

# 기계학습을 이용한 Joint Torque Sensor 기반의 충돌 감지 알고리즘 비교 연구

## A Comparative Study on Collision Detection Algorithms based on Joint Torque Sensor using Machine Learning

조성현<sup>1</sup>, 권우경<sup>†</sup>

Seonghyeon Jo<sup>1</sup>, Wookyong Kwon<sup>†</sup>

**Abstract:** This paper studied the collision detection of robot manipulators for safe collaboration in human-robot interaction. Based on sensor-based collision detection, external torque is detached from subtracting robot dynamics. To detect collision using joint torque sensor data, a comparative study was conducted using data-based machine learning algorithm. Data was collected from the actual 3 degree-of-freedom (DOF) robot manipulator, and the data was labeled by threshold and handwork. Using support vector machine (SVM), decision tree and k-nearest neighbors KNN method, we derive the optimal parameters of each algorithm and compare the collision classification performance. The simulation results are analyzed for each method, and we confirmed that by an optimal collision status detection model with high prediction accuracy.

**Keywords:** Collision Detection, Machine Learning, KNN, SVM, DT

### 1. 서 론

제4차 산업 혁명 시대가 도래하면서, 기존의 산업용 로봇이 아닌 인간과 협력이 가능한 로봇인 협동 로봇(Cobot, Collaboration Robot)이 등장했다<sup>[1]</sup>. 협동 로봇은 기존의 정형화된 작업뿐만 아니라 사람과 공간을 공유하여 능동적인 업무를 수행할 수 있다. 그만큼 로봇은 작업자와의 물리적 접촉이 불가피하며, 이에 따른 작업 중의 사람과 로봇의 충돌에 대한 연구가 많이 진행되고 있다<sup>[2]</sup>.

로봇의 충돌은 크게 사전 충돌, 감지, 분리, 인지, 분류, 반응, 충돌 후로 총 7단계의 프로세스로 나눌 수 있다. 하지만 일반적으로 로봇은 인간보다 빠르게 움직이기 때문에 사전에 충돌을 보장하기 어렵다<sup>[3]</sup>. 따라서 본 논문에서는 로봇 충돌의 주

요 단계인 충돌 감지에 대해서 크게 중점을 둔다. 충돌 감지는 로봇과 인간의 물리적인 접촉으로 인한 부상을 최소화하기 위해 실시간으로 로봇의 충돌 발생 여부를 감지하는 단계이다. 이 단계에서는 최소한의 오탐지와 민감한 충돌 감지를 위해 신호에 임계치를 선택하는 방법이 가장 실질적이며, 사용하는 신호에 따라 크게 센서리스 충돌 감지 방식과 센서 기반의 충돌 감지 방식으로 접근할 수 있다<sup>[3,4]</sup>.

센서리스 충돌 감지는 외부 센서를 이용하지 않고 내장된 모터와 제어기의 정보만을 사용하여 사람과 로봇의 충돌을 감지하는 것이다. 대표적으로 전류 신호 기반의 충돌 감지는 실제 모터의 전류와 예상되는 전류의 차이 또는 관측기 설계 기반으로 충돌을 감지하는데, 토크의 임계 값이 낮을수록 부정확하고 외력 추정을 위한 복잡한 연산으로 응답속도가 느리다<sup>[5,6]</sup>. 이를 개선하기 위한 방법으로 모멘텀 기반의 센서리스 충돌 감지가 있으며, 로봇의 동역학과 마찰 모델을 이용하여 로봇의 충돌을 감지한다. 그러나 복잡한 수식과 연산을 필요로 하고 시스템 모델의 오류를 줄여야 한다<sup>[7,8]</sup>. 대부분의 센서리스 충돌 감지는 센서 기반의 충돌 감지에 비해 외부 센서를 이용하지 않아 비용이 경제적이지만 모델의 정확성이 떨어지며, 응답 속도가 느리기 때문에 충돌 감지 성능이 떨어진다. 또한

Received : Jan. 22. 2020; Revised : Feb. 18. 2020; Accepted : Feb. 21. 2020

※ This project was funded by Electronics and Telecommunications Research Institute (ETRI), and is currently supported by the Korean government. [20ZD1110, Development of ICT Convergence Technology for Daegu-Gyeongbuk Regional Industry]

1. MS Students, Kyungpook National University, Daegu, Korea (nwlliterate@knu.ac.kr)

† Researcher, Corresponding author: Daegu-Gyeongbuk research center, Electronics and Telecommunications Research Institute (ETRI), Daegu, Korea (wkwon@etri.re.kr)

관절의 마찰력은 시간에 따라 변화하며 비선형성을 띠어 정확히 추정하는 데 한계가 있다.

