Anomaly Detection in Crowd

Infosys Springboard

May - July 2024

Meet the Team

Pallavi

Ranojay Sikdar

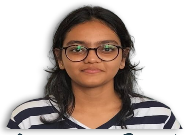
Priyanshu Kumar Saw

Vashishth

Ayush

A.Veera

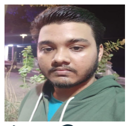
Shreeshta Reddy


Aarchi Shah

Kunjan Kumar Singh

Zubair

K V Sai Srujan


Rudra Banerjee

Sama Mohd Mustufa

V Venkaa Sai Kumar

V ivek Kumar Soni

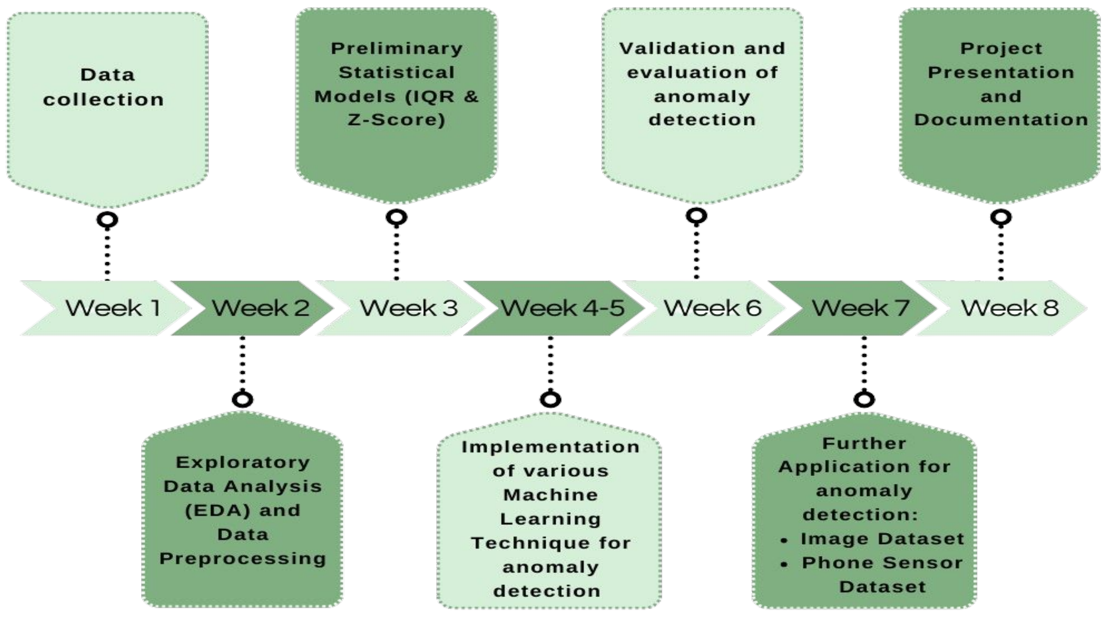
Agenda

- Introduction
- Project Timeline
- Anomaly Detection Dataset Analysis
- Phone Sensor Dataset Analysis
- Image Dataset Analysis
- Future Work

Problem Statement

- **Purpose:**
  - Monitor and analyze crowd behaviors to detect suspicious activities and potential hazards.
- **Data Collection:**
  - Gather smartphone datasets capturing various crowd behaviors.
  - Extract relevant features such as:
    - Crowd density
    - Movement patterns
    - Flow irregularities
- **Analysis:**
  - Apply statistical methods and machine learning algorithms to develop models.
  - Distinguish normal behavior from abnormal patterns.
- **Model Training and Validation:**
  - Train models on labeled datasets.
  - Validate performance using metrics like accuracy, precision, and recall.
- **Implementation:**
  - Deploy models in a real-time system.
  - Provide timely alerts to ensure safety and improve:
    - Event management
    - Urban planning
    - Security monitoring

