

# 서울 지능형 사물인터넷 해커톤

팀 Mechanics

한양대학교 응용물리학과 이규현

한양대학교 응용물리학과 이민성

한양대학교 응용물리학과 박민혁

2021-05-05 19:08

## 한강 사망 대학생 사건 `미궁` ..CCTV 분석·휴대폰 수색

# 한강 대학생 사망 사건을 아시나요?



## “아들이 잠실대교 부근에서 연락을 끊었다”... 유서도 없이 한강에서 숨진 채 발견

2023-09-07 21:41 이근수 기자

9시간 시차 두고 성인 여자, 남자 극단적 선택  
현장서 유서 발견 안돼...경찰 “범죄 혐의점 없어”

성인 여성과 남성이 9시간 시차를 두고 극단적 선택을 했다.



글로벌이코노믹 기사 프린트하기 프린트하기

## 김기덕 시의원 "한강 교량 투신자살 시도 5년 전보다 2배 증가"

기사입력 2023.02.26 15:06



마포대교

서울시의회 김기덕 의원(더불어민주당)은 지난 24일 서울시에서 받은 ‘최근 5년 간 한강 다리별 투신 현황자료’를 분석한 결과, 5년 전 추진한 2020년 행정사무감사에서 지적한 마포대교의 높은 자살 시도율이 현재까지 최고 수치를 기록하고 있어, 서울시에 대책 마련을 요구했다.

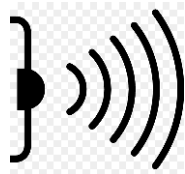
**한강 대학생 사망 사건을 아시나요? 꽃다운 나이 22세에 너무 안타까운 사건이라고 생각합니다.  
이 뿐만 아니라 여러 한강 교량 투신자살 사건이 증가하고 있으며,  
위의 기사를 보시면 아시다시피 서울시에서는 대책을 마련하려고 여러 생각을 하고 있습니다.**

**AIOT를 이용한 한강의 유체 흐름 파악**  
**→ 이상이 있다면 관할 소방서에 연락**

**∴ 자살 예방 가능**

사람이 자살을 하려고 다리 밑에 떨어지게 된다  
면 다리 밑에 설치된 파동  
센서가 순간적인 파동의 변화( $\Delta wave$ )를 감지하  
여 그때부터 n 시간 동안 유체의 흐름을 관찰하고  
기록하며 관할 소방서에 경보가 전송됩니다.  
이때 유체의 흐름에 따라 경보 1~5단계의 사이  
로 전송 됩니다.

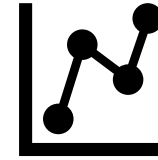
# HOW?



## 센서 네트워크 설치

한강의 유체 흐름을 모니터링하기 위해서는 센서 네트워크를 구축해야 합니다.

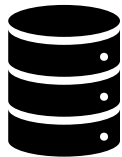
이 센서는 수위, 수온, 유속, 수질 등과 같은 정보를 수집해야 합니다.



## 데이터 분석 및 패턴 인식

수집된 데이터를 분석하여 정상적인 강의 유체 흐름 패턴을 정의하고, 비정상적인

상황을 감지할 수 있는 알고리즘을 개발해야 합니다.



## 데이터 통합과 통신

센서 데이터와 사람 감지 데이터를 통합하고 중앙 서버 또는 클라우드로 전송하여

실시간으로 분석하고 대응할 수 있어야 합니다.



## 알림 및 조치

사람이 강에서 떨어질 경우 즉각적으로 경고를 보내고 구조나 구호 작업을 조치해야

합니다. 이를 위한 통보 시스템 및 대응 프로토콜을 구축해야 합니다.

# 데이터 분석 및 패턴 인식

## 1. 데이터 전처리 및 특성 추출

데이터 전처리는 원시 파동 센서 데이터를 분석 가능한 형태로 변환하는 과정입니다.

아래는 이 과정을 단계별로 나타낸 코드입니다.

```
import numpy as np
import pandas as pd

# 가정: raw_data는 원시 파동 센서 데이터를 포함하는 pandas DataFrame이며,
# 'wave_height'라는 column에 파도 높이 정보가 있음.
raw_data = pd.read_csv('sensor_data.csv')

# 일정 시간 동안의 평균, 최대, 최소 파도 높이 등 다양한 통계치를 계산할 수 있습니다.
data = raw_data.copy()
data['mean_wave_height'] = data['wave_height'].rolling(window=10).mean()
data['max_wave_height'] = data['wave_height'].rolling(window=10).max()
data['min_wave_height'] = data['wave_height'].rolling(window=10).min()

# NaN 값 처리
data.fillna(method='bfill', inplace=True)
```

## 2. 이상 탐지 알고리즘 구현

비정상적인 상황을 감지하기 위해 이상 탐지(Anomaly Detection) 알고리즘이 주로 사용됩니다.