센서 기반의 충돌 감지는 광범위하게 연구되고 있다. 그 예로서, 비전 센서를 이용한 충돌 감지는 충돌이 발생하기 전 로봇의 궤적을 감속 및 변경함으로써 충돌을 사전에 방지할 수 있다. 하지만 카메라의 위치를 자세히 조정할 필요가 있고 인지 영역에서 다소 부정확하다<sup>[10-12]</sup>. 또 다른 방법인 촉각 센서를 이용한 충돌 감지는 로봇의 모든 부분에서 충돌을 감지할 수 있으나 센서의 배선이 많이 복잡하고 계산해야 될 센서의 데이터양이 많아 큰 연산량이 필요하다<sup>[13]</sup>. 이에 비해 토크 센서는 관절 별로 부착되어 센서의 데이터가 많지 않고 출력부에서 측정하기 때문에 토크의 마찰항을 무시할 수 있어 외력을 추정하기 쉽다<sup>[14]</sup>. 이러한 이유로 토크 센서 기반의 충돌 감지가 주목을 받고 있으며, 충돌 감지의 정확도와 민감도를 높이기 위해서는 토크 센서를 이용하여 충돌 감지 알고리즘을 설계해야 한다.

많은 센서 기반의 충돌 감지 알고리즘들은 정적 임계치를 기준으로 하여 충돌을 감지한다<sup>[5-7,9,13]</sup>. 하지만 정적 임계치를 높게 설정하면 충돌에 둔감하여 임계치 미만의 미세한 충돌은 감지할 수 없으며, 반대로 임계치를 낮게 설정하면 잦은 오탐지로 작업의 효율을 떨어뜨리게 된다<sup>[15]</sup>. 따라서, 이러한 문제점을 보완하기 위해서 본 논문에서는 동적 임계치를 기준으로 하여 충돌을 분류하고 신호 처리에서 많이 사용되고 있는 기계학습 기법 중 하나인 분류 알고리즘을 적용하여 충돌 감지 알고리즘을 제안한다.

제안 방법론으로는 크게 3단계로 1) 데이터 수집 및 전처리, 2) 충돌 감지를 위한 분류 모델 개발, 3) 최적의 충돌 판별 모델 선정으로 프로세스를 진행하였다. 이를 위해, 조인트 토크 센서를 이용하여 매니퓰레이터의 외력을 수집하였다. 수집된 데이터는 충돌(Collision)과 비충돌(Non-collision) 두 가지 상태로 정의하였으며, 지도 학습 알고리즘인 KNN, DT, SVM을 사용하여 학습시켰다. 또한 학습된 모델을 가지고 각각의 Hyperparameter를 선정하여 최적화함으로써 최적의 모델을 비교 분석하여 충돌 감지 알고리즘을 설계하였다.

## 2. 연구 대상 및 데이터 처리

본 장에서는 충돌 감지 연구의 대상과 정의된 충돌 상태 분류를 하기 위한 데이터 수집 및 전처리 과정을 소개한다.

### 2.1 연구 대상 구성

본 연구에서는 충돌 감지 알고리즘을 구현하기 위하여, [Fig. 1]과 같은 형태의 3-DOF 매니퓰레이터를 바탕으로 연구를



[Fig. 1] Shape of 3-dof Manipulator

[Table 1] Datasheet on the 3-dof Manipulator actuator

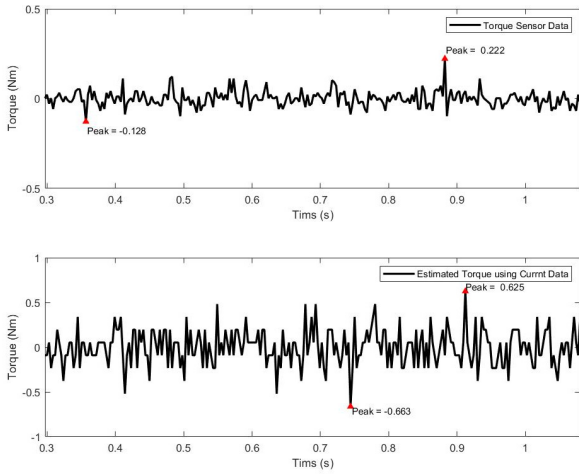
Series	SSHG-25	SSHG-32
Rating Torque	8.9 kgfm	18 kgfm
Average Maximum	14 kgfm	29 kgfm
Weight	4.3 kg	7.3 kg
Motor	300 W	500 W
RPM	3000 rpm	2500 rpm
Driver	Sinusoidal PWM control (16 kHz)	
Power	DC 48 V	
Comm. I/F	EtherCAT	

[Table 2] Datasheet on the 3-dof manipulator joint torque sensor

Series	TSR-0100S	TSR-0200S
Capacity	100 Nm	200 Nm
Stiffness	1940 Nm/°	4770 Nm/°
Accuracy	0.2% F.S	
Environment Temp.	-10 °C ~ 60 °C	
Supply Voltage	4.5 V ~ 6 V	
Communication	EtherCAT	

진행한다. 이 매니퓰레이터는 직접 제작되었으며, Harmonic Drive를 3개를 사용하여 3자유도를 가진다. 최대 도달거리는 1410 mm이며, 무게는 48 kg, 하중은 10 kg이다. 액츄에이터는 아래 [Table 1]과 같이 500 W MR32와 300 W MR25 모터를 사용했고, 감속을 위하여 100:1 기어비를 갖는 삼익 HDS의 SHG-25-100과 SHG-32-100 모델의 고 토크용 하모닉 드라이버를 사용한다.