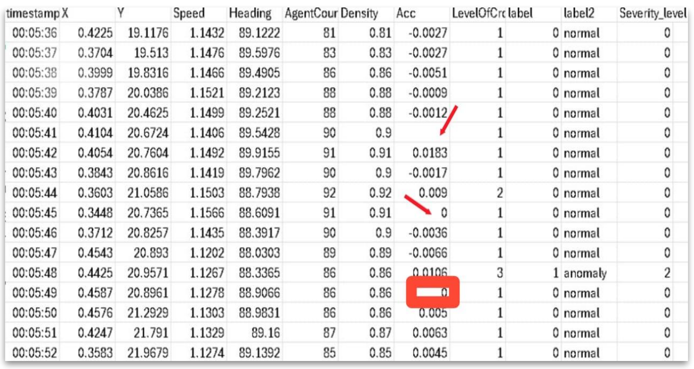
Week-Wise Implementation Plan


Data collection
Preliminary Statistical Models (IQR & Z-Score)
Validation and evaluation of anomaly detection
Project Presentation and Documentation
Week 1
Week 2
Week 3
Week 4-5
Week 6
Week 7
Week 8
Exploratory Data Analysis (EDA) and Data Preprocessing
Implementation of various Machine Learning Technique for anomaly detection
Further Application for anomaly detection:
• Image Dataset
• Phone Sensor Dataset

Anomaly Detection Dataset Analysis

Data Collection & Pre-Processing

- Problem statement defines that smartphone dataset is crucial for analysis and it must contain features to detect activities. Hence the dataset collected from Kaggle.com
- Analysing the dataset shows that there are 104 values are missing for the Acceleration feature. These values should be treated with other values.



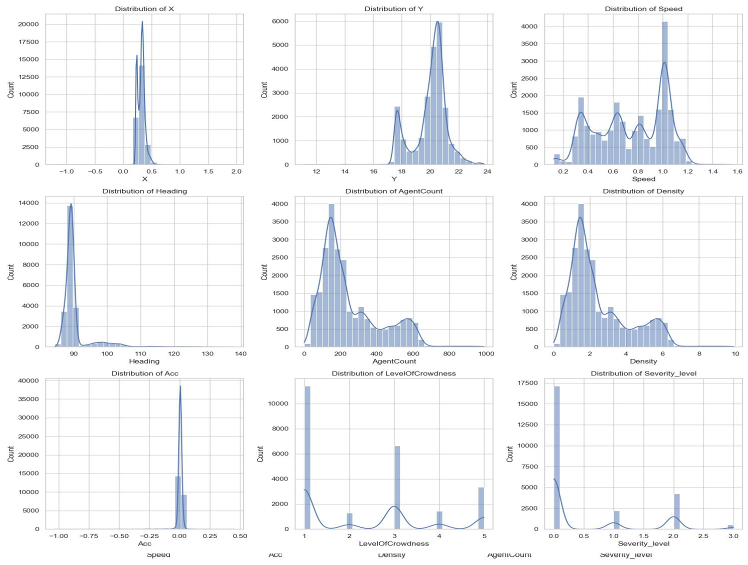
timestamp	X	Y	Speed	Heading	AgentCour	Density	Acc	LevelOfCrc	label	label2	Severity_level
00:05:36	0.4225	19.1176	1.1432	89.1222	81	0.81	-0.0027	1	0	normal	0
00:05:37	0.3704	19.513	1.1476	89.5976	83	0.83	-0.0027	1	0	normal	0
00:05:38	0.3999	19.8316	1.1466	89.4905	86	0.86	-0.0051	1	0	normal	0
00:05:39	0.3787	20.0386	1.1521	89.2123	88	0.88	-0.0009	1	0	normal	0
00:05:40	0.4031	20.4625	1.1499	89.2521	88	0.88	-0.0012	1	0	normal	0
00:05:41	0.4104	20.6724	1.1406	89.5428	90	0.9		1	0	normal	0
00:05:42	0.4054	20.7604	1.1492	89.9155	91	0.91	0.0183	1	0	normal	0
00:05:43	0.3843	20.8616	1.1419	89.7962	90	0.9	-0.0017	1	0	normal	0
00:05:44	0.3603	21.0586	1.1503	88.7938	92	0.92	0.009	2	0	normal	0
00:05:45	0.3448	20.7365	1.1566	88.6091	91	0.91	0	1	0	normal	0
00:05:46	0.3712	20.8257	1.1435	88.3917	90	0.9	-0.0036	1	0	normal	0
00:05:47	0.4543	20.893	1.1202	88.0303	89	0.89	-0.0066	1	0	normal	0
00:05:48	0.4425	20.9571	1.1267	88.3365	86	0.86	0.0106	3	1	anomaly	2
00:05:49	0.4587	20.8961	1.1278	88.9066	86	0.86	0	1	0	normal	0
00:05:50	0.4576	21.2929	1.1303	88.9831	86	0.86	0.005	1	0	normal	0
00:05:51	0.4247	21.791	1.1329	89.16	87	0.87	0.0063	1	0	normal	0
00:05:52	0.3583	21.9679	1.1274	89.1392	85	0.85	0.0045	1	0	normal	0

