

700104

Edge AI Coursework

James McMillan
202202995



What is an Edge AI system?

This assignment aims to create an edge device that can run simultaneously two artificial inelegance models.

For a device to be on the edge all of the computation must be performed locally on the devices hardware.

Within this assignment my edge device is a raspberry pi with a pi camera.



Assignment components

1

Monitoring

The application should be able to monitor and record key information about the systems components, specifically the CPU.

2

Detection

The application utilizes the live video stream to detect people and classify them into an area. (Left, right or center)

3

Prediction

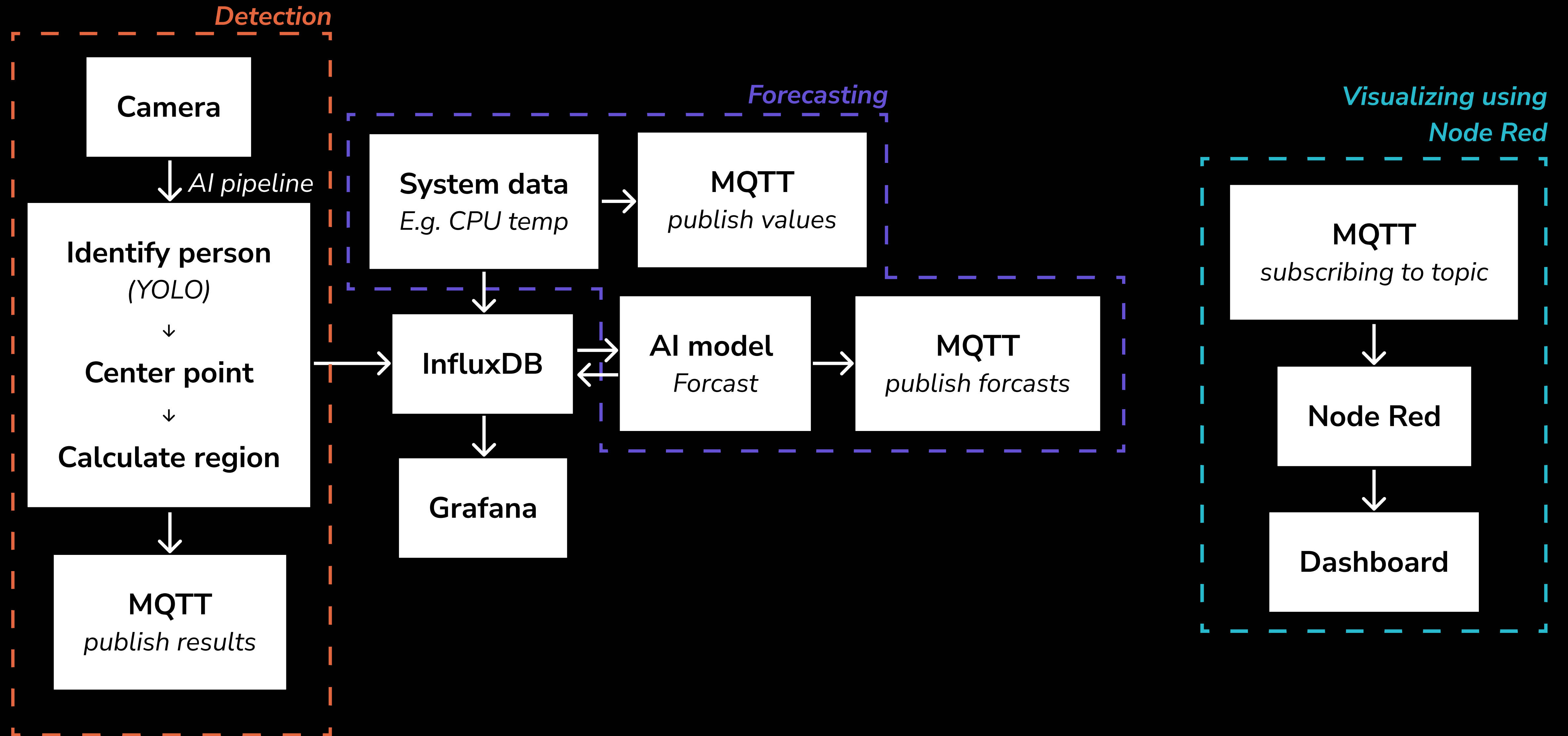
The application should be able to forecast the temperature of the devices CPU 30 minutes into the future.