토크 센서는 하모닉 드라이브에 직결되도록 구성되어 있으며, 위 [Table 2]와 같이 AL Robot사에 TSR-0200S와 TSR-0100S를 사용한다. 또한, 아래 [Fig. 2]는 토크 센서(위)와 전류 신호로



[Fig. 2] Comparison of estimated torque using current signals and joint torque sensor signals. The peak of estimated torque from current signal is 0.625 Nm and joint torque sensor is 0.222 Nm

추정된 토크(아래)를 비교하는 그래프이다. 토크의 Peak를 비교했을 때, 토크 센서의 Peak가 전류 신호로 추정된 토크보다 약 0.4 Nm 차이가 있다는 것을 알 수 있으며, 이는 충돌 감지 시에 토크 센서 신호가 전류 신호로 추정된 토크보다 미세한 충돌을 판별할 수 있어 충돌 판별에 적합하다는 것을 알 수 있다. 전류 신호로 추정 토크를 구하는 수식은 아래 식 (1)과 같다.  $\tau_{cs}$ 은 전류 신호로 추정된 토크이고  $K_t$ 는 토크 상수,  $I$ 는 전류 신호이다. 1축과 2축은 동일한 모터를 사용하였으며, 토크 상수 값은 각각 1축, 2축은  $K_t = 0.1768$ , 3축은  $K_t = 0.1461$ 의 값을 가진다.

$$\tau_{cs} = K_t \cdot I \tag{1}$$

### 2.2 데이터 수집

충돌 감지 알고리즘을 구현하기 위해 데이터 수집은 앞서 3-DOF 매니퓰레이터를 가지고 20 m<sup>3</sup> 크기의 공간에서 실험을 수행한다. 매니퓰레이터는 EtherCAT 통신으로 1 ms 단위로 다양한 항목의 데이터를 모니터링할 수 있으며, 본 실험에서는 시간, 각관절 별 위치, 전류 신호, 토크 센서 신호 등의 총 19 개의 항목을 수집하고 1 ms 간격으로 신호를 저장한다.

충돌 실험은 작업자와 충돌이 잦은 2축과 3축을 집중적으로 충돌시켰으며, 매니퓰레이터의 위치마다 로봇 동역학 모델의 오차가 발생하기 때문에 각기 다른 각도에서 실험을 진행한다. 위치는 300번 정도 옮기면서 데이터 수집했으며, 각 위치마다 토크 센서를 기준으로 일정 거리를 두어 7번 충돌시켰다. 관측된 충돌은 총 2100번이며, 관측 데이터 셋으로는 1,400,000개를 가지고 연구를 진행한다.

### 2.3 데이터 전처리

로봇의 충돌에 대해서 판별하기 위해서는 수집한 데이터로부터 외부에서 발생한 토크를 검출해야 한다. 따라서 다음 식은 조인트 토크 센서로부터 외부 토크를 수집하기 위한 로봇 매니퓰레이터의 동역학식이다.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_j - \tau_{ext} \tag{2}$$

식 (2)의 각 매트릭스는 관성, 구심력, 중력을 나타내며,  $\tau_j$ 는 조인트 토크 센서로 관측한 토크이고  $\tau_{ext}$ 는 외부에서 발생된 토크이다.

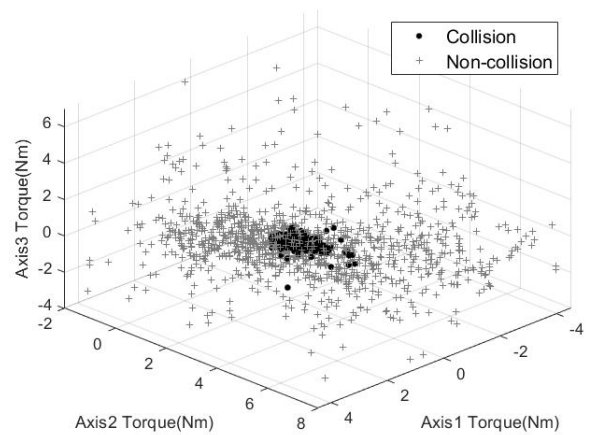
$$\begin{aligned} \tau_{ext} &= \tau_j - f(q, \dot{q}, \ddot{q}) \\ &= \hat{\tau}_{ext} - \tau_{err} \end{aligned} \tag{3}$$

식 (3)에서  $f$ 는 Newton-Euler 공식을 이용하여 관절의 위치, 속도, 가속도를 통해 동역학 모델의 토크를 계산한다<sup>16)</sup>.  $\tau_{err}$ 는  $f$ 에서 구해진 동역학 모델과 실제 동역학 모델의 차이이다. 추정된  $\hat{\tau}_{ext}$ 로부터 충돌의 감지하기 위해서는 임계치를 설정해야 하며, 다음과 같이 동적 임계치에 따라 충돌 여부를 설정한다.

$$\begin{aligned} \mu(t) &\rightarrow \{True, False\} \\ \mu_i(t) &= \begin{cases} True, & \text{if } |\mu_i(t)| > \epsilon_k \\ False, & \text{if } |\mu_i(t)| \leq \epsilon_k \end{cases} \\ \mu(t) &= [\mu_1(t), \mu_2(t), \dots, \mu_p(t)] \end{aligned} \tag{4}$$

식 (4)에서  $\mu(t)$ 는 모니터링 신호를 나타내며,  $t$ 는 함수의 수집된 신호의 시간이며,  $\epsilon_k$ 은 충돌 판별을 판별하기 위한  $k$ 번째 충돌의 동적 임계치를 나타낸다<sup>17)</sup>.