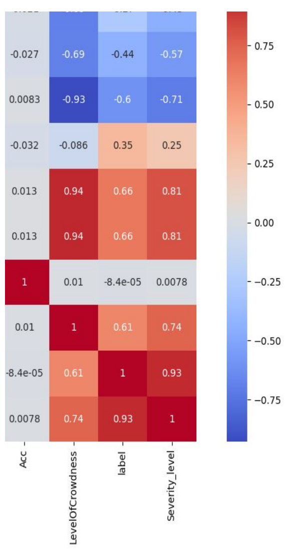
- Missing values treated by calculating mean value for distribution of acceleration feature and then all null values (missing values) replaced with mean value.
- All numerical features are converted into same format i.e int data type.

Exploratory Data Analysis

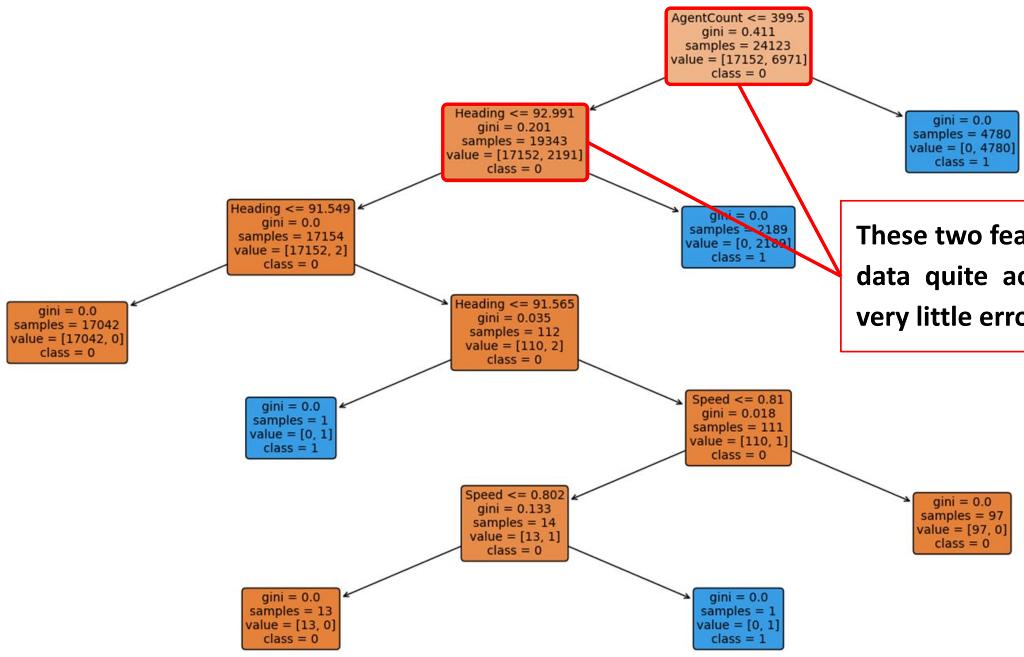
Correlation Heatmap Distribution Plot

- 'Speed' and '
  correlation.
- 'AgentCount'
  positive correl





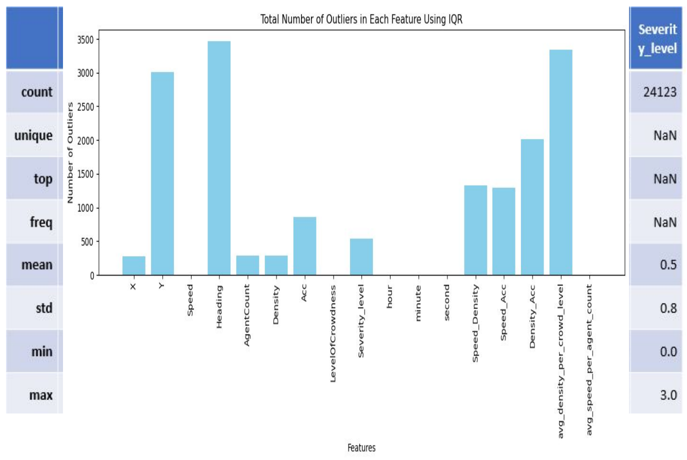
Plotting a Decision Tree



These two features split the data quite accurately with very little error.

Initial Statistical Analysis

- Initial statistical analysis for anomaly detection in smartphone datasets involves examining the data for any irregularities or outliers that may indicate unusual behavior or errors.
- Calculate **Z-scores** to identify data points that deviate significantly from the mean.
- Compute the **IQR** to detect outliers by identifying data points that fall below the first quartile (Q1) or above the third quartile (Q3) by 1.5 times the IQR.



	Severity_level
count	24123
unique	NaN
top	NaN
freq	NaN
mean	0.5
std	0.8
min	0.0
max	3.0

Model Building And Evaluation

- **Model Training :**
  - The model learns patterns from a labeled dataset, identifying normal and abnormal behaviors.
  - It adjusts parameters to minimize prediction errors and better detect anomalies.
- **Model Testing:**
  - The model's performance is evaluated using unseen data.
  - Predictions are compared with actual outcomes to assess accuracy.
- **Evaluating the model results:**
