Project Title:
Efficient Data Stream Anomaly Detection

Project Description:
Your task is to develop a Python script capable of detecting anomalies in a continuous data stream. This stream, simulating real-time sequences of floating-point numbers, could represent various metrics such as financial transactions or system metrics. My focus is on identifying unusual patterns, such as exceptionally high values or deviations from the norm.

My solution

General description of my solution

The **Efficient Data Stream Anomaly Detection** project involves developing a Python-based system to detect anomalies in real-time data streams, simulating metrics such as energy consumption and energy price with seasonal fluctuations. A Python Flask server generates real-time data, where values fluctuate between specified minimums and maximums, creating realistic patterns for anomaly detection. The system utilizes Prometheus and the Prophet model to forecast data and identify anomalies when values deviate from predictions, computing key metrics like Mean Absolute Error (MAE) and anomaly counts, which are visualized in Grafana dashboards. An incident detection algorithm categorizes anomalies into severity levels based on predefined rules, triggering alerts when thresholds are exceeded across metrics. With Prometheus and Grafana facilitating real-time monitoring and visualizations of system performance, anomalies, and severity levels, the system intelligently manages transient anomalies to minimize false alerts.

Simulator (simulator.py)

I created a simulator using a Python Flask server to generate real-time data for two metrics: energy consumption and energy price. The simulator produces a random floating-point value every 2 seconds.

For seasonality, I implemented a system where each metric fluctuates between a specified minimum and maximum value, gradually increasing to the maximum and then decreasing back to the minimum over the course of a minute.

- Energy consumption:
    - Min = 1.0, Max = 500.0
    - Increment value = 20
- Energy price:
    - Min = 10.0, Max = 1000.0
    - Increment value = 50

This structure creates a realistic simulation with regular periodic changes in the metrics.

Simulator usage
python simulator.py

Monitor (monitor.py)

The script monitors real-time metrics by fetching data from Prometheus and applying a Prophet model for forecasting. It tracks and detects anomalies when actual values deviate from predicted ranges. Key metrics like Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and anomaly counts are computed and exported as Prometheus metrics.

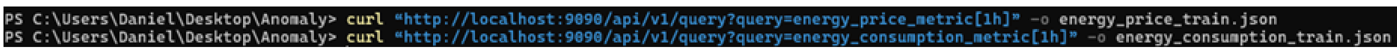
The script runs in a loop, updating metrics every 30 seconds and uses Prometheus Gauges to visualize the minimum, actual, and maximum predicted values, enabling real-time monitoring of system performance and anomalies.

The **Prophet mode**l is a time series forecasting tool that identifies trends and seasonality in data. It predicts future values with confidence intervals and flags anomalies when observed values fall outside these ranges. It's ideal for handling seasonal patterns, making it effective for detecting anomalies in energy consumption and price data.

Training data

To train the Prophet model, I downloaded datasets for energy consumption and energy price. These datasets contain time series data that represent the metrics to be monitored for anomalies. The data was preprocessed and used to fit the Prophet model, which learns the underlying trends and seasonality patterns for accurate forecasting. These datasets serve as the foundation for detecting deviations in future data, ensuring robust anomaly detection.

energy_consumption_train.json and energy_price_train.json



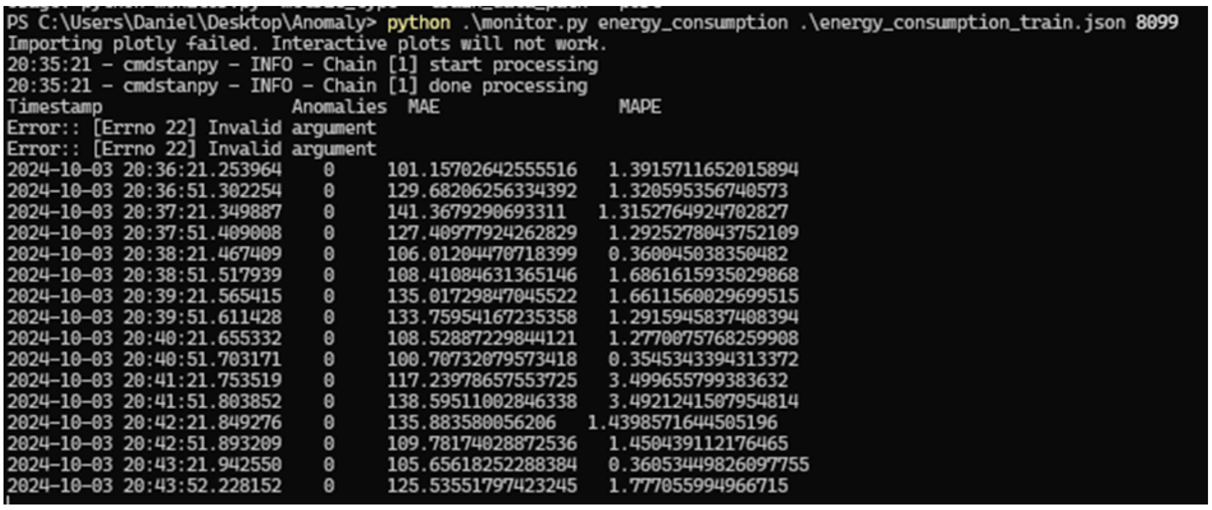
```
PS C:\Users\Daniel\Desktop\Anomaly> curl "http://localhost:9090/api/v1/query?query=energy_price_metric[1h]" -o energy_price_train.json
PS C:\Users\Daniel\Desktop\Anomaly> curl "http://localhost:9090/api/v1/query?query=energy_consumption_metric[1h]" -o energy_consumption_train.json
```