$\hat{\tau}_{ext}$ 을 동적 임계치로 충돌을 감지해도 [Fig. 3]과 같이 잔여



[Fig. 3] Result of data preprocessing

진동으로 인해 충돌 한 번에 여러 번 감지하게 된다. 따라서, 다음과 같이  $k$ 번째 충돌이 시작된 시간과 끝나는 시간으로 충돌 범위를 설정한다.

$$T(x_k) = \begin{cases} 1, & \text{if } t_{start} \geq x_k \geq t_{end} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

식 (5)는  $T$ 는 시간에 따른  $k$ 번째 충돌이 일어난 여부를 0과 1로 나타내며,  $t_{start}, t_{end}$ 는  $k$ 번째 충돌이 시작하는 시간과 끝나는 시간을 나타낸다.

본 연구에서는 각 축의 센서로부터 추정된  $\hat{\tau}_{ext}$ 의 최대 토크의 범위가 각기 다르기 때문에 정규화 과정이 필요하다. 따라서 본 연구에서는 수식 식 (6)과 같이 표준화를 수행하였다.

$$\hat{\tau}_n = \frac{\hat{\tau}_n - \min(\hat{\tau}_n)}{\max(\hat{\tau}_n) - \min(\hat{\tau}_n)} \quad (n = 1, 2, 3) \quad (6)$$

앞선 데이터 전처리를 통해 [Fig. 3]과 같이 조인트 토크 센서에 따른 충돌 여부를 분류한다. 또한 충돌 여부에 따른 데이터를 충돌(Collision)과 비충돌(Non-collision)으로 나누어 저장하며 다음 장에서 충돌 판별 알고리즘 개발을 위한 응답 변수로써 사용된다.

### 3. 충돌 감지를 위한 충돌 판별 모델 개발

본 장에서는 충돌 판별을 위해 분류 알고리즘인 KNN, DT, SVM를 검증, 모델 평가를 소개하고 설명한다.

#### 3.1 분류 알고리즘

기계 학습은 크게 지도학습, 비지도학습, 강화학습의 세 가지 범주로 나눌 수 있는데, 본 논문에서 다루는 분류 알고리즘은 지도학습 중에 속한다. 지도학습은 데이터와 그에 따른 라벨이라는 원하는 목표의 데이터를 학습시킴으로써 라벨이 지정되지 않은 데이터를 학습된 모델을 통해 예측한다. 주로 데이터 분석이나 새로운 데이터를 매핑하기 위한 추론 함수로써 사용할 수 있다. 대표적으로 분류 알고리즘은 KNN (K-nearest neighbors), Linear regression, Logistic regression, SVM (Support vector machine), DT (Decision tree), Naive bayesian 등이 있다. 본 논문에서는 충돌 감지 알고리즘을 위한 구현을 위해 KNN, DT, SVM 총 3개의 분류 알고리즘을 적용한다.

##### 3.1.1 KNN (K-nearest neighbors)

KNN 알고리즘은 훈련 데이터에서 가장 가까운  $k$ 개의 이웃

[Table 3] Distances and their formulation

Distance	Formulation
Euclid	$d_{st} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Manhattan	$d_{st} = \sum_{i=1}^n  x_i - y_i $
Chebyshev	$d_{st} = \max_i  x_i - y_i $
Jaccard	$d_{st} = \frac{ x_i \cap y_i }{ x_i \cup y_i }$
Standard Euclidean	$d_{st} = \sqrt{\sum_{i=1}^n \frac{(x_i - y_i)^2}{V}}$

클래스를 거리 함수 기반으로 하여, 테스트 클래스를 추정한다<sup>[17]</sup>. 여기서, 데이터의 가까움을 측정하기 위해 [Table 3]과 같은 거리 함수들을 사용한다<sup>[18,19]</sup>. 따라서 본 논문에서는 식 (7)과 같이  $k$ 와 거리 함수를 Hyper-parameter로 사용하여 최적의 KNN 모델을 구한다.

$$\operatorname{argmax}_{d_{st}, k \in s} acc_{cv}(d_{st}, k) \quad (7)$$

##### 3.1.2 DT (Decision tree)

DT 알고리즘은 하나의 루트 노드를 중심으로 여러 개의 분기 및 자식, 터미널 노드로 구성이 되는데<sup>[20]</sup>, 알고리즘 특성상 복잡한 모델을 생성하기 때문에 분기와 노드의 개수를 적절하게 설정해야 한다<sup>[21]</sup>. 또한, 노드의 분할 방법에 따라 ID3, C4.5, CART (Classification and Regression Tree) 등과 같이 다양한 알고리즘 있으며, CART 알고리즘은 DT 방법론 중에 가장 잘 알려진 방법론 중 하나이다. CART 알고리즘은 불순도(Impurity)의 감소량에 따라 분리하는데, 불순도는 크게 지니 계수와 엔트로피 지수로 다음 식 (8)과 같이 정의할 수 있다<sup>[22]</sup>.