  - Analyze the confusion matrix to understand the distribution of predictions..
  - High accuracy indicates the model's good generalization ability.
  - For imbalanced datasets, consider additional metrics like precision, F1 Score and recall.
- **Hyperparameter Tuning:**
  - Fine-tuning the model's parameters to increase accuracy and improve detection of anomalies.

Model Description

We built a suite of models and evaluated each of them to finalize the model that gives the best results for our dataset

- **KNN:**
  - KNN for anomaly detection identifies outliers by calculating the distance of each data point to its 'k' nearest neighbors.
- **Isolation Forest:**
  - Effective in detecting anomalies in high-dimensional datasets.
- **K-Means:**
  - K-Means is a clustering algorithm used to group similar data points into clusters based on their features. For anomaly detection, the idea is to identify data points that do not fit well into any of the clusters.
- **Decision Tree:**
  - In the context of anomaly detection, decision trees can be used to classify data points as normal or anomalous.
- **CNN:**
  - This model it directly learns from data, and are excellent in dealing with sensor data. They detect spatial relationships between different sensors.
- **Gradient Boosting:**
  - This model uses sequential decision trees to accurately detect anomalies by focusing on correcting prediction errors.

Model Comparison

Based on the different evaluation methods, we conclude that KNN, Decision Trees, and CNN give us the best results with high accuracies, precision , etc….