Monitor usage

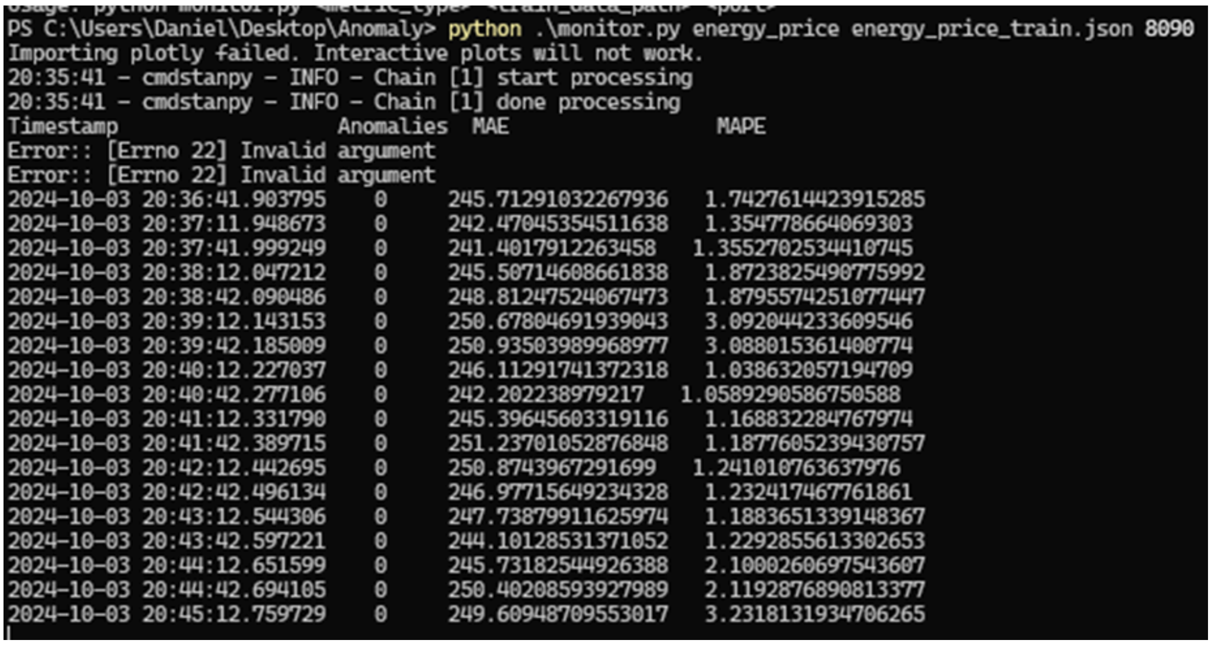
I set metric_type, train_data_path and port as command line variables so that I can be able to run a script for more than 1 metric at the same time facilitating monitoring of many metrics.

Usage: python monitor.py <metric_type> <train_data_path> <port>

Screenshot of monitoring logs for each metric



```
PS C:\Users\Daniel\Desktop\Anomaly> python .\monitor.py energy_consumption .\energy_consumption_train.json 8099
Importing plotly failed. Interactive plots will not work.
20:35:21 - cmdstanpy - INFO - Chain [1] start processing
20:35:21 - cmdstanpy - INFO - Chain [1] done processing
Timestamp                  Anomalies  MAE                 MAPE
Error:: [Errno 22] Invalid argument
Error:: [Errno 22] Invalid argument
2024-10-03 20:36:21.253964    0     101.15702642555516   1.3915711652015894
2024-10-03 20:36:51.302254    0     129.68206256334392   1.320595356740573
2024-10-03 20:37:21.349887    0     141.3679290693311   1.315276492470282 7
2024-10-03 20:37:51.409008    0     127.40977924262829   1.2925278043752109
2024-10-03 20:38:21.467409    0     106.01204470718399   0.360045038350482
2024-10-03 20:38:51.517939    0     108.41084631365146   1.6861615935029868
2024-10-03 20:39:21.565415    0     135.01729847045522   1.6611560029699515
2024-10-03 20:39:51.611428    0     133.75954167235358   1.2915945837408394
2024-10-03 20:40:21.655332    0     108.52887229844121   1.2770075768259908
2024-10-03 20:40:51.703171    0     100.70732079573418   0.3545343394313372
2024-10-03 20:41:21.753519    0     117.23978657553725   3.499655799383632
2024-10-03 20:41:51.803852    0     138.59511002846338   3.4921241507954814
2024-10-03 20:42:21.849276    0     135.883580056206   1.4398571644505196
2024-10-03 20:42:51.893209    0     109.78174028872536   1.450439112176465
2024-10-03 20:43:21.942550    0     105.65618252288384   0.36053449826097755
2024-10-03 20:43:52.228152    0     125.53551797423245   1.77705599496671 5
```



```
Usage: python monitor.py <metric_type> <train_data_path> <port>
PS C:\Users\Daniel\Desktop\Anomaly> python .\monitor.py energy_price energy_price_train.json 8090
Importing plotly failed. Interactive plots will not work.
20:35:41 - cmdstanpy - INFO - Chain [1] start processing
20:35:41 - cmdstanpy - INFO - Chain [1] done processing
Timestamp                  Anomalies  MAE                 MAPE
Error:: [Errno 22] Invalid argument
Error:: [Errno 22] Invalid argument
2024-10-03 20:36:41.903795    0     245.71291032267936   1.7427614423915285
2024-10-03 20:37:11.948673    0     242.47045354511638   1.354778664069303
2024-10-03 20:37:41.999249    0     241.4017912263458   1.3552702534410745
2024-10-03 20:38:12.047212    0     245.50714608661838   1.8723825490775992
2024-10-03 20:38:42.090486    0     248.81247524067473   1.8795574251077447
2024-10-03 20:39:12.143153    0     250.67804691939043   3.092044233609546
2024-10-03 20:39:42.185009    0     250.93503989968977   3.088015361400774
2024-10-03 20:40:12.227037    0     246.11291741372318   1.038632057194709
2024-10-03 20:40:42.277106    0     242.202238979217   1.0589290586750588
2024-10-03 20:41:12.331790    0     245.39645603319116   1.168832284767974
2024-10-03 20:41:42.389715    0     251.23701052876848   1.1877605239430757
2024-10-03 20:42:12.442695    0     250.8743967291699   1.241010763637976
2024-10-03 20:42:42.496134    0     246.97715649234328   1.232417467761861
2024-10-03 20:43:12.544306    0     247.73879911625974   1.1883651339148367
2024-10-03 20:43:42.597221    0     244.10128531371052   1.2292855613302653
2024-10-03 20:44:12.651599    0     245.73182544926388   2.1000260697543607
2024-10-03 20:44:42.694105    0     250.40208593927989   2.1192876890813377
2024-10-03 20:45:12.759729    0     249.60948709553017   3.2318131934706265
```