$$I_{gini}(x) = 1 - \sum_{i=1}^n p^2(i) \quad (8)$$

$$I_{entropy}(x) = - \sum_{i=1}^n p(i) \log_2 p(i)$$

따라서, 본 논문에서는 DT의 방법론 중 하나인 CART 알고리즘을 사용하여, 식 (9)와 같이 불순도 계산법과 터미널 노드의 최소 개수, 분기의 최대 개수를 Hyper-parameter로 사용하여 최적의 DT 모델을 구한다.

$$\operatorname{argmax}_{I, n_{\min}, B_{\max} \in s} acc_{cv}(I, n_{\min}, B_{\max}) \quad (9)$$

##### 3.1.3 SVM (Support vector machine)

마지막으로, SVM 알고리즘은 원래 이진 분류를 목적으로 고안되었으나 한계점을 해결하고자 여러 멀티클래스를 분류하는 방식이 제안되었다<sup>[23]</sup>. 하지만, 본 논문에서는 충돌과 비

충돌 두 개의 클래스로 나누어 분류하기 때문에, 멀티클래스를 분류하는 방식을 고려하지 않는다. 이진 분류 SVM 알고리즘은 데이터를 두 개의 클래스로 나누는 경계인 최적의 초평면(Hyperplane)을 찾아 분류하는 알고리즘이다<sup>[24,25]</sup>. 분류하는 초평면은 다음과 같이 정의할 수 있다.

$$f(x) = x' \beta + b \tag{10}$$

식 (10)에서 벡터  $\beta$ 는 초평면과 직교하는 벡터를 정의한 것이다. 여기서, 최적의 초평면 얻기 위해서는 클래스와 초평면 간의 관측 값이 포함되지 않는 공간을 마진(margin)이라고 하는데, 이 마진이 최대 너비를 가지는 것을 말한다. 따라서, Lagrange 최적화 방법을 사용하여, 식을 정리하면 다음과 같이 정의할 수 있다<sup>[25]</sup>.

$$\beta' = \sum_{i=1}^n a_i y_i G(x_i, x_j) + b' \tag{11}$$

with  $0 \leq a_i \leq C$  and  $1 \leq i \leq n$

식 (11)에서  $y_i$ 는 이진 클래스 라벨을 나타내며, C는 Lagrange 승수법의 제약(Constraint)이라 하고, G(.)는 커널 함수의 매트릭스를 나타낸다. SVM는 Polynomial Kernel, Sigmoid Kernel 등 다양한 커널 함수가 있지만, 충돌 데이터는 비선형의 원형 초평면을 가지기 때문에 가장 효과적인 커널 함수인 Gaussian Kernel을 사용하며, 다음 식과 같이 정의할 수 있다.

$$G(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{12}$$

식 (12)에서  $\gamma$ 는 커널의 크기를 나타내며, 높은 정확도를 얻기 위해서는 C와  $\gamma$ 의 적절하게 선택해야 한다<sup>[26]</sup>. 따라서, 본 논문에서는 식 (13)과 같이 C와  $\gamma$ 를 Hyper-parameter로 사용하여 최적의 SVM 모델을 구한다.

$$\operatorname{argmax}_{C, \gamma \in S} \operatorname{acc}_{cv}(C, \gamma) \tag{13}$$

### 3.2 검증

검증은 데이터를 통해 모델을 학습시키고 훈련이 제대로 성공했는지를 확인하기 위해서 반드시 필요한 과정이다. 훈련 모델은 학습된 데이터에 대해서 적절한 예측을 보여줄 수 있지만, 새로운 데이터를 정확하게 예측하지 못할 수도 있어 교차 검증을 진행해야 한다. 교차 검증은 일반적으로 두 부분으로 나누어 진행하며, 데이터의 일부를 모델의 학습에 이용하고 학습된 모델을 테스트하기 위해 나머지 데이터를 사용한다<sup>[27]</sup>. 교차 검증에는 Re-substitution Validation, Hold-out validation,

K-fold cross-validation, Leave one out cross-validation 등의 기법이 주로 사용되며, 본 논문에서는 K-fold Cross-validation을 적용한다. K-fold cross-validation는 데이터를 동일한 k개의 폴드로 분할하고 테스트를 위한 폴드 1개를 제외한 나머지 폴드를 학습시킨다. 이를, k번 반복 수행한다<sup>[28]</sup>. 일반적으로 k의 값이 높을수록 교차 검증의 정확도는 높아지나 과적합(Overfitting) 문제가 발생할 수 있으므로, 적절한 K 값을 설정해야 한다<sup>[29]</sup>. 따라서 본 논문에서는 K의 값을 3, 5, 10, 20으로 하여 분석하였다.

### 3.3 모델의 성능 평가

본 논문에서는 분류 모델의 성능을 평가하기 위해서 평균 제곱 오차(MSE, Mean-Squared Error)와 오차 행렬(Confusion Matrix)를 두가지를 사용한다.