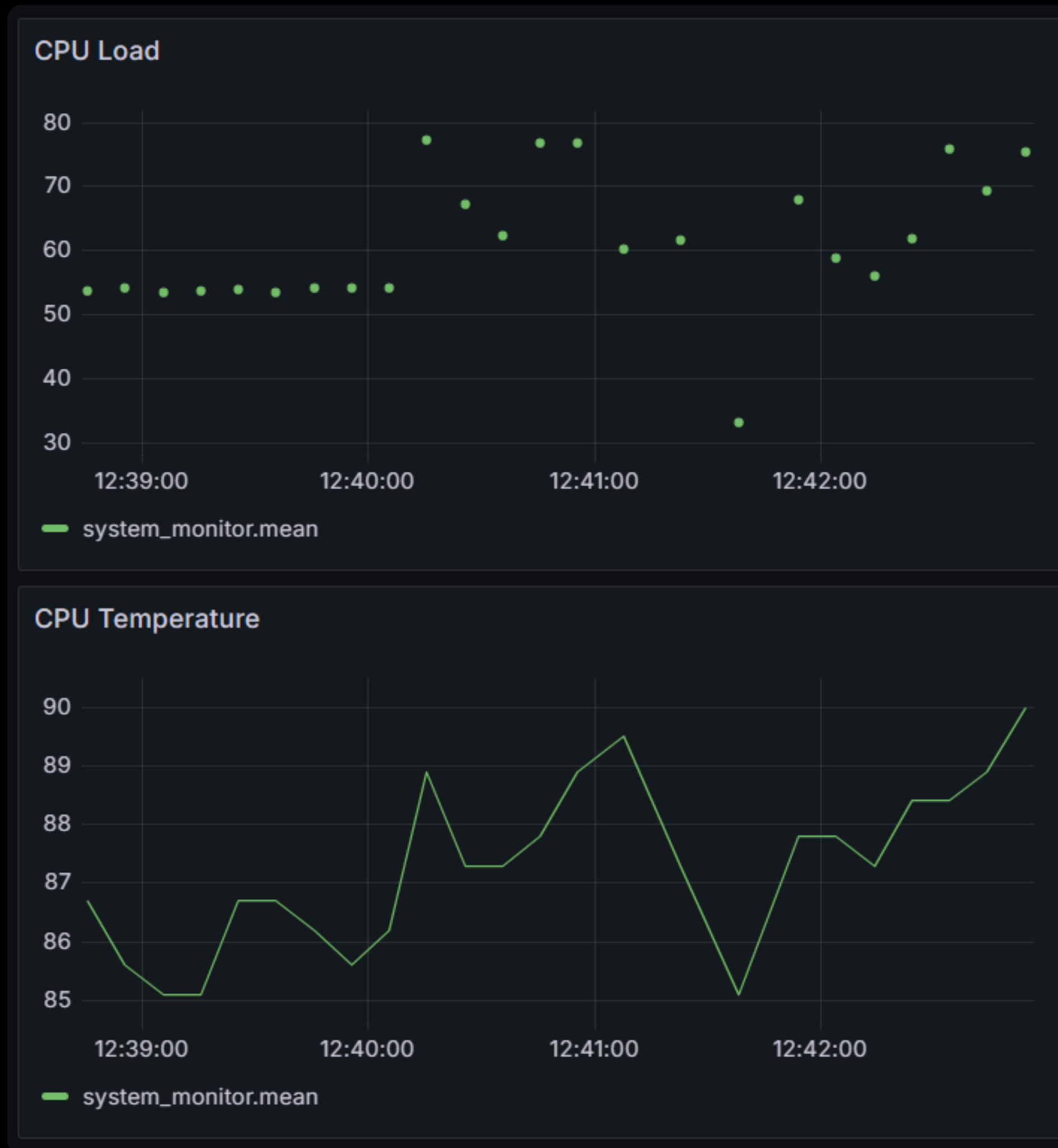
4

Ethics

When developing the application a strong focus should be placed on ethical considerations.

System architecture





CPU

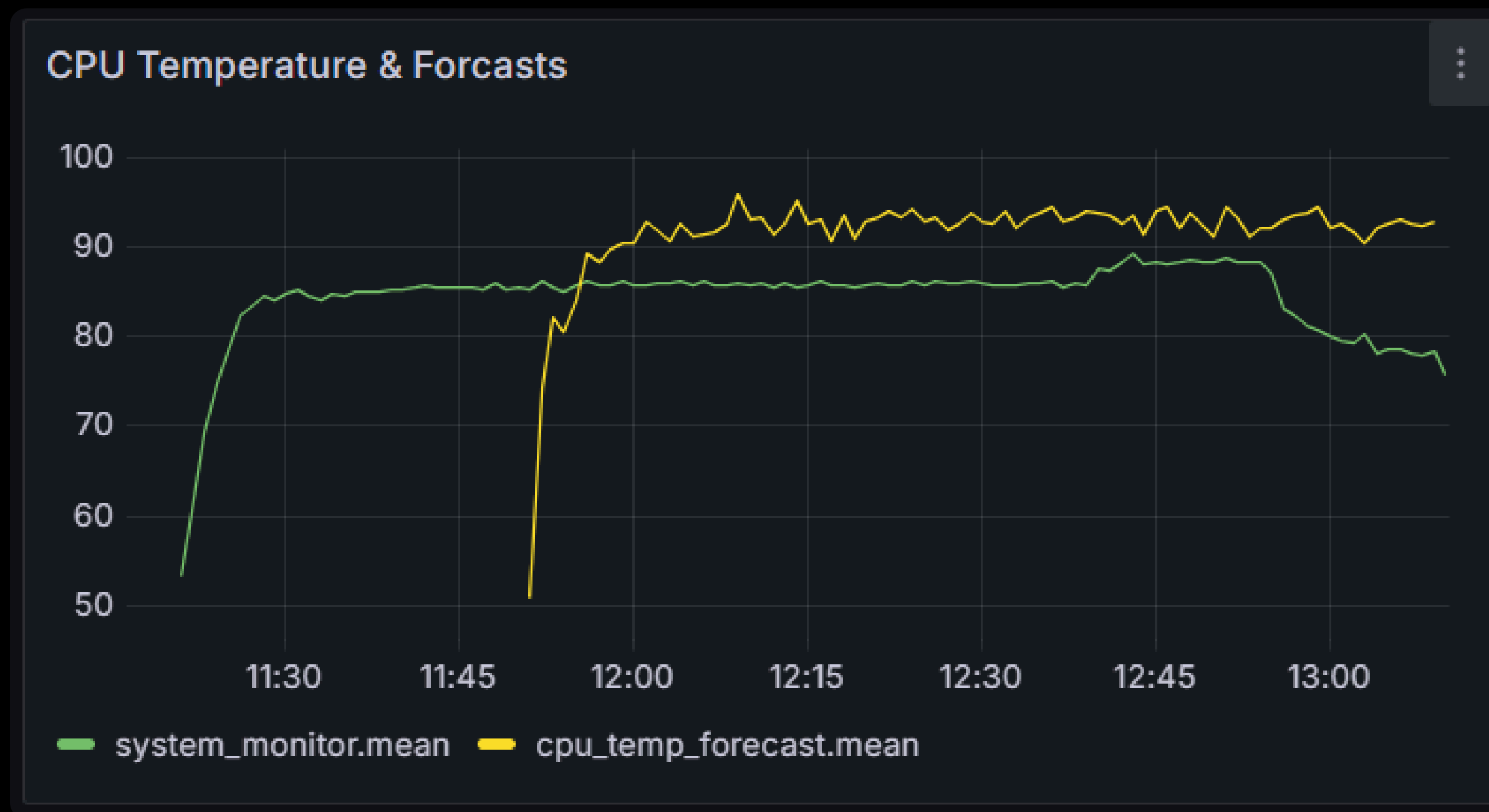
Monitoring

The application utilizes a service to get information about the CPU before these being written to Influx DB.

- CPU Usage
- CPU Load
- CPU temperature

Once stored in Influx DB these can then be plotted using Grafana to show the overall trends, allowing the end user to proactively make changes.