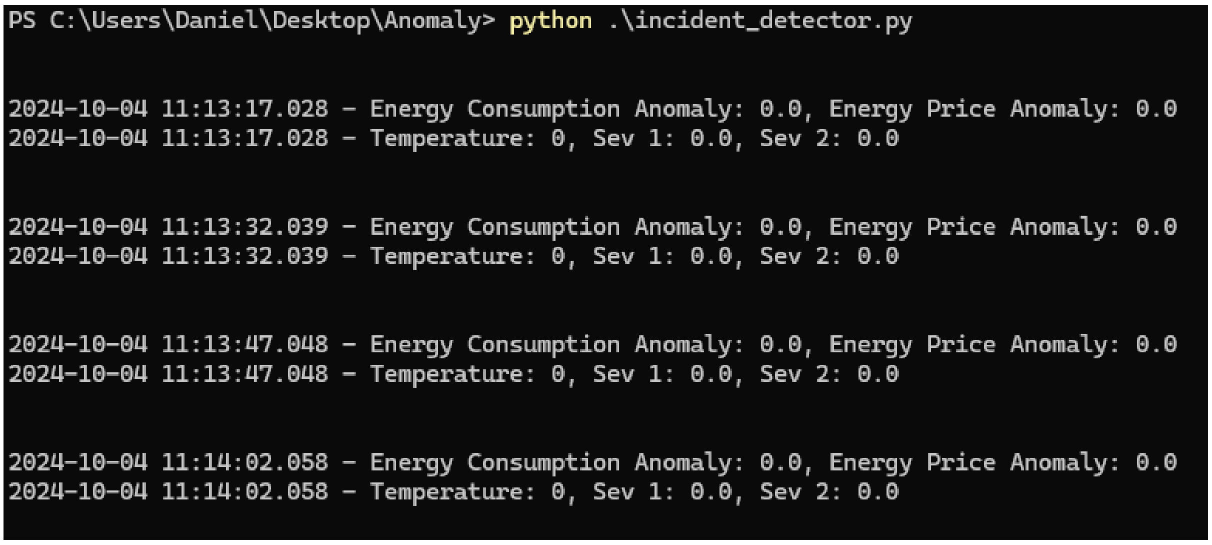
Incident detector (incident_detector.py)

I developed an algorithm to collect and assess anomalies by checking their severity based on predefined rules. The algorithm monitors two metrics, energy consumption and energy price, and categorizes incidents into two levels of severity:

- **Severity 2**: Triggered when the threshold is exceeded, but only one metric has an anomaly.
- **Severity 1**: Triggered when both metrics have anomalies and the threshold is exceeded.

The algorithm continuously monitors the anomaly counts, applying decay to reduce accumulated values over time, and raises alerts when conditions for Severity 1 or 2 are met.

Incident detector logs

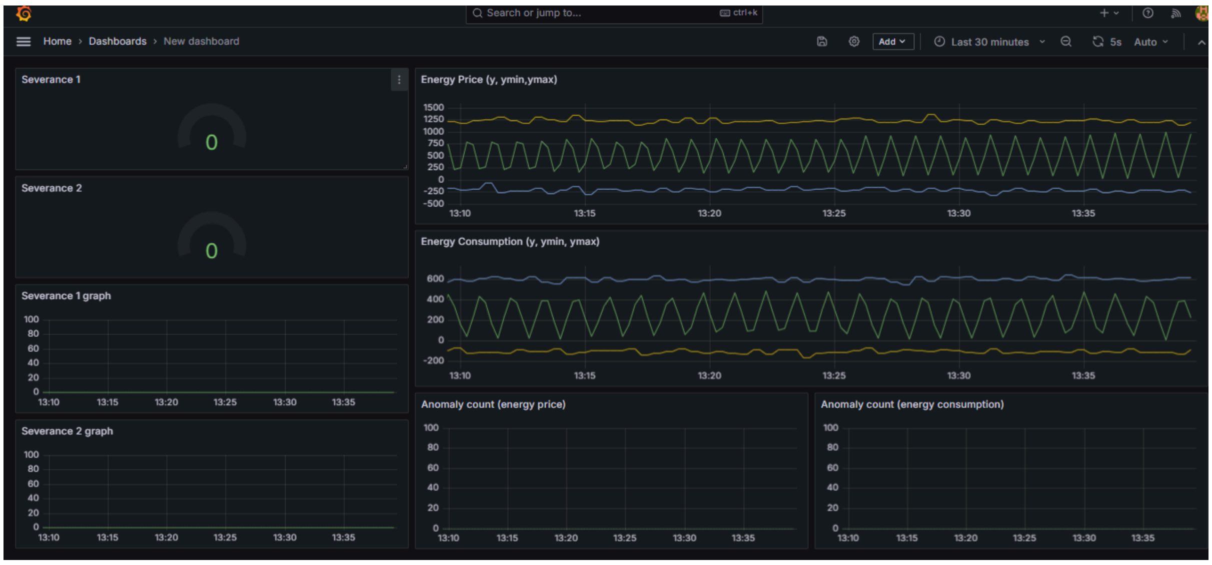


```
PS C:\Users\Daniel\Desktop\Anomaly> python .\incident_detector.py


2024-10-04 11:13:17.028 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 0.0
2024-10-04 11:13:17.028 - Temperature: 0, Sev 1: 0.0, Sev 2: 0.0


2024-10-04 11:13:32.039 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 0.0
2024-10-04 11:13:32.039 - Temperature: 0, Sev 1: 0.0, Sev 2: 0.0


2024-10-04 11:13:47.048 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 0.0
2024-10-04 11:13:47.048 - Temperature: 0, Sev 1: 0.0, Sev 2: 0.0


2024-10-04 11:14:02.058 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 0.0
2024-10-04 11:14:02.058 - Temperature: 0, Sev 1: 0.0, Sev 2: 0.0
```

Visualisation Dashboard

I used Prometheus and Grafana for real-time monitoring and visualization of the metrics. Prometheus collected data from the anomaly detection system, storing it as time series data. I then utilized Grafana to create dynamic dashboards, allowing for clear visualization of key metrics such as anomaly counts, incident severity, and performance indicators. This setup provided an intuitive way to track system behavior and detect issues in real-time.

