LIVESTOCK REMOTE MONITORING USING MACHINE LEARNING CLASSIFICATION TECHNIQUE

By, *Ramesh Dinesh Pai.*

(201104047)

DEFINING THE PROBLEM

Traditionally, farmers face challenges in closely monitoring individual livestock, leading to difficulties in early health detection and assessing overall animal welfare. To address this, an increasing number of sensors are being deployed to collect data on livestock behaviour. This project aims to leverage machine learning technique for the accurate identification of multiple unitary behaviours in livestock based on data collected from these sensors. Later the model will be deployed in a web application for remote monitoring, enabling early detection of improper behaviour and alerting farmers for intervention.

Project Overview:

Phase 1: Development of Machine Learning Model for Predicting Normal Behaviours.

* The initial phase focuses on creating a machine learning model capable of predicting normal behaviours exhibited by livestock.
* Leveraging data collected from livestock sensors, the model will be trained to recognize various target behaviours such as sitting, standing, walking, and grazing.
* Data pre-processing techniques will be employed to enhance the model's predictive accuracy.

Phase 2: Model Deployment for Remote Monitoring

* The developed model will be deployed in a web application used to establish a baseline for normal behaviours, against which real-time data from sensors will be compared.

Objectives:

* Developing a machine learning model capable of accurately predicting various normal behaviours in livestock.
* Deploying the machine learning model in the web application for remote monitoring purpose.

Significance:

* Improved animal welfare through remote monitoring and early detection of health issues.
* Enhanced efficiency in livestock management, allowing farmers to respond quickly to abnormal behaviour.

DATA COLLECTION

Sensor:

A livestock sensor (contains accelerometer and other sensors) was used for the purpose of collecting data. Sensor was set to lower sampling frequency (1 Hz) for extending battery life while still capturing essential information. Higher sampling frequencies consume more power.

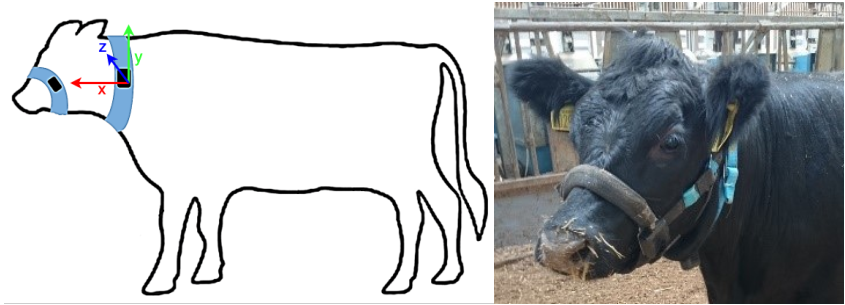
Sensor Component to be used for Data Collection:

Accelerometer:

* Measures acceleration along multiple axes (e.g., x, y, z).
* Captures variations in movement and posture of the livestock.
* Recorded as a time series with values for each axis (x, y, z) at a specified sampling frequency (e.g., 1 Hz)

Sensor Placement:

Choosing where to put livestock sensor is crucial because it affects the quality of the information we get. Each option has its pros and cons, and the decision depends on the project's goals. The neck is stable and good for capturing key movements. The tail is useful for emotional cues, but its movement can make readings less steady. The leg is good for specific movements but may not capture overall actions well. Deciding where to place the sensor should match the project's main goals. If it's about head or body movements, the stable neck is a good choice. In this project, for capturing various livestock behaviours, the neck provides stability and a good view, making it a reasonable pick.



Onsite Data Collection:

