끝점의 시간 경계 정보가 포함되어 있다. TextGrid 파일, 한글 텍스트 파일, G2P 과정[4]을 거쳐 생성된 음소영어표기 파일을 이용하여 녹음 음성 파일을 모음, 단문, 장문 음성 파일로 분할 한다.

#### 2. 음성 특징 파라미터 추출

음성 DB를 이용하여 피로도를 측정하기 위하여 음성신호 자체가 아니라 음성신호로부터 추출한 특징 파라미터를 사용한다. 음성 특징 파라미터는 음성기술 분야에서 널리 사용하는 음원 정보 및 성도 필터 파라미터 등이다. 음성 파라미터는 Praat[5]과 VoiceSauce[6] 툴을 이용하여 모음, 단문, 장문 단위로 추출한다.

음원 정보 파라미터는 피치 주기의 역수인 기본주파수 평균값(F0\_avg), 표준편차(F0\_std), 최댓값(F0\_max), 최솟값(F0\_min), 피치 주기 변화율인 Jitter, 각각 3개와 5개 프레임 Jitter 평균인 RAP(Relative Average Perturbation)와 PPQ(Pitch Perturbation Quotient), 음성신호의 진폭 변화율인 Shimmer, 각각 3개, 5개, 11개 프레임 Shimmer 평균인 APQ3(Amplitude Perturbation Quotient), APQ5, APQ11, 각 하모닉의 진폭(H1, H2, H4), 각 하모닉의 진폭 차이(H1-H2, H2-H4), 주기성의 강도를 나타내는 CPP(Cepstral Peak Prominence) 등이다.

성도 필터 파라미터는 포먼트 주파수 4개(F1, F2, F3, F4), 각 포먼트 대역폭(B1, B2, B3, B4), 각 포먼트의 진폭(A1, A2, A3, A4), 장구간 스펙트럼 기울기인 LTAS(Long-term Average Spectral Slope), 각 주파수 대역에서의 하모닉과 잡음 스펙트럼의 비율인 HNR(Harmonic-to-Noise Ratio) 4개(HNR05, HNR15, HNR25, HNR35) 등이다.

#### 3. 피로 참값 도출

피험자의 피로 수준을 산정하기 위하여 음성 DB를 녹음하기 전에 주관적 피로 수준을 보여주는 데이터와, 피로의 정도를 묻는 문항으로 구성된 설문지에 대한 응답 데이터를 수집한다. 먼저, 피험자 본인의 컨디션을 백분율로 표시한 값과 1에서 5까지 5개 피로 등급을 역순으로 제시한 값을 종합하여 주관적 피로 수준을 선정한다. 다음에, 설문지를 이용한 다면적 피로 척도 총점과

주관적 피로 수준을 비교 분석하여 최종적인 피로 수준 참값을 산정한다.

#### IV. 모델 학습과 성능 평가 방법

##### 1. 실험 환경과 훈련 및 테스트 DB

기계학습 모델을 학습하기 위하여 사용한 소프트웨어는 Python 버전 3.7.13, Numpy 버전 1.21.6, Scikit-learn 버전 1.0.2 등이다[7].

본 연구는 2-클래스 분류 문제를 다루며, 피로 수준 1은 ‘피로하지 않다’로서 클래스 0, 피로 수준 5는 ‘매우 피로하다’로서 클래스 1로 설정한다. 실험을 진행하기 전에 임의로 훈련과 테스트 DB를 분할 하여 사용한다. 학습에 사용된 훈련 DB 샘플 수는 클래스 별로 138개씩 276개이며, 테스트 DB 샘플 수는 31개씩 62개이다.

##### 2. 모델 학습 방법

###### 2.1 로지스틱 분류

먼저 훈련 샘플을 랜덤하게 섞고 훈련과 테스트 데이터의 평균을 0, 분산을 1로 정규화 한다. 다음에 로지스틱 분류 모델을 기본값으로 세팅하고, 5-fold 교차검증 그리드 검색 방법을 이용하여 훈련 DB에서 모델을 최적화하는 파라미터를 찾는다. 이 파라미터는 규제 강도를 나타내는 식 (6)  $\lambda$ 의 역수인  $C$ 인데, 큰  $C$  값은 규제가 작아지고 작은 값은 규제가 커진다. 이렇게 구한 최적의  $C$  값으로 테스트 DB에서 로지스틱 분류 모델의 성능을 평가한다.

###### 2.2 서포트 벡터 머신

우선 훈련 샘플을 섞고 훈련과 테스트 샘플의 크기를 [0, 1] 범위로 조정한다. 이것은 샘플의 모든 특성의 영향을 균등하게 하면서 원래 데이터의 분포를 유지하는 스케일링 방법이다. 서포트 벡터 머신 모델은 커널 함수로 RBF를 사용하여 기본값으로 세팅하고, 5-fold 교차검증 그리드 검색 방법을 이용하여 최적의 하이퍼-파라미터를 구한다. 이 파라미터는 식 (10)의  $C$ 와 식 (11)의  $\gamma$ 이다.