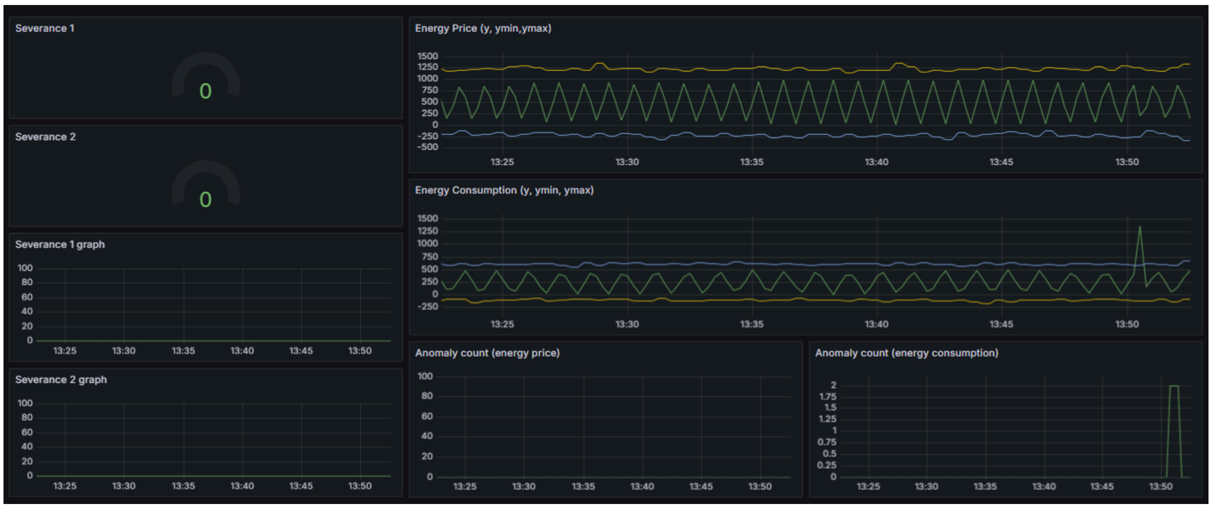
Grafana Dashboard


Search or jump to...
Home › Dashboards › New dashboard
Add
Last 30 minutes
5s
Auto
Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

I manually introduced anomalies into the system to observe its behavior and assess how it handles unexpected deviations from normal data patterns.