CPU TEMPERATURE

Prediction

Utilizing the information stored as part of the monitoring service the application should be able to predict the CPU temperature 30 minutes into the future.

The results of which should the be stored in Influx DB and visualized in Grafana.



Python3 generatemodel.py

MAKING PREDICTION MODEL

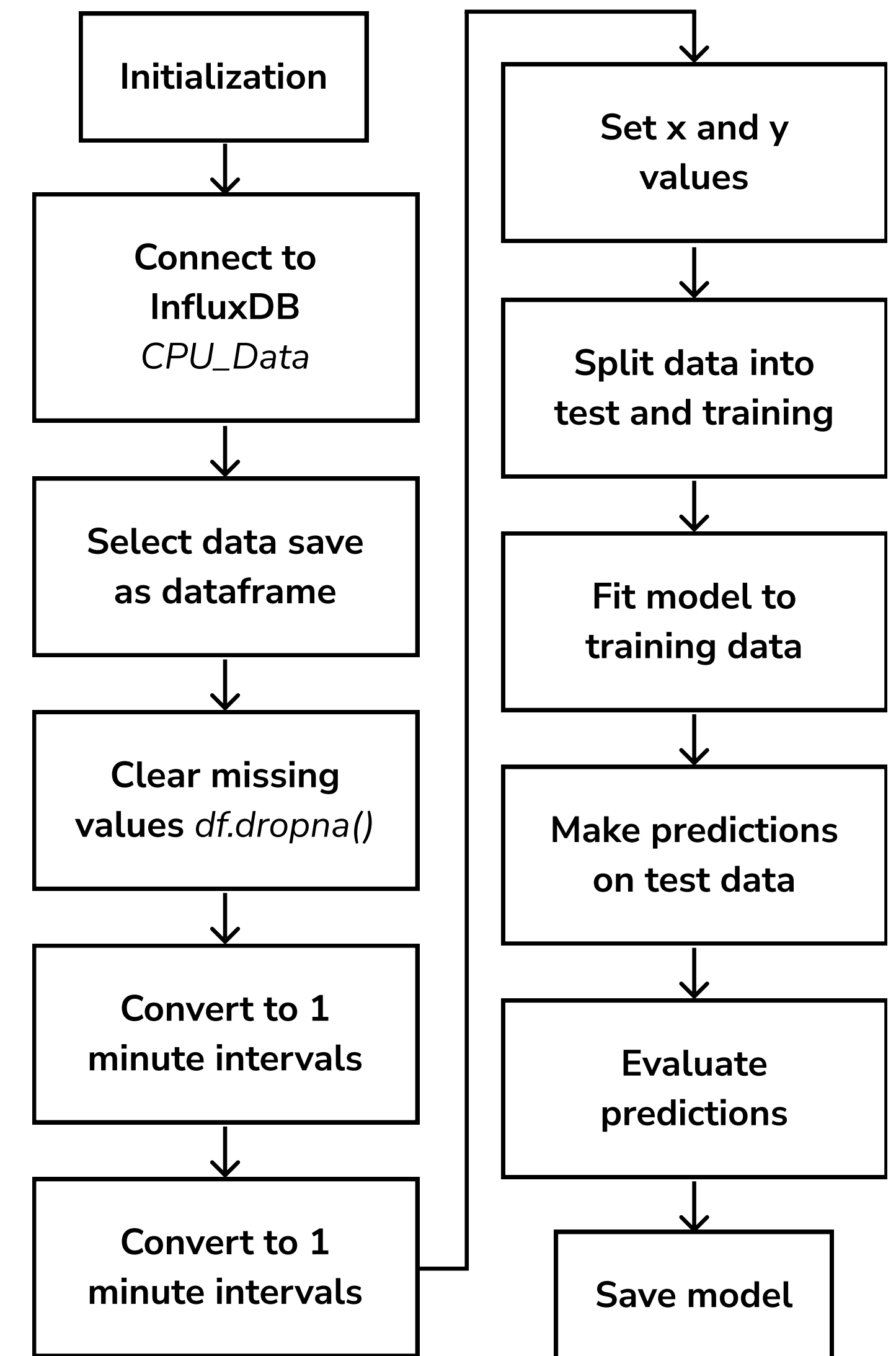
Ridge regression

A type of linear regression, this means that the model aims to fit a straight line to the data. Tho with ridge regression the model penalizes when it relies upon a single feature too much. This helps to reduce overfitting.