###### 2.3 의사결정나무와 랜덤 포레스트

의사결정나무와 랜덤 포레스트 모델을 학습하기 위해 스케일링은 하지 않는데, 이것은 분할 기준에 사용

되는 특성의 값을 그대로 이용하기 위해서이다. 먼저 훈련 샘플을 섞고 의사결정나무와 랜덤 포레스트 모델은 불순도 지표로 지니를 사용하여 기본값으로 세팅한다. 5-fold 교차검증 그리드 검색 방법을 이용하여 최적의 하이퍼-파라미터를 구하는데, 이 파라미터는 의사결정나무에서는 나무의 최대 깊이이고, 랜덤 포레스트 모델에서는 의사결정나무 분류기의 수, 부트스트랩 훈련 샘플의 수, 분할 기준에서 탐색 되는 특성의 수이다.

##### 3. 모델 성능 평가 방법

5-fold 교차검증 그리드 검색 방법을 이용하여 훈련 데이터에서 최적의 모델 하이퍼-파라미터를 탐색할 때 성능을 측정하는 방법과, 테스트 데이터에 대한 모델의 성능을 평가하는 방법은 다음과 같다.

먼저 혼동 행렬(Confusion Matrix)은 표 1과 같다. 표에서 피로 수준 1은 N, 피로 수준 5는 P로 나타낸다.

표 1. 혼동 행렬  
 Table 1. Confusion matrix

		예측 값	
		N	P
참값 (실제값)	N	True Negative (TN)	False Positive (FP)
	P	False Negative (FN)	True Positive (TP)

정확도(Accuracy)는 전체 데이터에서 모델이 올바르게 예측한 데이터의 비율로서 다음 식으로 구할 수 있다.

$$Accuracy = \frac{TN + TP}{TN + FN + FP + TP} \quad (12)$$

정밀도(Precision)는 모델이 피로 수준 5로 예측한 것 중에서 참값 5를 올바르게 예측하는 비율이고, 재현율(Recall)은 피로 수준 참값이 5일 때 모델이 피로 수준 5를 올바르게 예측하는 비율이다. F1 스코어는 정밀도와 재현율의 조화 평균으로 두 가지 척도의 장단점을 보완해 주며, 모델의 성능을 측정하는 데 널리 사용되는 평가 지표이다.

$$Precision = \frac{TP}{FP + TP} \quad (13)$$

$$Recall = \frac{TP}{FN + TP}$$

$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall}$$

TPR(True Positive Rate)과 FPR(False Positive Rate)은 다음과 같다.

$$TPR = \frac{TP}{FN + TP} = Recall \quad (14)$$

$$FPR = \frac{FP}{TN + FP}$$

ROC 곡선(Receiver Operating Characteristic Curve)은 가로축이 FPR, 세로축이 TPR인 그래프에서 FPR이 변할 때 TPR이 변화하는 형태를 보여준다. FPR이 0이며 TPR이 1인 모델이 완전한 성능을 나타낸다. ROC\_AUC(Area Under Receiver Operating Characteristic Curve)는 ROC 곡선 아래의 면적을 나타내며, 1에 가까운 값일수록 좋은 성능을 보여준다. ROC\_AUC는 분류 모델을 선택할 때 유용한 지표이다.

본 연구에서는 최적의 모델 하이퍼-파라미터를 탐색할 때 ROC\_AUC 스코어를 사용하고, 테스트 데이터에 대한 모델의 성능은 혼동 행렬, 정확도, 정밀도, 재현율과 F1 스코어를 사용하여 평가한다.

### V. 모델 성능 평가

테스트 DB를 이용하여 본 연구에서 제안한 기계학습 모델의 성능을 평가한 결과는 다음과 같다.

로지스틱 분류의 성능을 평가한 결과, 혼동 행렬은 표 2에 보이고 정확도는 0.758, 정밀도는 0.808, 재현율은 0.677, F1 스코어는 0.737이다. 서포트 벡터 머신의 성능을 실험한 결과, 표 3에 혼동 행렬이 보이고 정확도는 0.710, 정밀도는 0.741, 재현율은 0.645, F1 스코어는 0.690이다. 의사결정나무의 성능을 평가한 결과, 혼동 행렬은 표 4에 보이고 정확도는 0.629, 정밀도는 0.833, 재현율은 0.323, F1 스코어는 0.465이다. 피로 수준 1로 치중되어 예측되고, F1 스코어가 0.5 보다 작은

결과를 보인다. 랜덤 포레스트의 성능을 실험한 결과, 표 5에 혼동 행렬이 보이고 정확도는 0.677, 정밀도는 0.667, 재현율은 0.710, F1 스코어는 0.688이다. 표 4와 5를 비교해보면 의사결정나무보다 성능이 향상됨을 알 수 있다.