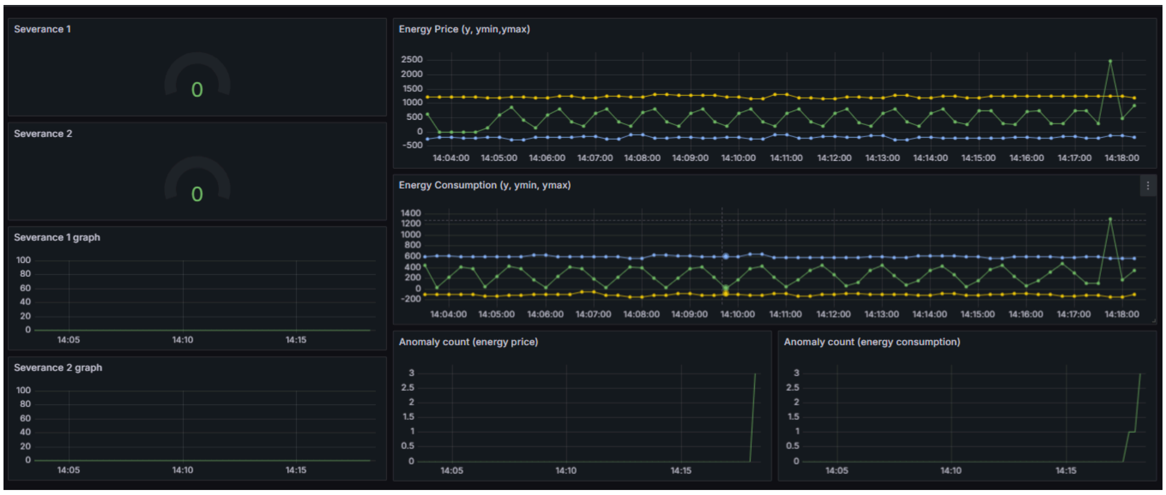
Different cases

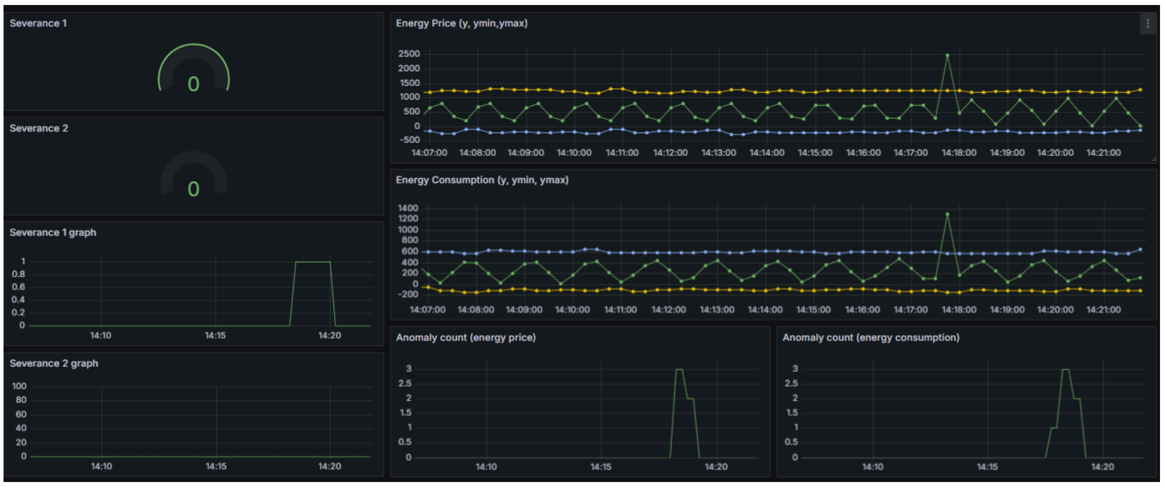
- **A transient spike in one metric**


Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

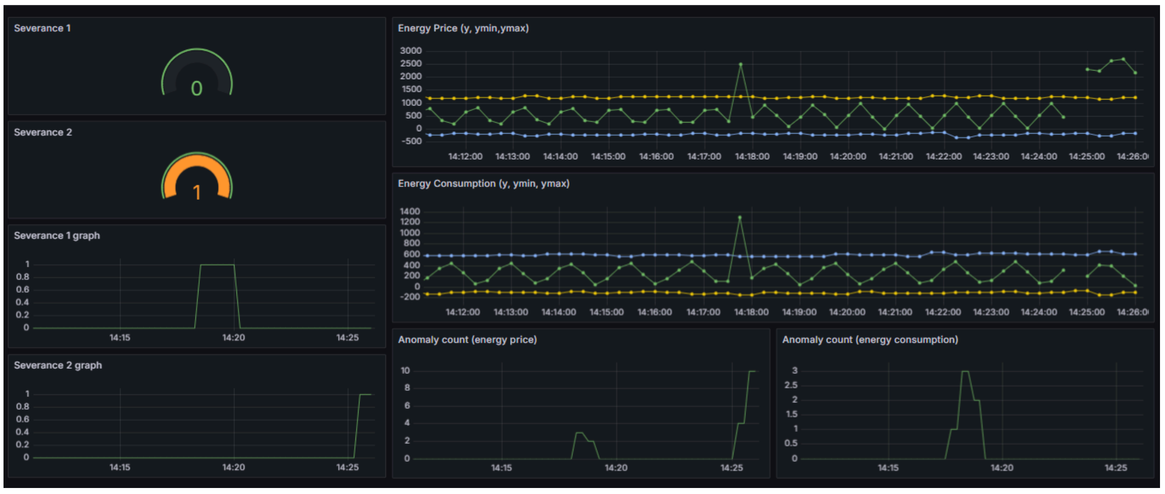
In case of a transient anomaly, the system is intelligent enough to recognize it and won’t trigger an alert every time, ensuring more accurate incident detection.

- **A transient spike in both metrics**

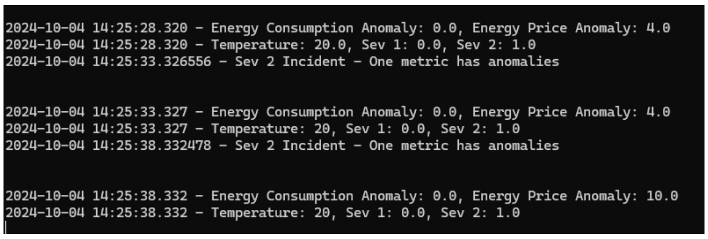

Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)


Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

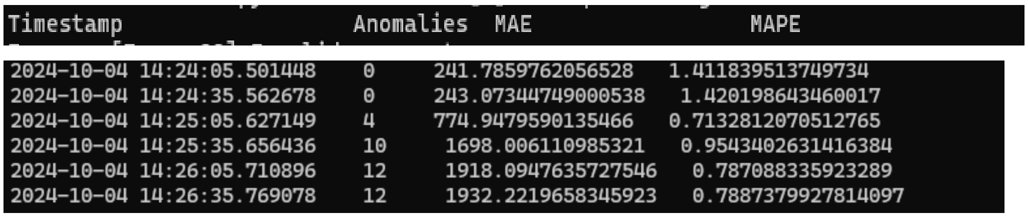
- **A prolonged spike in one metric beyond the threshold**


Severance 1
0
Severance 2
1
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

Below are the incident and monitor logs



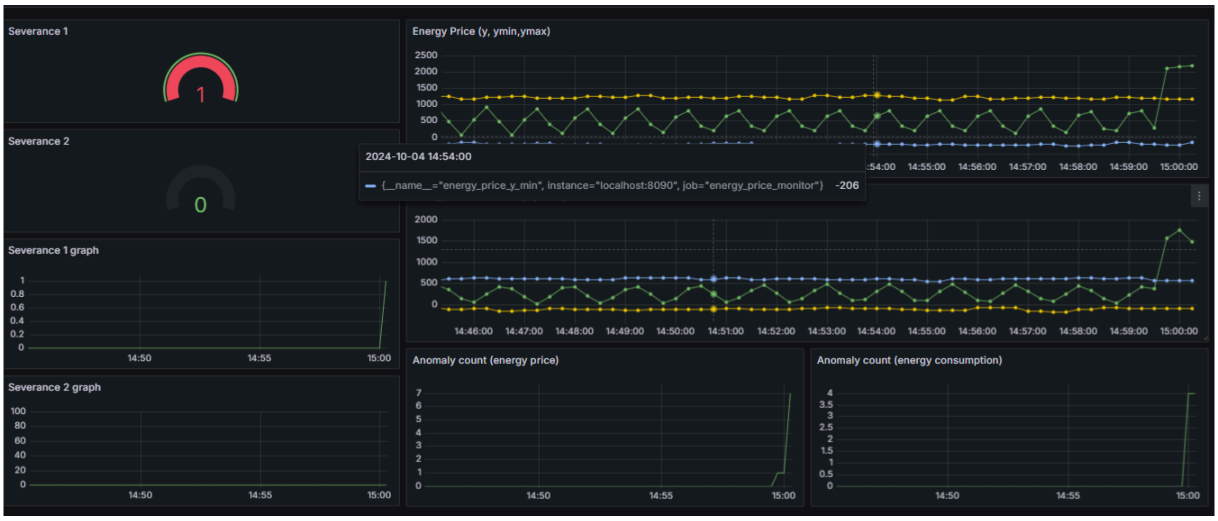
```
2024-10-04 14:25:28.320 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 4.0
2024-10-04 14:25:28.320 - Temperature: 20.0, Sev 1: 0.0, Sev 2: 1.0
2024-10-04 14:25:33.326556 - Sev 2 Incident - One metric has anomalies


2024-10-04 14:25:33.327 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 4.0
2024-10-04 14:25:33.327 - Temperature: 20, Sev 1: 0.0, Sev 2: 1.0
2024-10-04 14:25:38.332478 - Sev 2 Incident - One metric has anomalies


2024-10-04 14:25:38.332 - Energy Consumption Anomaly: 0.0, Energy Price Anomaly: 10.0
2024-10-04 14:25:38.332 - Temperature: 20, Sev 1: 0.0, Sev 2: 1.0
```