Model evaluation

Mean absolute error: 3.576

Root mean squared error: 3.714

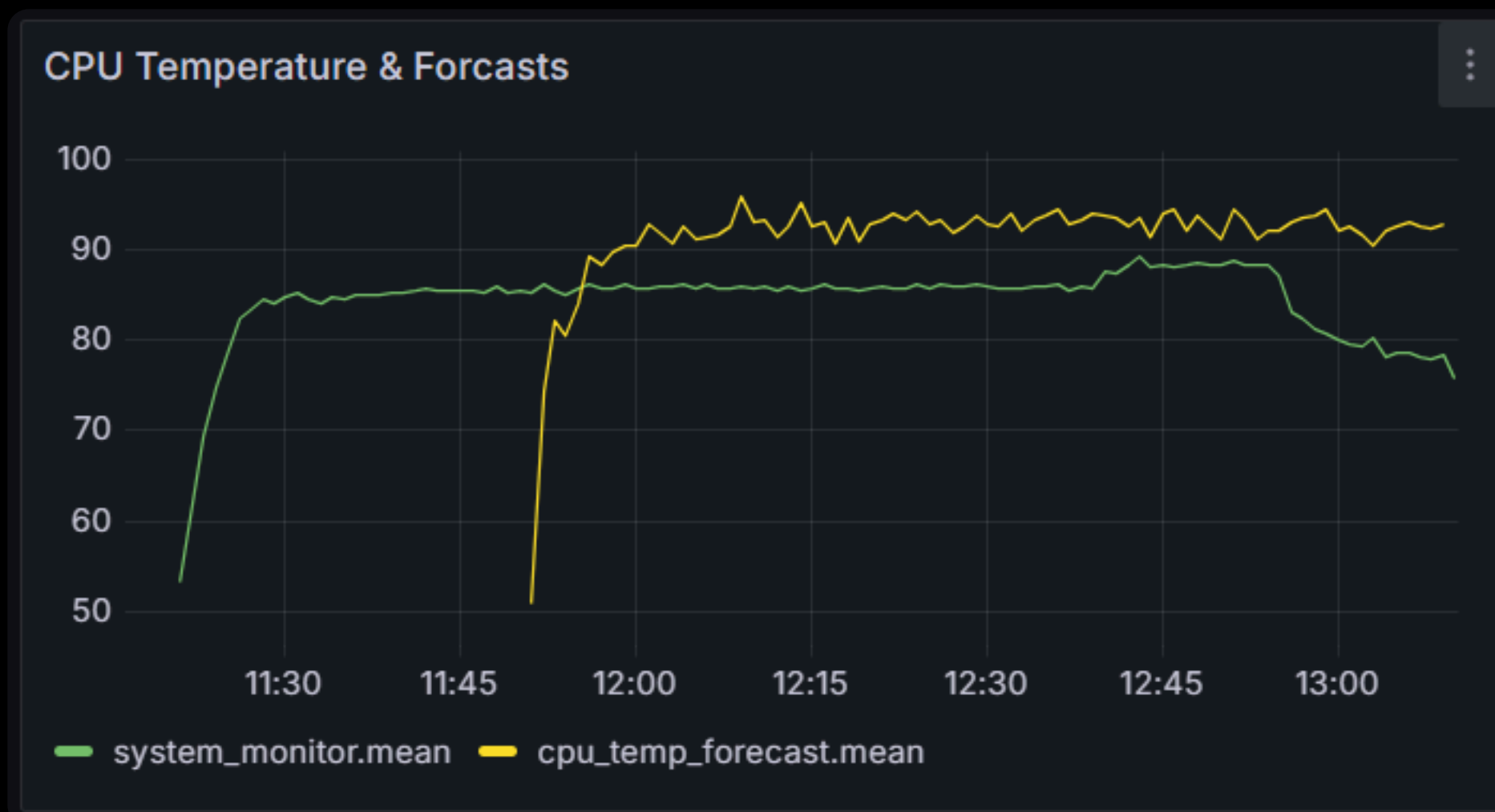


Python3 makeforecast.py

MAKING PREDICTION MODEL

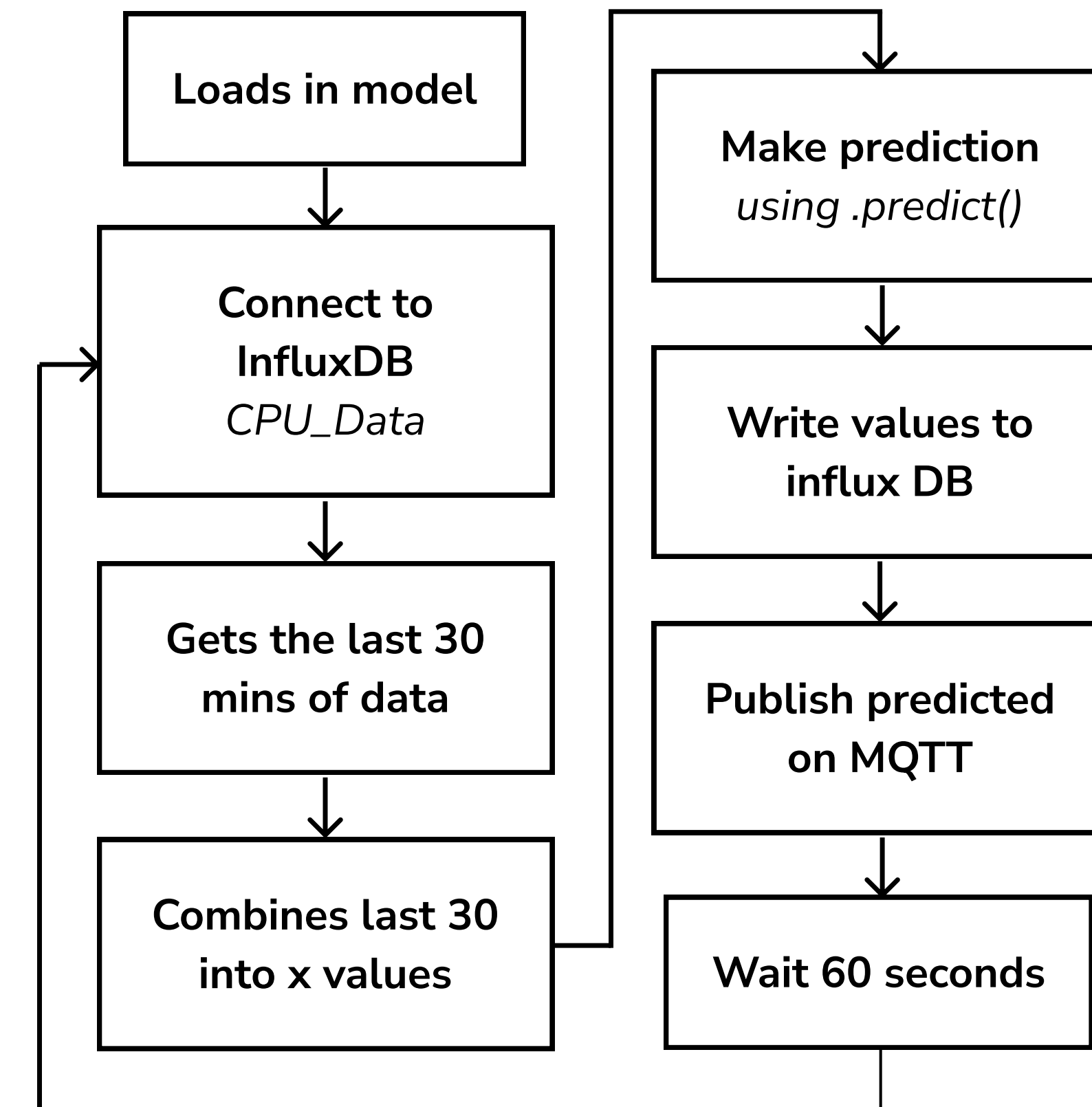
Forecasting

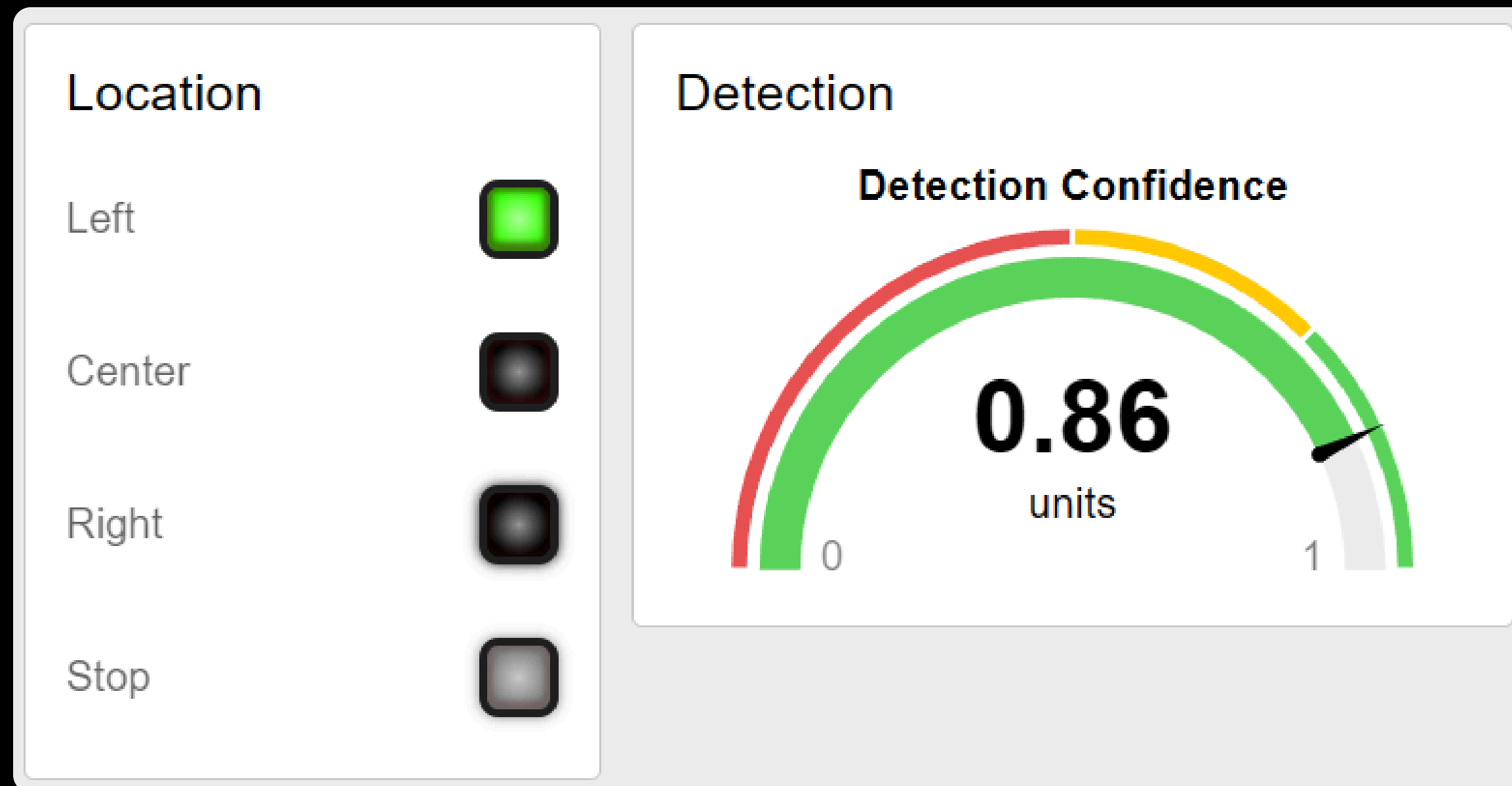
Using the ridge regression model generated trained previously the application should be able to predict the CPU temperature for the device in 30 minutes time. Once the model has made its predictions these predictions should be written to Influx DB where they can then be plotted using Grafana.



Key uses

- System monitoring
- Early warning if the system is about to overheat
- Allows for a proactive response such as turning a fan on.





PERSON DETECTION

Detection

Utilizing the camera's live stream the application should be able to detect a person within the video.

Following this they should be classified based on where they are within the frame.