#### 3.3.1 MSE (Mean-Squared Error)

평균 제곱 오차는 영상 처리나 기계학습에서 주로 사용되는 정확성을 측정하는 방법 중 하나이다. 실제 라벨 데이터와 모델이 예측한 데이터의 오차를 제곱하여 평균으로 구한다. 수식은 다음 식 (14)와 같이 정의한다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{14}$$

#### 3.3.2 오차 행렬(Confusion Matrix)

오차 행렬은 학습 모델로 예측된 데이터를 행, 실제 데이터를 열로 두고 평가하는 방식이다<sup>[30]</sup>. 이진 오차 행렬에서는 True와 False로 나누기 때문에, 각각 행렬을 TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives)으로 나눌 수 있다. 이를 통하여 정밀도(Precision), 재현율(Recall), 정확도(Accuracy) 등을 알 수 있으며<sup>[31]</sup>, 수식은 다음 식 (15)과 같이 정의한다.

$$\begin{aligned} Accuracy &= (TP + TN) / Total \\ Precision &= TP / (TP + FP) \\ Recall &= TP / (TP + FN) \end{aligned} \tag{15}$$

이를 바탕으로 하여 본 연구에서는 학습 기반의 충돌 판별 모델을 개발하였으며, 학습된 모델의 성능을 비교 분석하였다.

## 4. 충돌 판별 모델을 이용한 성능 비교

각 지도학습 알고리즘 별 Hyperparameter를 선정하였으며, 최적화된 모델들을 비교하였다.

### 4.1 최적의 Hyperparameter 설정

각각 알고리즘을 대상으로 [Table 4]와 같이 Hyper-parameter 를 선택하였으며, 그에 따른 범위를 지정하였다. 또한, 최적의 Hyperparameter을 얻기 위해서 [Table 5]와 같은 결과를 도출 하였다.

[Table 4] Hyperparameter range and types for each model. Note  $y_i$  is the number of observations

Algorithm	HP	Type	Start	End
KNN	K	Integer	1	100
	Distances	Nominal	*	
DT	Node size	Integer	1	$y_i - 1$
	Branch	Integer	1	$\frac{y_i}{2}$
	Impurity	Nominal	**	
SVM	Cost	Log	1E-4	1E10
	$\gamma$	Log	1E-11	1E2

\* Euclid, Manhattan, Chebychev, Jaccard, S-Euclidean

\*\* Gini index, Cross-entropy

[Table 5] Results of tuning hyper-parameters for each model. Note “CV Result” stands for MSE result

Model	Fold	Optimization-HP	CV Result
KNN	3	k: 31 d: S-Euclidean	0.048650%
	5	k: 29 d: Euclid	0.048933%
	10	k: 27 d: S-Euclidean	0.048179%
	20	k: 23 d: S-Euclidean	0.048200%
DT	3	node: 102 branch: 143420 I: Gini	0.047607%
	5	node:58 branch: 1935 I: Gini	0.047622%
	10	node95 branch: 2669 I: Entropy	0.04745%
	20	node:143 branch: 987 I: Gini	0.04753%
SVM	3	C: 0.017399 $\gamma$ : 0.061401	0.07358%
	5	C: 0.26981 $\gamma$ : 0.0065932	0.05225%
	10	C: 0.62055 $\gamma$ : 0.012997	0.05389%
	20	C: 1.3408 $\gamma$ : 0.014881	0.05289

튜닝 결과, 최적의 Hyperparameter로 설정된 KNN은 매개 변수 K는 20~30 내외의 값을 가지고, 거리 함수로는 Standard Euclidean을 많이 사용되었다는 것을 볼 수 있었다. 또한 DT는 터미널 노드의 크기가 50~150 내외, 가지의 개수가 1,000~100,000개 사이, 불순도의 척도가 Gini Index를 많이 사용했다는 것을 알 수 있었다. 마지막으로 SVM은 C는 0.01~1.3의 내외,  $\gamma$ 은 0.006 ~ 0.012의 값을 보였다. MSE의 지수는 KNN은 평균 0.048%, DT는 평균 0.047%, SVM 평균 0.058%의 성능을 나타냈으며, 본 논문에서 선정한 알고리즘 3개 모두 0.05% 내외의 수치를 갖는 것을 알 수 있다. 또한, Fold의 개수는 대부분 10 또는 20인 경우의 높은 성능을 나타냈었다.

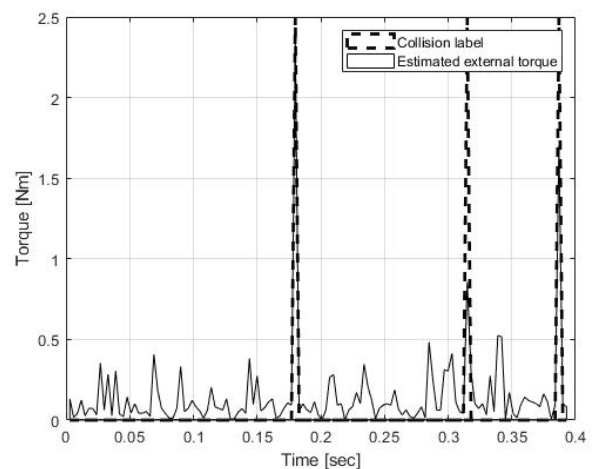
### 4.2 모델 성능 비교

앞선 결과에 따른 각 알고리즘 별 최적의 Hyper-parameter 적용한 모델을 가지고 알고리즘의 실행시간, Recall, Precision, Accuracy를 [Table 6]과 같이 결과를 도출하였다.