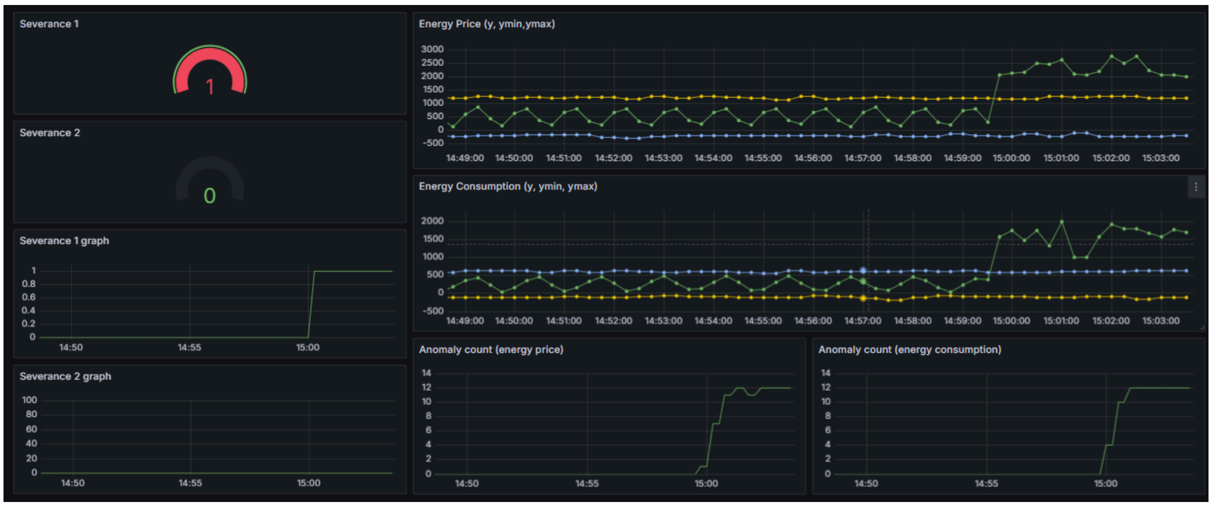


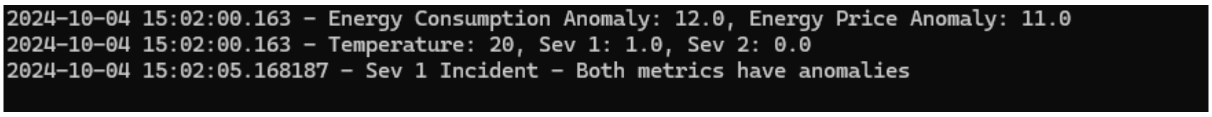
```
Timestamp                  Anomalies  MAE                  MAPE
2024-10-04 14:24:05.501448    0      241.7859762056528   1.411839513749734
2024-10-04 14:24:35.562678    0      243.07344749000538   1.420198643460017
2024-10-04 14:25:05.627149    4      774.9479590135466   0.7132812070512765
2024-10-04 14:25:35.656436    10      1698.006110985321   0.9543402631416384
2024-10-04 14:26:05.710896    12      1918.0947635727546   0.787088335923289
2024-10-04 14:26:35.769078    12      1932.2219658345923   0.7887379927814097
```

- **A prolonged spike in both metrics beyond the threshold**

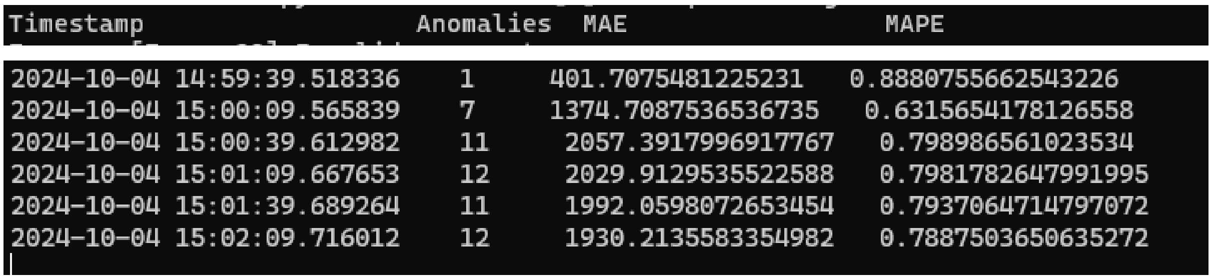
This is a severance 1 incident since both metrics have anomalies and the threshold has been passed.


Severance 1
1
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
2024-10-04 14:54:00
{__name__="energy_price_y_min", instance="localhost:8090", job="energy_price_monitor"} -206
Anomaly count (energy price)
Anomaly count (energy consumption)

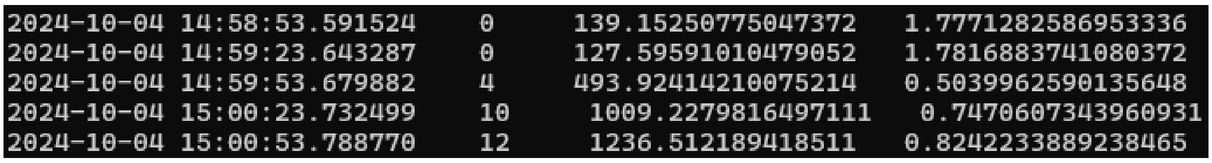

Severance 1
1
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)



```
2024-10-04 15:02:00.163 - Energy Consumption Anomaly: 12.0, Energy Price Anomaly: 11.0
2024-10-04 15:02:00.163 - Temperature: 20, Sev 1: 1.0, Sev 2: 0.0
2024-10-04 15:02:05.168187 - Sev 1 Incident - Both metrics have anomalies
```