- Left
- Right
- Center



Python3 detect.py

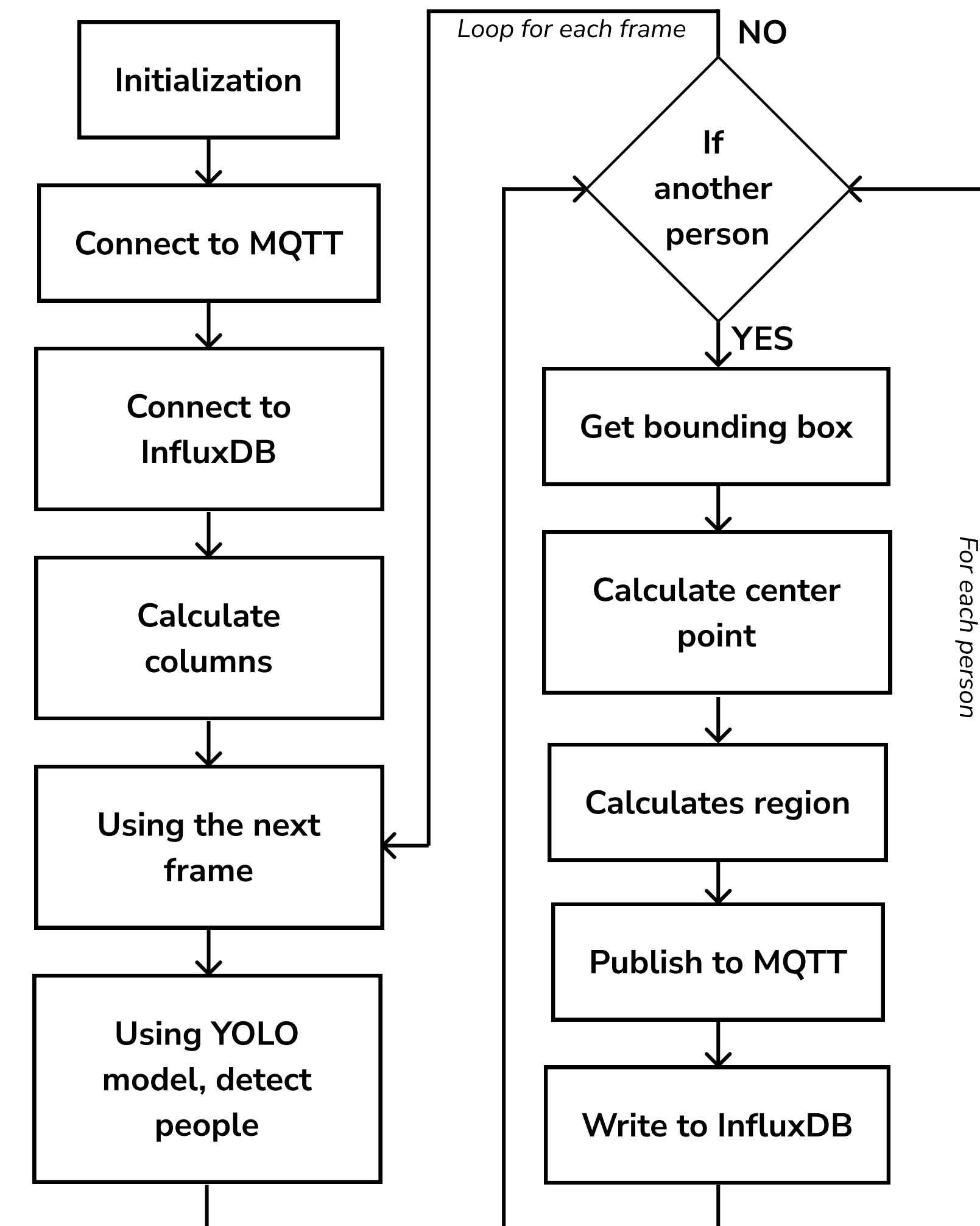
PERSON

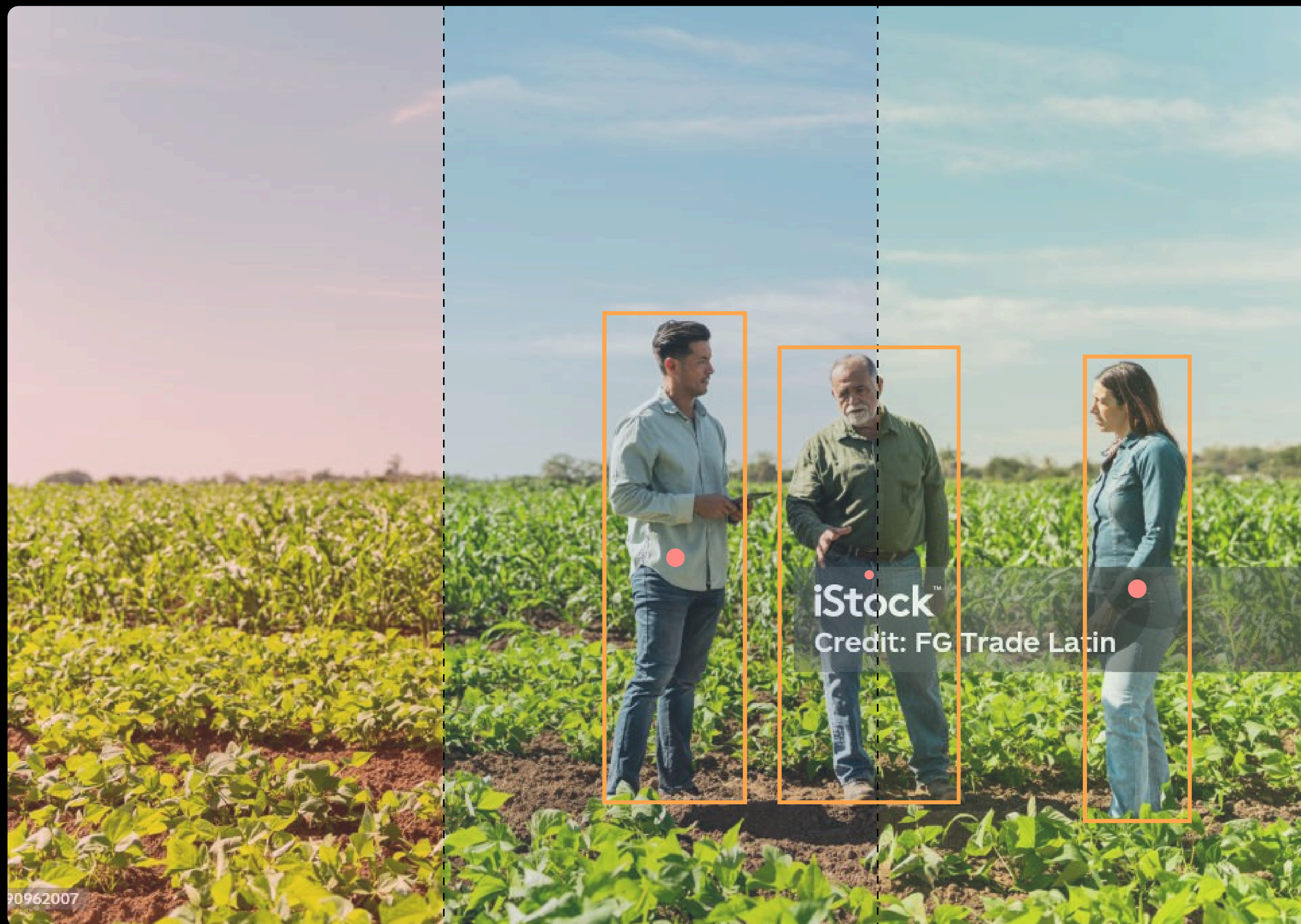
Detection

The application should be able to detect and classify a person into a region based on where they are within the frame.

For detecting people i have used the pre trained YOLO model, the model already has the ability to detect people and provide bounding box data. Pre trained on the COCO dataset.

Once the application has each person's bounding box coordinates it calculates the center of this and using this information determines which region the person is within.