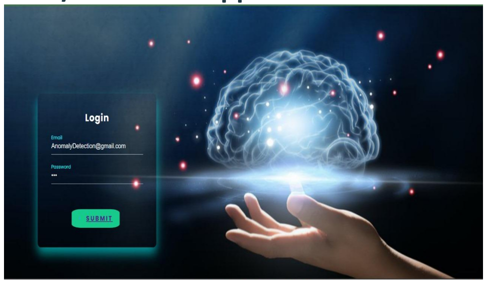
Models	Accuracy	Precision	Recall	F1 Score
KNN	0.99	0.99	0.99	0.99
Isolation Forest	0.91	0.82	0.87	0.85
K_Means	0.90	0.98	0.70	0.80
Decision Tree	0.99	1.00	0.99	0.99
CNN	0.99	0.99	0.99	0.99
Gradient Boosting	0.99	0.98	0.97	0.97

Phone Sensor Dataset Analysis

Model Application on Phone Sensors

- **Source:** Kaggle
- **Collection Context:**
  - Recorded while driving in city traffic.
- **Parameters Recorded:**
  - **Location:** Longitude, Latitude
  - **Movement:** Speed, Distance, Time
  - **Sensors:**
    - **Accelerometer (Acc):** Acc X, Acc Y, Acc Z
    - **Gyroscope (Gyro):** Gyro X, Gyro Y, Gyro Z
  - **Direction:** Heading
- **Labels:**
  - **0:** Normal driving
  - **1:** Aggressive driving
- **Usage:** Train ML models to classify driving behavior (normal vs. aggressive).
- **Key Features:** Focus on accelerometer and gyroscope readings.

UI/UX of the Application Created:



Phone Sensor Anomaly Model Comparison

Based on the different evaluation methods, we conclude that KNN, Decision Trees, and DBSCAN give us the best results with high accuracies, precision .

Models	Accuracy	Precision	Recall	F1 score
KNN	79.29%	83.41%	81.24%	82.31%
Isolation Forest	77.26%	86.53%	73.01%	79.204%
SVM	76.24%	85.82%	70.89%	77.64%
Decision Tree	79.82%	83.36%	82.42%	82.89%
DBSCAN	80.38%	84.97%	81.30%	83.09%

Image Dataset Analysis

Anomaly Model Impact on UCSD Data

- **Dataset Overview**
  - Two subsets (Peds1: 34 training, 36 testing; Peds2: 16 training, 12 testing).
  - Ground truth: frame-level flags and pixel-level masks for some clips.
- **Types of Anomalies**
  - Non-pedestrian entities (bikers, skaters, carts, wheelchairs,bicycles).
  - Anomalous pedestrian movements
- **Data Characteristics**
  - Natural anomalies in 200-frame clips.
  - Detailed evaluation with pixel-level annotations.

★ Dataset was taken from the kaggle.com

Image Anomaly Model Comparison

Model	Accuracy	Description
Convolutional Neural Network (CNN)	0.92	Achieved the highest accuracy. Effective for image data due to convolutional layers learning spatial hierarchies.
Spatiotemporal Convolutional Autoencoder(STACE)	0.67	It is a type of neural network designed to learn efficient representations of spatiotemporal data, which involves both spatial and temporal dimensions.
Autoencoder	0.60	Unsupervised learning model. Effective in reducing dimensionality and capturing data patterns for anomaly detection.
Artificial Neural Network (ANN)	0.57	General-purpose neural network. Less effective for image data compared to CNN.
Multi-Layer Perceptron (MLP)	0.57	Similar to ANN, with more layers. Struggled with image data.
You Only Look Once(YOLO)	0.52	YOLO is a real-time object detection system that processes images with a single neural network, making it both fast and accurate for various applications.

Future Scope

- **Real-Time Anomaly Alerts:**
  In real life, we need to have real-time updates on anomaly detection among crowds. Therefore, we need to apply these machine learning techniques to real-time videos. This can be done within applications or through APIs.

- **API Development:**
  Create an API to integrate the anomaly detection model with various applications, making it accessible for real-time use in industries like security surveillance and quality control.