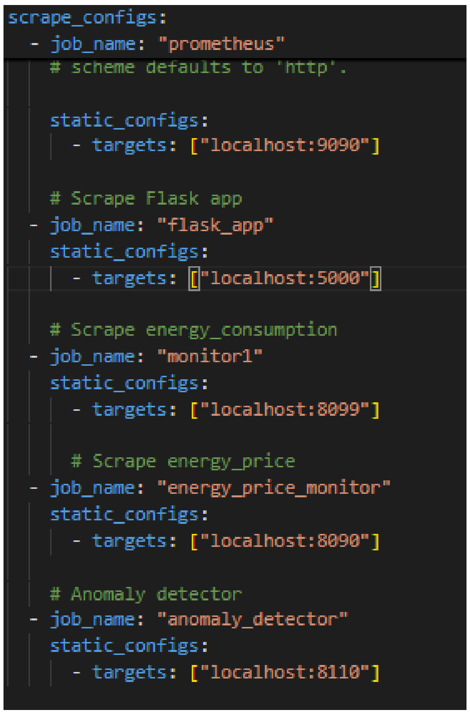
```
Timestamp                   Anomalies  MAE                 MAPE
2024-10-04 14:59:39.518336    1     401.7075481225231    0.8880755662543226
2024-10-04 15:00:09.565839    7     1374.7087536536735    0.6315654178126558
2024-10-04 15:00:39.612982    11     2057.3917996917767    0.798986561023534
2024-10-04 15:01:09.667653    12     2029.9129535522588    0.7981782647991995
2024-10-04 15:01:39.689264    11     1992.0598072653454    0.7937064714797072
2024-10-04 15:02:09.716012    12     1930.2135583354982    0.7887503650635272
```



```
2024-10-04 14:58:53.591524    0     139.15250775047372    1.7771282586953336
2024-10-04 14:59:23.643287    0     127.59591010479052    1.7816883741080372
2024-10-04 14:59:53.679882    4     493.92414210075214    0.5039962590135648
2024-10-04 15:00:23.732499    10     1009.2279816497111    0.7470607343960931
2024-10-04 15:00:53.788770    12     1236.512189418511    0.8242233889238465
```

Prometheus (prometheus.yml)

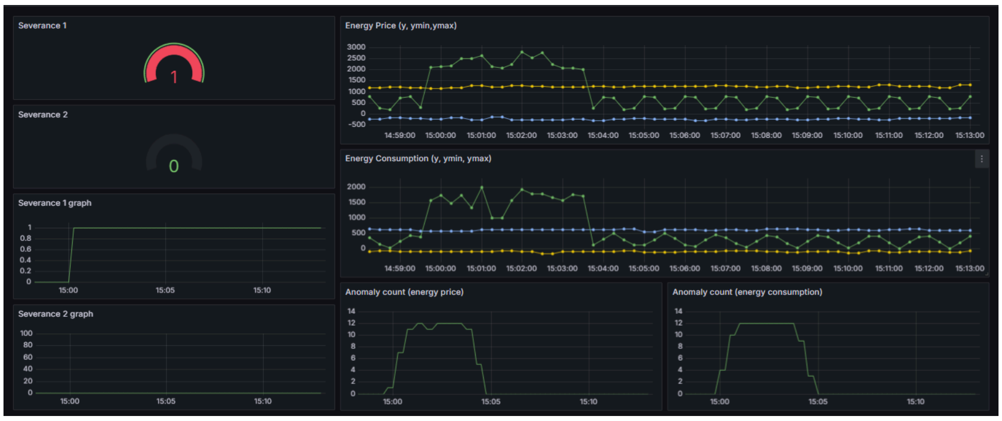
I specified all the relevant ports for the application in the Prometheus configuration YAML file. This file defines the targets for scraping metrics from the various components of the system, including the simulator and monitoring scripts. By centralizing the port configurations within the Prometheus YAML, I ensured a clear and organized setup, making it easy to manage and adjust the monitoring of different metrics as needed.

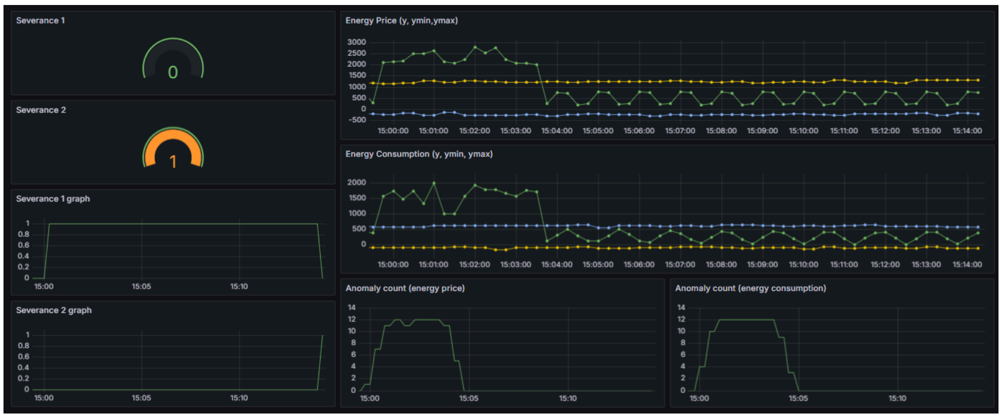


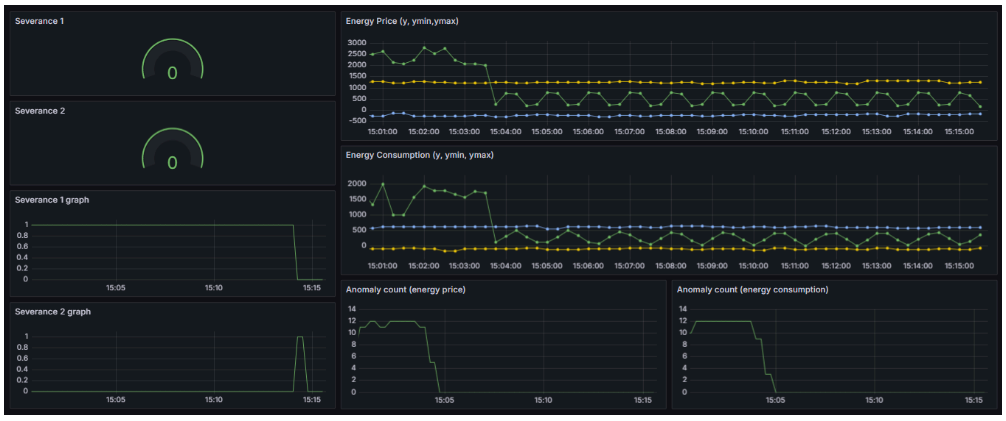
```yaml
scrape_configs:
  - job_name: "prometheus"
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

    # Scrape Flask app
  - job_name: "flask_app"
    static_configs:
      - targets: ["localhost:5000"]

    # Scrape energy_consumption
  - job_name: "monitor1"
    static_configs:
      - targets: ["localhost:8099"]

      # Scrape energy_price
  - job_name: "energy_price_monitor"
    static_configs:
      - targets: ["localhost:8090"]

    # Anomaly detector
  - job_name: "anomaly_detector"
    static_configs:
      - targets: ["localhost:8110"]
```