Gets the frames size

Splits it into three

**Identifies the people
and gets bounding
box**

**Calculates center
point**

**Determines which
region they are
within**

**Saves to Influx DB
and publishes on
MQTT**

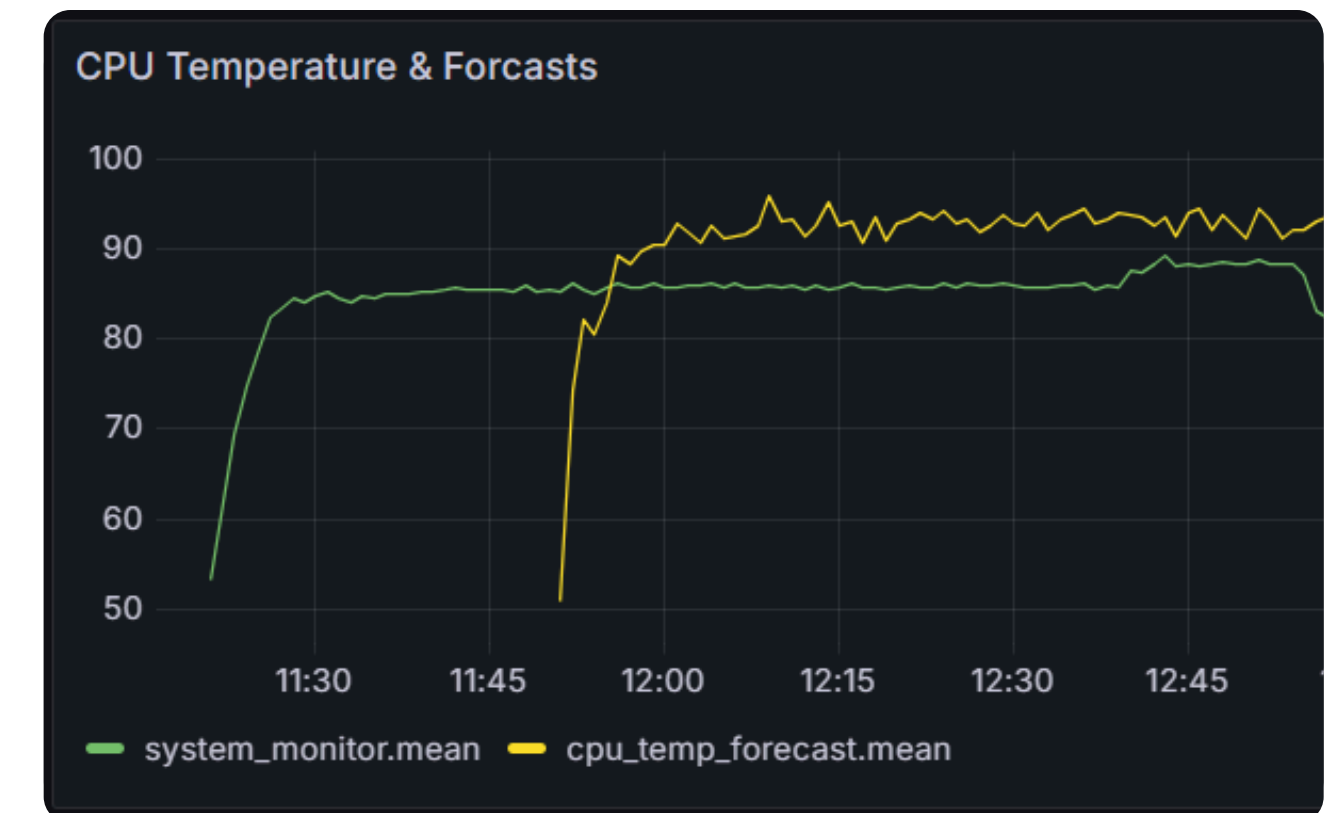
MODEL CLASSIFICATIONS

Identifications of the model are shown within both Node-Red and Graffana

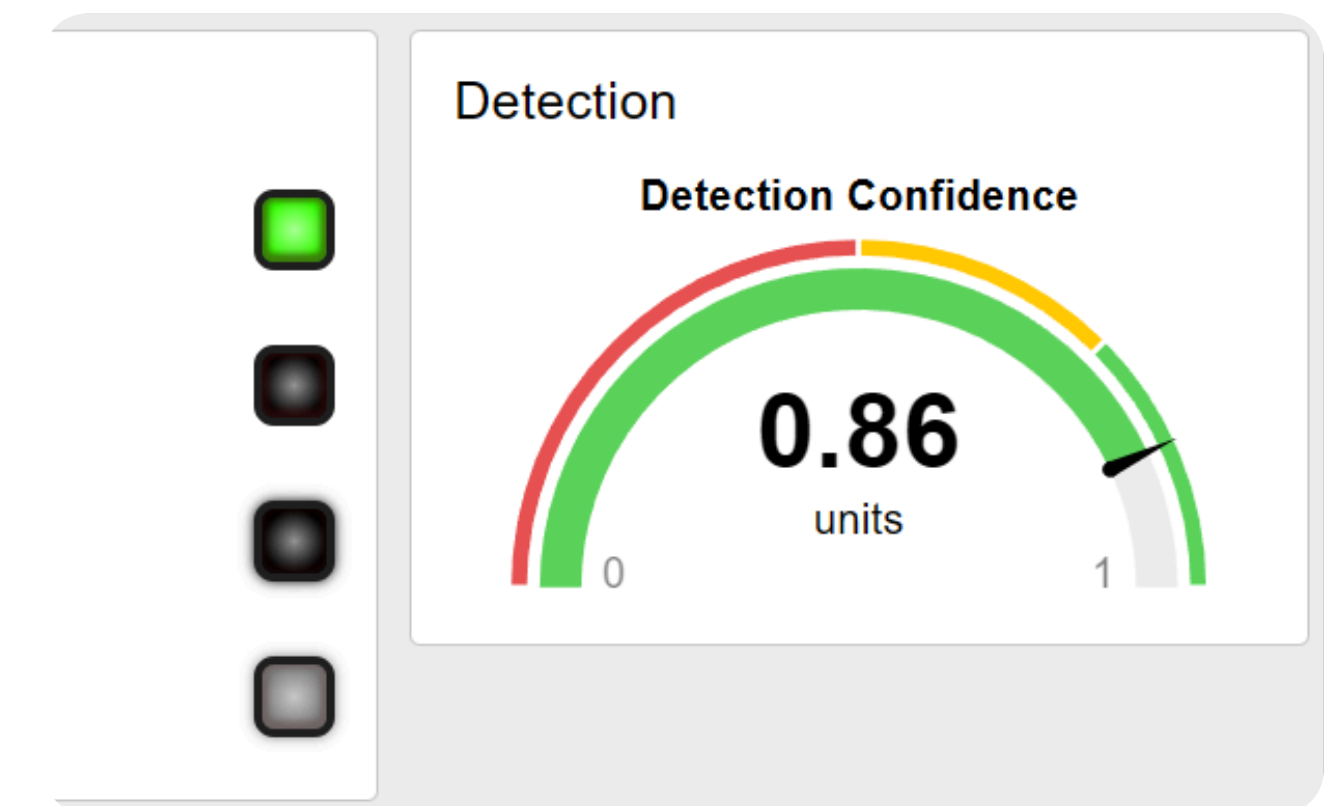
The applications results can be visualized in both Grafana and Node-Red. Grafana works by querying the Influx DB against a set of conditions to then plot the data.

