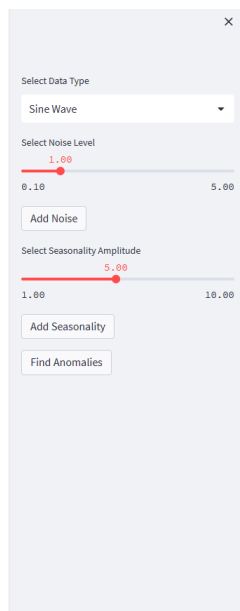


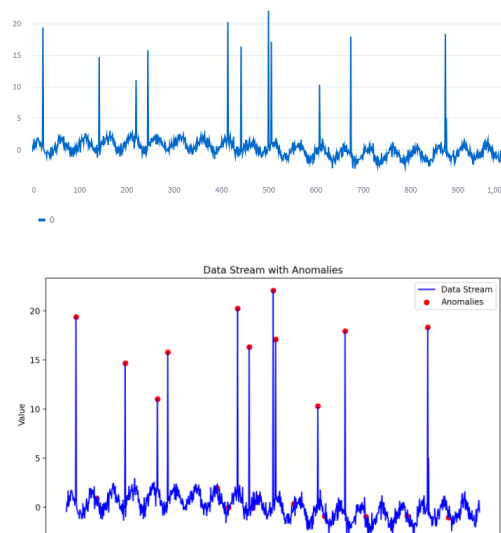
Algorithm Selection: Here, I used seasonal trend decomposition using LOESS (STL) for seasonality and trend detection combined with a simple statistical method like the Z-score for anomaly detection (Taken Z-threshold = 3). Here, I can also use ANN and Autoencoders for this purpose, but these cannot perform well for seasonal data.

Data Stream Simulation: I created a function that can generate Linear, Sine-Wave, and Random-Walk type data with handy noise to detect anomalies

Visualization: For visualization, I used a Streamlit library as it can provide a nice UI; here, I added some tabs to change the data stream(e.g. Linear, Sine, Random) and also added features to increase and decrease seasonality and noise to detect anomalies.

A Streamlit web application interface for anomaly detection. It features a 'Select Data Type' dropdown menu currently set to 'Sine Wave'. Below it is a 'Select Noise Level' slider ranging from 0.10 to 5.00, with a red dot at 1.00. There is an 'Add Noise' button. Further down is a 'Select Seasonality Amplitude' slider ranging from 1.00 to 10.00, with a red dot at 5.00. There are 'Add Seasonality' and 'Find Anomalies' buttons at the bottom.

Anomaly Detection in Data Stream



For this tool you can refer to this network URL : <http://10.145.117.60:8501>

If it does not work pls, follow this procedure :

1. Install my requirement file (all libraries that are mentioned)
2. Download my main.py file and save to a desired location
3. Open your CMD and go to the folder where your file is saved
4. run this command : **streamlit run <filename>.py**

Afterwards you will be able to see this nice visualization tool, it consist of many data stream noise adding features

If you are not able to perform above step :

I am also adding a **main.ipynb** file which consists of code after running the python code you are just able to detect and see atleast one visualization.