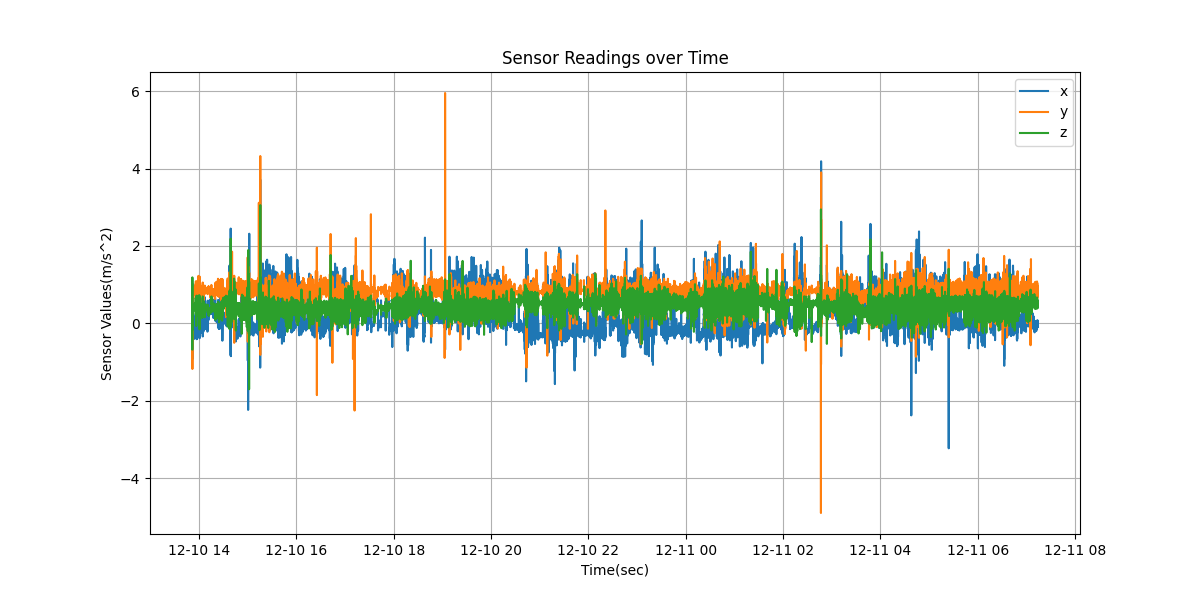
During the onsite data collection, the livestock sensor was attached to the neck of livestock to capture its behaviours. Due to the sensor's limited battery life, the data was captured for each individual behaviour (standing, sitting, walking and grazing) one at a time, charging the sensor between sessions. This ensured patterns for each behaviour was collected effectively, resulting in dataset of high quality.

Sample Dataset:

For now, since the livestock sensor was not developed, an online sample dataset was used for the project. The sample dataset was collected from an accelerometer with sampling rate already labelled with behavioural labels (standing, sitting, walking and grazing). Characteristics of dataset,

* Total time duration: 17.37 hours
* Number of Columns: 8
* Number of features: 3 (x, y, z)
* Number of labels: 4
* Target behavioural labels:
* Number of rows: 62540
* Number of readings for each behaviour:
  + Sitting: 11810
  + Standing: 44870
  + Walking: 18130
  + Grazing: 25490
* Number of readings falling in the intersection of behaviours: 28680

Data Visualization of Sensor Data (m/s^2) v/s Time (sec):



**Dataset Format:**



Here,

**Features:** x, y, z (acceleration value in m/s^2 of accelerometer in x, y, z direction)

**Target Behavioural Labels:** Sitting, Standing, Walking and Grazing.

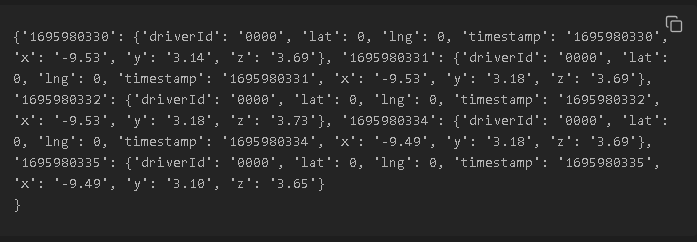
**DATA PRE-PROCESSING**

Data pre-processing is crucial to enhance the quality of raw data by addressing inconsistencies, missing values, and outliers. It ensures data reliability, improves model performance, and facilitates accurate analysis by creating a clean, standardized dataset for machine learning algorithms to yield meaningful insights.

**A] Data Conversion:**

The livestock sensor data is stored real time in fire base database and downloaded as a CSV file. The data that is received is in a nested dictionary format, similar to JSON. Each timestamp is a key, and its value is another dictionary with attributes like *'driverId'*, *'lat'*, *'lng'*, *'timestamp'*, *'x'*, *'y'*, and *'z'*.

**Raw Data Format:**



To process this data, we have to convert it into a format that is more suitable for the specific use case. In the context of livestock behaviour prediction, there is a need to extract relevant features or reformat the data to suit the input requirements of machine learning model.