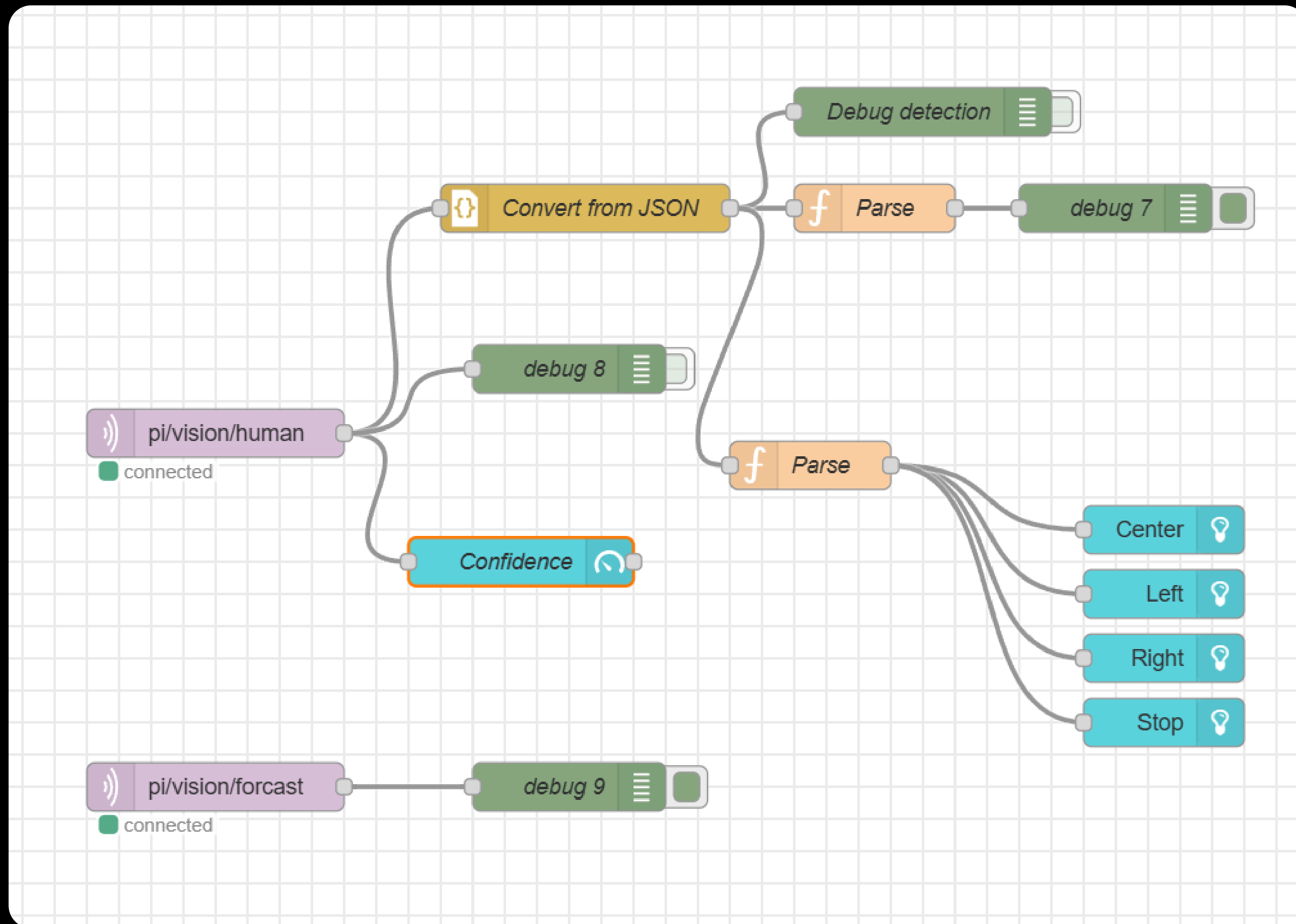
Where as Node-Red utilizes MQTT and listens for transmissions, then reading these and displaying the message. As such Node Red provides more current information compared to Grafana which shows it historically.

GRAFANA



NODE-RED





NODE-RED

Node red is made up of a series of interconnected code blocks which together form a flow.

In this flow it awaits to hear a message on MQTT, before passing it from a json format and calculating which field has been selected. Before passing these results into Dashboard 2.0 blocks.

Mean confidence



Queries 1

Transformations 0

▼

A

(Detection - influxDb)

?

📄

FROM	default	ai_vision_human	×	WHERE	+			
SELECT	field	(confidence)	×	mean	()	×	+	
GROUP BY	time	(\$__interval)	×	fill	(null)	×	+	
TIMEZONE	(optional)	ORDER BY TIME	ascending	▼				
LIMIT	(optional)	SLIMIT	(optional)					
FORMAT AS	Time series	▼	ALIAS	Naming pattern				

Grafana

Unlike Node-red, Grafana uses database queries to get its data. Meaning instead of just having data from now it has historical data, allowing for longer term analysis.

Future improvements

Currently the system functions based on the specification in the assignment. This functionality could be expanded to include:

01 Increase frame rate

02 More regions (Top, middle, bottom)

03 Ability to detect more objects

04 Additional system monitoring (E.g. memory usage)

05 Improved accuracy for forecasting CPU temperature