표 2. 로지스틱 분류에서 혼동 행렬

Table 2. Confusion matrix in logistic classification

		피로수준 예측값	
		1	5
피로 수준참값	1	0.84 (26)	0.16 (5)
	5	0.32 (10)	0.68 (21)

표 3. 서포트 벡터 머신에서 혼동 행렬

Table 3. Confusion matrix in support vector machine

		피로수준 예측값	
		1	5
피로 수준참값	1	0.77 (24)	0.23 (7)
	5	0.35 (11)	0.65 (20)

표 4. 의사결정나무에서 혼동 행렬

Table 4. Confusion matrix in decision tree

		피로수준 예측값	
		1	5
피로 수준참값	1	0.94 (29)	0.06 (2)
	5	0.68 (21)	0.32 (10)

표 5. 랜덤 포레스트에서 혼동 행렬

Table 5. Confusion matrix in random forest

		피로수준 예측값	
		1	5
피로 수준참값	1	0.65 (20)	0.35 (11)
	5	0.29 (9)	0.71 (22)

표 6. 제안한 기계학습 모델의 성능 비교

Table 6. Performance comparison of the proposed machine learning models

	로지스틱 분류	서포트 벡터 머신	랜덤 포레스트
정확도	0.758	0.710	0.677
정밀도	0.808	0.741	0.667
재현율	0.677	0.645	0.710
F1 스코어	0.737	0.690	0.688

표 6에 본 연구에서 제안한 기계학습 모델의 성능이 비교되어 있다. 로지스틱 분류는 정확도, 정밀도, F1 스코어에서, 랜덤 포레스트는 재현율이 가장 우수한 성능을 보인다.

## VI. 결 론

일상생활에서 실시간으로 피로 수준을 측정하기 위하여, 본 논문에서는 음성 데이터를 이용하여 피로도를 분류하는 방식을 제안한다. 본 연구에서는 로지스틱 분류, 서포트 벡터 머신과 랜덤 포레스트 등의 기계학습 모델을 사용하였는데, 이것은 수집한 음성 데이터의 양을 고려하여 선택한 방식이다.

본 연구는 ‘피로하지 않다’를 피로 수준 1로, ‘매우 피로하다’를 피로 수준 5로 설정하여 2-클래스 분류 문제를 다룬다. 수집된 음성 DB를 사전에 훈련 DB와 테스트 DB로 나누어 실험에 사용한다. 음성 데이터에서 음성 특징 파라미터를 추출하여 기계학습 모델을 학습 시킨다.

성능을 평가한 결과, 로지스틱 분류, 서포트 벡터 머신과 랜덤 포레스트는 정확도가 0.677 ~ 0.758로 좋은 성능을 보여주었다. 이 중에서 로지스틱 분류가 가장 우수한 성능을 보여주었는데, 이것은 훈련에 사용된 데이터 크기와 관련이 있다고 볼 수 있다.

본 연구의 실험 결과로부터 음성신호를 이용하여 피로도를 분류하는 것이 가능하다는 것을 알 수 있으며, 향후 음성 데이터를 추가로 수집하여 딥러닝 방법을 이용한 피로도 분류 모델을 연구할 계획이다. 로지스틱 분류에서 우수한 성능을 보여 준 결과로부터, 딥러닝에서도 좋은 성능을 기대할 수 있다고 볼 수 있다.

## References

[1] U. S. Choi and M. S. Song, "Concept analysis: Fatigue," *Korean Journal of Women Health Nursing*, Vol. 9, No. 1, pp. 61-69, 2003

[2] Kaldi speech recognition toolkit, [Computer program] retrieved August 2022 from <https://github.com/kaldi-asr/kaldi>

[3] Korean forced aligner, [Computer program] retrieved August 2022 from <http://hyungwonsnot.ebook.blogspot.com/2017/05/korean-forced-aligner.html>

[4] Y. W. Lim, J. R. Cho, J. M. Lee, M. W. Koo, "The development of grapheme-to-phoneme conversion based on LSTM for Korean language," *Proceedings of the Korea Software Congress 2017*, pp. 2,004-2,006, 2017

[5] P. Boersma and D. Weenink, "Praat: doing phonetics by computer," [Computer program]. version 6.2.06, retrieved August 2022 from <https://www.fon.hum.uva.nl/praat/>

[6] Y. L. Shue, P. Keating, C. Vicenik, K. Yu "VoiceSauce: a program for voice analysis," *Proceedings of the ICPhS XVII, 1846-1849*, 2011, [Computer program]. version 1.37, retrieved August 2022 from <http://www.phonetics.ucla.edu/voicesauce/>

[7] S. Raschka and V. Mirjalili, *Python machine learning - third edition*, Packt Publishing, 2019

※ 본 연구는 대한민국 정부(산업통상자원부 및 방위사업청)의 재원으로 민군협력진흥원에서 수행하는 민군겸용기술개발사업의 연구비 지원으로 수행되었습니다. (과제번호 20-CM-BD-13)