여기에서는 Isolation Forest라는 기법을 사용해보겠습니다.

```
from sklearn.ensemble import IsolationForest

# 학습용 데이터 준비 (일부 시간만 사용하여 정상 상태 학습)
train_data = data[data.index < '2023-06-01']

# Isolation Forest 모델 생성 및 학습
model = IsolationForest(contamination=0.01)
model.fit(train_data[['mean_wave_height', 'max_wave_height', 'min_wave_height']])

# 모든 데이터에 대한 예측 수행 (1: 정상, -1: 비정상)
data['anomaly'] = model.predict(data[['mean_wave_height', 'max_wave_height', 'min_wave_height']])
```

위 코드에서 contamination 매개변수는 비정상적인 데이터의 비율을 가정한 것으로, 이를 조절하면서 최적의 값을 찾아야 합니다.

이제 anomaly column에서 -1 값을 가지는 데이터가 감지된 이상 상황으로 판단할 수 있습니다. 이 경우에는 해당 정보를 소방서에 보고하는 코드를 추가하면 됩니다.

이러한 방식은 단순화된 예시로, 실제 시스템에서는 다양한 요소와 상황을 고려하여 알고리즘을 개선 및 최적화해야 할 수 있습니다. 또한 파동 센서 외에도 여러 다른 종류의 센서(예: 온도, 습도 등)를 함께 사용하여 더욱 정확한 판단을 내릴 수 있습니다.

# 데이터 통합과 통신

**데이터 수집 및 통합:** 센서에서 생성된 데이터를 실시간으로 수집하고, 필요한 경우 다른 데이터 소스(예: 사람 감지 센서)와 통합해야 합니다. 이를 위해 IoT 기반의 솔루션을 사용할 수 있습니다. 예를 들어, MQTT(MQ Telemetry Transport) 프로토콜 등을 사용하여 센서에서 생성된 데이터를 실시간으로 수집하고 중앙 서버나 클라우드로 전송할 수 있습니다.

**데이터 저장 및 처리:** 수집된 데이터는 중앙 서버나 클라우드에 저장되어야 합니다. 여기에는 다양한 DBMS(Database Management System)나 빅데이터 처리 시스템(Hadoop, Spark 등)이 사용될 수 있습니다.

**실시간 분석 및 대응:** 저장된 데이터는 실시간으로 분석되어야 합니다. 스크립트 언어(Python 등) 또는 스트림 처리 엔진(Apache Flink, Apache Storm 등)을 활용하여 실시간 분석을 실행할 수 있습니다. 이상 징후가 탐지되면 즉각적인 알림이나 대응 조치가 이루어져야 하는데, 이 부분은 API 호출, 메일 발송 등의 방식으로 구현될 수 있습니다.

**모니터링 및 관리:** 마지막으로 전체 시스템의 작동 상태와 성능을 모니터링하며 필요한 경우 관리 및 최적화 작업이 진행되어야 합니다.

위 과정들은 크게 보면 IoT(Internet of Things), Big Data, Cloud Computing 기술들이 결합된 형태입니다.



# 알림 및 조치

사람이 강에서 떨어질 경우 즉각적으로 경고를 보내고 구조나 구호 작업을 조치하기 위한 통보 시스템 및 대응 프로토콜을 구축하는 것은 크게 두 부분으로 나누어 볼 수 있습니다.

1. 경고 알림 시스템: 이상 징후가 탐지되면 즉시 관련 당국에 알림을 보내는 시스템이 필요합니다. 이를 위해 API, SMS, 이메일 등 다양한 방법을 사용할 수 있습니다.
2. 구조/구호 대응 프로토콜: 감지된 이상 상황에 따라 적절하게 대응할 수 있는 프로토콜이 필요합니다. 이는 사전에 소방서, 경찰서 등과 협의하여 정해두어야 합니다.

아래는 파이썬에서 Twilio API를 사용하여 SMS 알림을 보내는 예시입니다. 아래 코드에서 `your_account_sid`, `your_auth_token`, `twilio_phone_number`, `'destination_phone_number'` 부분은 실제 Twilio 계정의 SID와 토큰, 그리고 발신자 및 수신자 번호로 변경해야 합니다.

```
from twilio.rest import Client

def send_alert(message):
    # Twilio 계정 정보 설정
    account_sid = 'your_account_sid'
    auth_token = 'your_auth_token'
    client = Client(account_sid, auth_token)

    # SMS 메시지 전송
    message = client.messages.create(
        body=message,
        from_='twilio_phone_number',
        to='destination_phone_number'
    )
```



그리고 아래와 같이 비정상 상황 감지 시 `send_alert` 함수를 호출할 수 있습니다.

```
# 모든 데이터에 대한 예측 수행 (1: 정상, -1: 비정상)
data['anomaly'] = model.predict(data[['mean_wave_height', 'max_wave_height', 'min_wave_height']])

# 비정상 상황 발생 시 알림 전송
if data['anomaly'].iloc[-1] == -1:
    send_alert("비정상 상황 발생! 확인 요망.")
```

다만 실제 운영 환경에서는 세부적인 로직과 에러 처리 등 고려해야 할 요소가 많으므로 위 코드들은 단순한 예시에 불과하다는 점을 유의해야 합니다.