- **Advanced Architectures:**
  Explore the use of more sophisticated neural network architectures, such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs), for improved anomaly detection capabilities.

- **Cross-Domain Applications:**
  Adapt and apply the CAE model to other anomaly detection tasks in fields like medical imaging, industrial automation, and financial fraud detection.

Thank You

Appendix

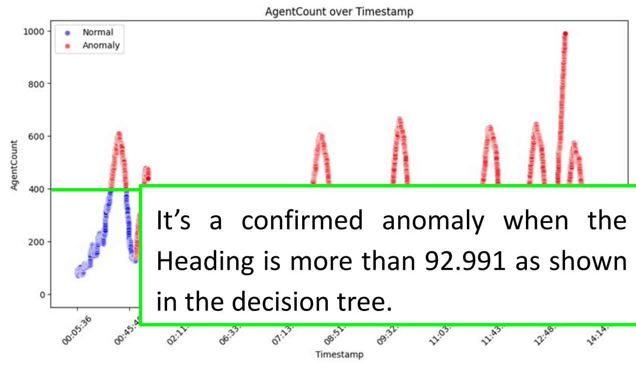
Initial findings in the dataset

- **Data Inspection:**
  - Thoroughly examined the timestamp column within the dataset.
  - Identified irregularities such as gaps and repeated timestamps.
- **Hypothesis Formulation:**
  - Developed two potential explanations based on the observed inconsistencies:
    - Single device with multiple recording sessions.
    - Multiple devices with concurrent recordings.
- **Further Investigation:**
  - Initiated in-depth analysis to validate the proposed hypotheses.

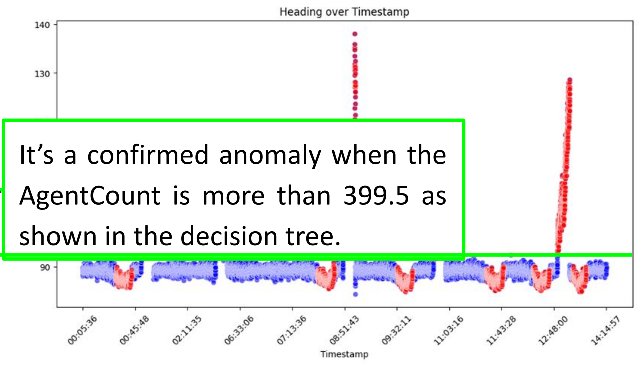
Case 1: Same device but different hours

- **Data Loading and Preparation:**
  - Loaded the raw crowd data into a suitable data structure.
  - Converted timestamps to a numerical format (seconds) for efficient comparison.
- **Hourly Segmentation:**
  - Identified the start of each hour based on timestamp changes.
  - Divided the data into hourly segments.
- **Gap Detection and Handling:**
  - Checked for gaps or inconsistencies within each hourly segment.
  - Separated segments with gaps for further processing.
- **Data Consolidation:**
  - Combined the processed segments into a unified dataset.
- **Data Output:**
  - Saved the cleaned and processed data into a new CSV file for subsequent analysis.

Plotting some graphs using the CSV we got



It's a confirmed anomaly when the Heading is more than 92.991 as shown in the decision tree.



It's a confirmed anomaly when the AgentCount is more than 399.5 as shown in the decision tree.

Case 2: Same Hour but Different Devices

- **Data Loading and Preparation:**
  - Loaded timestamped data into a DataFrame.
  - Converted timestamps to seconds for efficient analysis.
- **Gap Detection and Segmentation:**
  - Identified breaks or gaps in the timestamp sequence.
  - Segmented data into separate CSV files based on these gaps.
- **Data Analysis:**
  - Determined that approximately 18 devices contributed data per hour.
  - Observed varying timestamp ranges across devices.
- **Data Visualization:**
  - Created individual plots for each device's data.
  - Combined plots into a video to visualize device movement over time.

Video Generated from the Data