최적의 Hyperparameter로 선정된 모델을 동일한 개수의 테스트 데이터를 가지고 예측하였을 경우, 모든 모델의 Recall은 94%, Precision은 96% 내외의 값을 가졌다. 하지만, Accuracy는

[Table 6] Hyperparameter range and types for each model. Note  $y_i$  is the number of observations

Algorithm	Recall	Precision	Accuracy	Time
KNN-10 fold	94.0%	96.2%	96.4%	41 ms
KNN-20 fold	94.7%	95.4%	95.7%	15 ms
DT-10 fold	94.0%	96.1%	95.7%	93 ms
DT-20 fold	93.9%	96.2%	95.0%	5.3 ms
SVM-5 fold	93.3%	96.9%	97.1%	6.5 ms
SVM-10fold	94.0%	96.2%	96.4%	7.1 ms



[Fig. 4] Predicted result of actual data into a model using hyper-parameter

최저 95%에서 최고 97%의 차이를 가진 것을 알 수 있다. 또한, 각각 모델의 평균 예측 시간은 DT, KNN, SVM 순으로 낮다. 또한, 각각 모델의 평균 예측 시간이 KNN는 28.1 ms, DT는 49.3 ms, SVM는 6.8 ms을 가진다. 즉, SVM모델을 적용한 충돌 판별 알고리즘이 높은 정확도와 빠른 시간을 가진다는 것을 알 수 있다. 이를 바탕으로 최적의 SVM 모델을 가지고 실제 매니플레이터의 데이터에 적용하는 경우 [Fig. 4]와 같은 결과를 나타낸다.

## 5. 결 론

본 논문에서는 기계학습 기법 중에 지도학습 알고리즘을 적용하여, 조인트 토크 센서 기반으로 충돌을 판별하기 위한 단계로 1) 데이터 수집 및 전처리 2) 충돌 감지를 위한 분류 모델 개발, 3) 최적의 모델 선정의 3단계를 제안하였다. 제안 사항을 통해 토크 센서의 외부 토크만 추출하여 충돌 상태를 충돌(Collision)과 비충돌(Non-collision)상태로 분류하였으며, 각 알고리즘의 Hyper-parameter를 최적화하여 비교 분석하였다. 그 결과 충돌 판별 알고리즘을 구축하는데 있어서 KNN, DT, SVM 중에서 SVM을 이용하는 것이 높은 정확도와 낮은 실행시간을 가지는 것을 확인할 수 있었다. 다만, 다양한 기계 학습 기법 중의 지도학습 알고리즘만을 적용한 연구라는 점과 충돌 상황 프로세스의 충돌 탐지에서 그쳤다는 점에서 한계가 존재한다. 그러나 학습 기반의 충돌 판별이 높은 정확도로 민감하게 반응했다는 것과 낮은 지연시간으로 인식했다는 것에 의미가 있을 것이며, 로봇 안전 시스템의 기초 연구로 활용 가능하다. 본 연구를 통해 향후, 충돌 감지에서 그치지 않고 충돌 식별, 반응 단계까지 학습한 알고리즘을 개발하고 기술을 적용한 프레임워크 및 로봇 제어 시스템을 개발할 예정이다.

## References

- [1] M. Bortolini, E. Ferrari, M. Gamberi, F. Pilati, and M. Faccio, "Assembly system design in the Industry 4.0 era: a general framework," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5700-5705, Jul., 2017, DOI: 10.1016/j.ifacol.2017.08.1121.
- [2] L. Wang, R. Gao, J. Váncza, J. Krüger, X. V. Wang, S. Makris, and G. Chryssolouris, "Symbiotic human-robot collaborative assembly," *CIRP Annals*, vol. 68, no. 2, pp. 701-726, 2019, DOI: 10.1016/j.cirp.2019.05.002.
- [3] S. Haddadin, A. De Luca, and A. Albu-Schaffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292-1312, Dec., 2017, DOI: 10.1109/tro.2017.2723903.
- [4] S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," *IEEE Access*, vol. 5, pp. 26754-26773, 2017, DOI: 10.1109/access.2017.2773127.
- [5] S. D. Lee, M. C. Kim, and J. B. Song, "Sensorless collision detection for safe human-robot collaboration," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, DOI: 10.1109/iros.2015.7353701.
- [6] P. Aivaliotis, S. Aivaliotis, C. Gkourmelos, K. Kokkalis, G. Michalos, and S. Makris, "Power and force limiting on industrial robots for human-robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 346-360, Oct., 2019, DOI: 10.1016/j.rcim.2019.05.001.
- [7] Y. Tian, Z. Chen, T. Jia, A. Wang, and L. Li, "Sensorless collision detection and contact force estimation for collaborative robots based on torque observer," *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Qingdao, China, 2016, DOI: 10.1109/robio.2016.7866446.
- [8] P. Cao, Y. Gan, and X. Dai, "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1729881419853713, May, 2019, DOI: 10.1177/1729881419853713.
- [9] J. Xiao, Q. Zhang, Y. Hong, G. Wang, and F. Zeng, "Collision detection algorithm for collaborative robots considering joint friction," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, p. 172988141878899, Jul., 2018, DOI: 10.1177/1729881418788992.
- [10] E. Matsas, G. C. Vosniakos, and D. Batras, "Prototyping proactive and adaptive techniques for human-robot collaboration in manufacturing using virtual reality," *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 168-180, Apr., 2018, DOI: 10.1016/j.rcim.2017.09.005.
- [11] A. Mohammed, B. Schmidt, and L. Wang, "Active collision avoidance for human-robot collaboration driven by vision sensors," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 9, pp. 970-980, Dec., 2016, DOI: 10.1080/0951192x.2016.1268269.
- [12] R. C. Luo and C. W. Kuo, "Intelligent Seven-DoF Robot With Dynamic Obstacle Avoidance and 3-D Object Recognition for Industrial Cyber-Physical Systems in Manufacturing Automation," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1102-1113, May, 2016, DOI: 10.1109/jproc.2015.2508598.
- [13] J. Kim, A. Alspach, and K. Yamane, "3D printed soft skin for safe human-robot interaction," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, DOI: 10.1109/iros.2015.7353705.
- [14] Y. Lou, J. Wei, and S. Song, "Design and Optimization of a Joint Torque Sensor for Robot Collision Detection," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6618-6627, Aug., 2019, DOI: 10.1109/jsen.2019.2912810.
- [15] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision Detection for Industrial Collaborative Robots: A Deep Learning Approach," *IEEE Robotics and*