Appendix

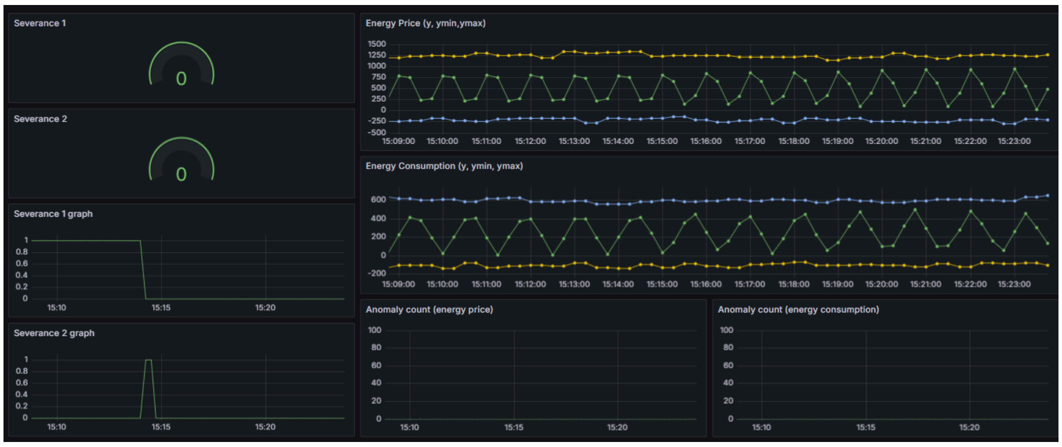
System goes back to normal after incident anomalies are fixed.


Severance 1
1
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

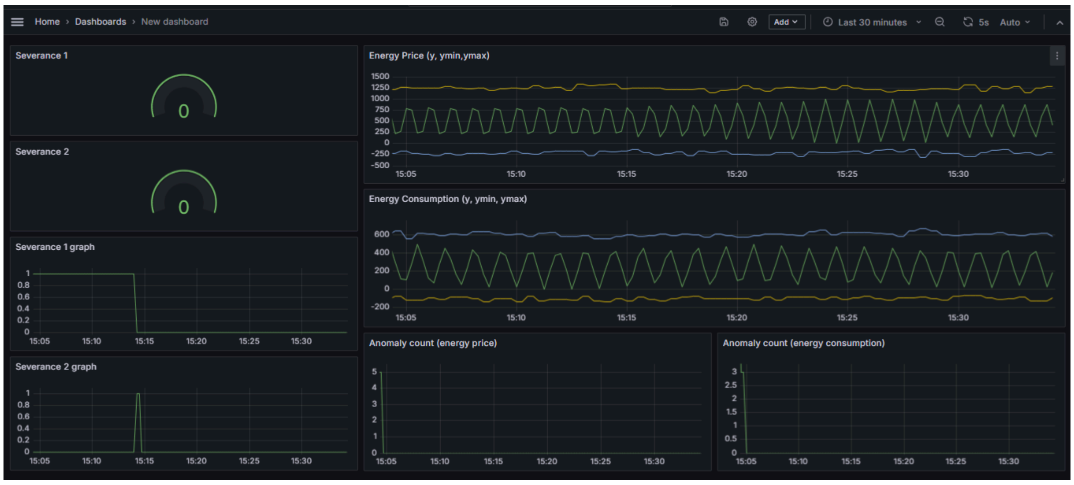

Severance 1
0
Severance 2
1
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)


Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

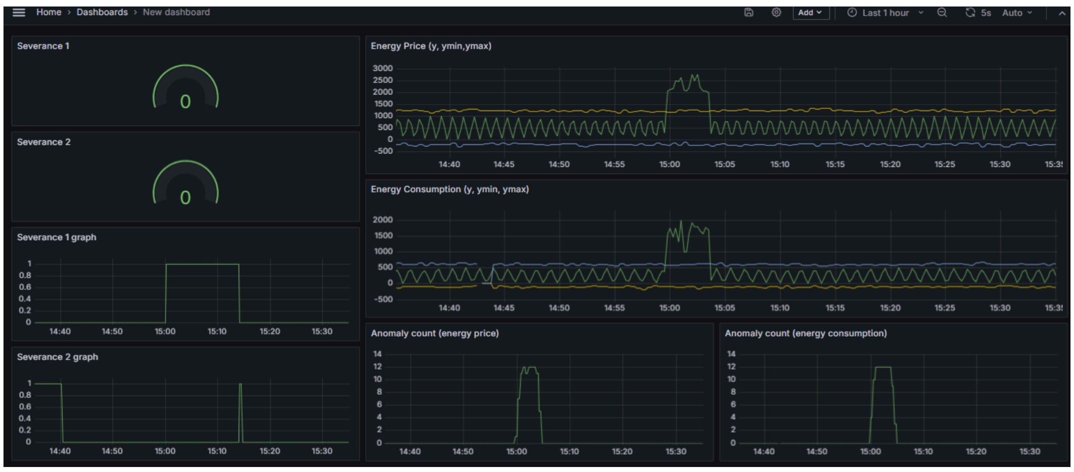
Other screen shots of the dashboard


Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

30 minutes range


Home › Dashboards › New dashboard
Add
Last 30 minutes
5s
Auto
Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)

1 hour range


Home › Dashboards › New dashboard
Add
Last 1 hour
5s
Auto
Severance 1
0
Severance 2
0
Severance 1 graph
Severance 2 graph
Energy Price (y, ymin,ymax)
Energy Consumption (y, ymin, ymax)
Anomaly count (energy price)
Anomaly count (energy consumption)