**Program:**

****

**Output:**

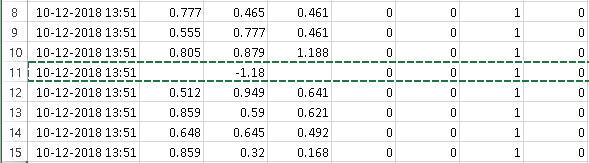
|  |  |  |  |
| --- | --- | --- | --- |
| **time\_stamp** | **x** | **y** | **z** |
| 10-12-2018 13:51 | 0.438 | 0.844 | 0.309 |
| 10-12-2018 13:51 | -0.184 | -0.82 | -0.68 |
| 10-12-2018 13:51 | 0.652 | 0.711 | 0.148 |
| 10-12-2018 13:51 | 1.031 | 0.703 | 0.395 |
| 10-12-2018 13:51 | 0.711 | 0.48 | 0.473 |
| 10-12-2018 13:51 | 0.641 | 0.547 | 0.504 |
| 10-12-2018 13:51 | 0.777 | 0.465 | 0.461 |

**B] Data Cleaning:**

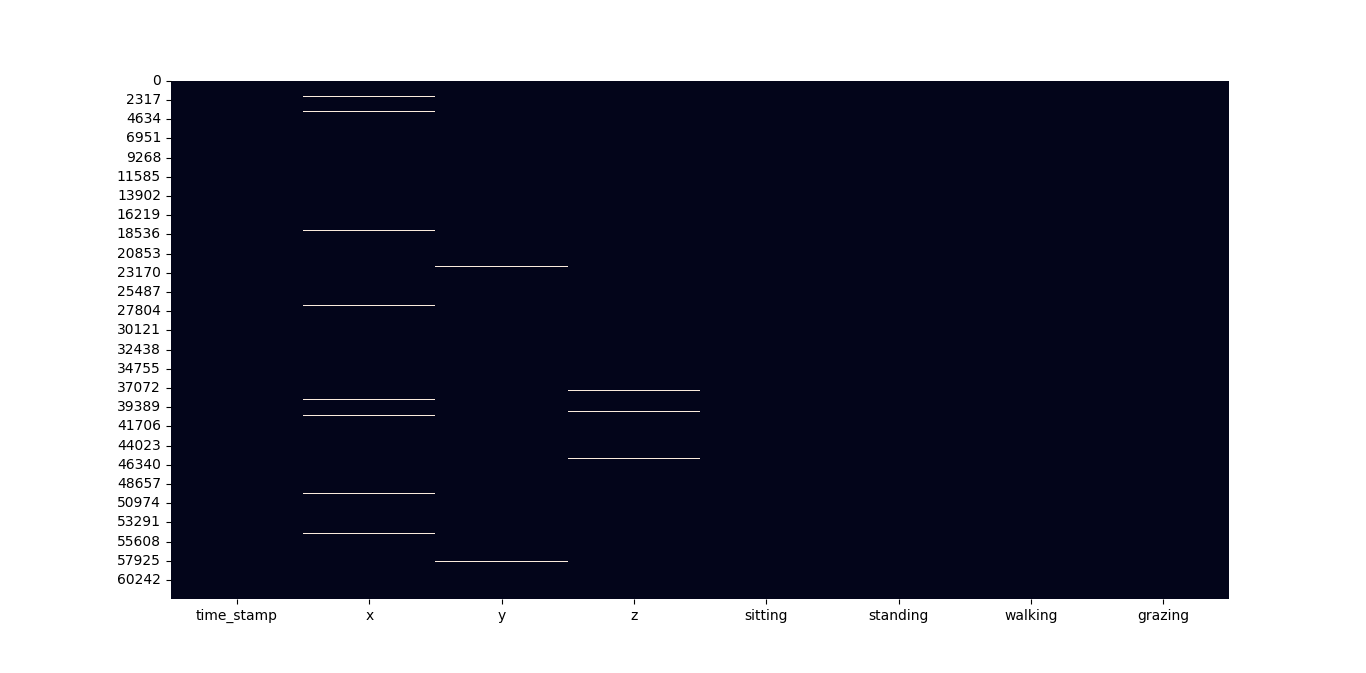
**Finding and imputing missing values:**

Missing values are values that are not present in a dataset for certain observations or features. These missing values can occur for various reasons, such as data collection errors, sensor malfunctions, or simply because some information was not recorded. Dealing with missing values is an essential step in the data pre-processing phase, and it can significantly impact the performance of machine learning models.

Example:



Heat-map representation is used to visualize the missing values in the dataset,



* Each row corresponds to a row in the dataset.
* Each column corresponds to a column in the dataset.
* The cells are color-coded to represent whether the corresponding value in the dataset is missing (usually white or a different colour) or not missing (usually a distinct colour)

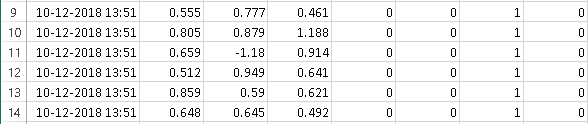
Total missing values:



Imputing missing values:

Using the combination of forward fill and backward fill, calculating their average and imputing the point,

Example:



Program:



**Removing Outliers:**