*Automation Letters*, vol. 4, no. 2, pp. 740-746, Apr., 2019, DOI: 10.1109/lra.2019.2893400.

[16] A. De Luca and L. Ferrajoli, "A modified newton-euler method for dynamic computations in robot fault detection and control," *2009 IEEE International Conference on Robotics and Automation*, Apr., 2019, DOI: 10.1109/lra.2019.2893400.

[17] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, Haikou, China, 2007, DOI: 10.1109/fskd.2007.552.

[18] A. P. Pawlovsky, "An ensemble based on distances for a kNN method for heart disease diagnosis," *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, Honolulu, HI, USA, 2018, DOI: 10.23919/elinfocom.2018.8330570.

[19] P. Schratz, J. Muenchow, E. Iturrity, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling*, vol. 406, pp. 109-120, Aug., 2019, DOI: 10.1016/j.ecolmodel.2019.06.002.

[20] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas, "A Survey of Evolutionary Algorithms for Decision-Tree Induction," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 3, pp. 291-312, Aug., 2019, DOI: 10.1016/j.ecolmodel.2019.06.002.

[21] R. G. Mantovani, T. Horvath, R. Cerri, J. Vanschoren, and A. C. P. L. F. de Carvalho, "Hyper-Parameter Tuning of a Decision Tree Induction Algorithm," *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, Recife, Brazil, 2016, DOI: 10.1109/bracis.2016.018.

[22] M. Jaworski, P. Duda, and L. Rutkowski, "New Splitting Criteria for Decision Trees in Stationary Data Streams," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2516-2529, Jun., 2018, DOI: 10.1109/tnnls.2017.2698204.

[23] C. W. Hsu, and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, Mar., 2002, DOI: 10.1109/72.991427.

[24] L. M. He, X. B. Yang, and H. J. Lu, "A Comparison of Support Vector Machines Ensemble for Classification," *2007 International Conference on Machine Learning and Cybernetics*, Hong Kong, China, 2007, DOI: 10.1109/icmlc.2007.4370773.

[25] T. Hastie, R. Tibshirani, and J. Friedman and J. Franklin, "Support Vector Machines and Flexible Discriminants," *The Elements of Statistical Learning*, Springer, 2009, ch. 12, sec. 2, pp 417-421, DOI: 10.1007/978-0-387-84858-7\_12.

[26] E. Duarte and J. Wainer, "Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters," *Pattern Recognition Letters*, vol. 88, pp. 6-11, Mar., 2017, DOI: 10.1016/j.patrec.2017.01.007.

[27] Y. Zhang and Y. Yang, "Cross-validation for selecting a model selection procedure," *Journal of Econometrics*, vol. 187, no. 1, pp. 95-112, Jul., 2015, DOI: 10.1016/j.jeconom.2015.02.006.

[28] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 569-575, Mar., 2010, DOI: 10.1109/tpami.2009.187.

[29] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Bhimavaram, India, 2016, DOI: 10.1109/iacc.2016.25.

[30] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340-341, pp. 250-261, May, 2016, DOI: 10.1016/j.ins.2016.01.033.

[31] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427-437, Jul., 2009, DOI: 10.1016/j.ipm.2009.03.002.

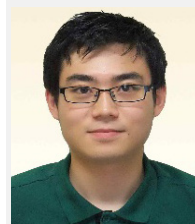


**조 성 현**

2014 한국산업기술대학교 컴퓨터공학부 (공학사)

2020~현재 경북대학교 전자공학부 석사과정 재학 중

관심분야: Robotic Manipulation, Machine Learning



**권 우 경**

2011 포항공과대학교 전자전기공학부(공학사)

2012 포항공과대학교 철강대학원 제어 및 자동화 전공(공학석사)

2017 포항공과대학교 철강대학원 제어 및 자동화 전공(공학박사)

2018~현재 ETRI 연구원

관심분야: Robotic Manipulation, Control systems, and Sampled-data Control