# Anomaly detection in smart buildings

## A case study of the University of Amsterdam's Lab42 building

submitted in partial fulfillment for the degree of master of science

Ágoston Reischl
15813568

Master Information Studies
information systems
Faculty of Science
University of Amsterdam

Submitted on June 27, 2025

| | UvA Supervisor |
|---|---|
| **Title, Name** | Dr. Victoria Degeler |
| **Affiliation** | UvA Supervisor |
| **Email** | v.o.degeler@uva.nl |

## ABSTRACT

This research develops an anomaly detection focused cyber-physical system for the Lab42 building of the University of Amsterdam. While the building is equipped with numerous sensors, that provide near real-time data on air quality, temperature and natural and artificial lights, this information is underutilized. The study investigates whether the existing sensor data is sufficient or additional sensors or data sources are needed. It is also explored if the solution can be generalized to other buildings with similar amount, but different source of data.

Several methods are evaluated, and two of them – NAIA and LSTM-Autoencoder – are selected based on their compatibility and data requirements. These are integrated into a proof-of-concept pipeline, which is validated through experiments simulating five different types of anomalies.

Results show that the proposed system can detect 89-98% of injected anomalies, generally above 83%, and in most cases above 98% recall. This proves the system's reliability in detecting anomalies, enabling preventive maintenance, improved sustainability and lower operational costs. The research confirms that the HVAC sensor-based approach can be affective and scalable in smart building environments.

## KEYWORDS

Cyber-Physical Systems, Digital Twin, Anomaly Detection, HVAC Sensor Data, LSTM-Autoencoder

## GITHUB REPOSITORY

https://github.com/reischlago/lab42_anomaly_detection

## 1 INTRODUCTION

As one of the newest buildings of the University of Amsterdam, Lab42 is equipped with a large amount of sensors that gather information about the temperature, air quality and presence in the offices, class-, study- and meeting rooms, halls and other open study spaces. The data generated by these sensors are currently underutilized, but a great amount of historical data is also stored. This large amount of data calls for some sort of monitoring system that visualizes this data in real-time and allows its analysis. These sensors are mainly for HVAC (Heating, Ventilation and Air Conditioning) monitoring purposes, that includes temperature and air quality data with the addition of natural and artificial light sensor data.

A cyber-physical system can "effectively integrate cyber and physical components using the modern sensor, computing and network technologies" [2]. An example of such a system is the Digital Twin, that is, as defined by Michael Grieves and John Vickers in 2017, "a set of virtual information constructs that fully describes a potential or actual physical manufactured product [...], any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin." [9]

A Digital Twin, if connected to its physical counterpart, can visualize all the real-time data the sensors collect, and provides a great tool to analyze, monitor and quickly report any anomalies, showing them where they are, and allows rapid human response if needed.

The thesis aims to answer the main question: **RQ1: How can a cyber-physical system of UvA's Lab42 building be used to monitor and report anomalies in the building?**
Subsequently, the following questions also need to be answered to be able to provide a well-founded statement to the above:

- SRQ1: What type of data is required to confidently recognize anomalies in the building?
- SRQ2: How can anomalies be detected with great confidence, relying solely on the data provided by the HVAC sensors?

There is a secondary question, that initiates the discussion: **RQ2: What building specific configuration is built into the proposed system?**
This question aims to find if the anomaly detection can be generalized, so it could help optimize other buildings as well.

Additional questions that reflect on the secondary question are:

- SRQ3: What building specific data cleaning, manipulation or organization is necessary for the anomaly detection system, that could raise difficulties in applying it to other buildings?
- SRQ4: What sort of data quality aspects can prevent anomaly detection in the proposed system?

These HVAC sensor data are available for most of the commercial buildings or can be retrieved in a way that does not necessarily require the installation of additional sensors, which is why, a highly reliable system based on only HVAC or other generally available data, could improve building maintenance and cost effectiveness in other buildings. That could ultimately also improve the sustainability of a building as well [3].

This research addresses the gap of the lack of a robust and accurate anomaly detection system, that relies on only four HVAC sensor data, contrasting to other methods mentioned in the following section, that relies on more diverse data. The proposed system is applied for the Lab42 building of the University of Amsterdam.

## 2 RELATED WORK

The literature review was performed by using a scholarly keyword search for each topic below, to gather all the related state of the art technologies, current theories and recent breakthroughs. It also serves as the foundation of the thesis, declaring and clarifying important definitions, such as what a Digital Twin is, or what are the criteria for anomaly detection. The following subsections aim to provide the basis for answering the research questions.

### 2.1 Cyber-physical systems

A cyber-physical system (CPS) is a system where the physical processes are controlled, monitored and generally affected by embedded computers or sensors as defined by Edward A. Lee [11]. While the concept of CPS is not new, there is an increasing interest and development in the topic, especially in the emerging technology of Digital Twins, that couples data-driven models "to a physical system to enhance its functionality" [18].

As mentioned in the introduction, a Digital Twin (DT) is a real-time copy of a physical object, that allows experimentation, data visualization and many other actions that would be impossible, or expensive to do on the physical object itself. While Somers et. al [18] observed in a systematic review of Digital-twin-based testing for cyber–physical systems in 2023 that "Digital twin definitions are wide ranging and often domain specific causing confusion due to a lack of consensus", the main characteristics of such a model or system are clear. The authors of the review classified DTs in three main categories:

- Supervisory
- Interactive
- Predictive

The categories are very self explanatory, as they describe the level of feedback interaction between the physical asset and the digital replica: from monitoring through controlling to predicting behaviour.

There is an increasing interest in the use of Digital Twins, however the idea is not new, similar concepts with different terminology exist since the early 1990s [17]. Some even consider NASA's tools used in the Apollo missions in the 1960s as the first iteration of DTs, however, they did not use this name for it [7]. DTs sit in the technological intersection of Internet of Things, Artificial Intelligence, Extended Reality and Cloud Computing [4], as these provide the key elements: continuous stream of real-time data, analysis and predictions, digital representation, and easy access of data from anywhere.

These characteristics provide a perfect tool for monitoring system behavior and the vast amount of historical data is a great foundation of machine learning models for predictions and outlier analysis.

There are studies that investigate how DTs can be used for maintenance purposes, that focus identifying anomaly patterns to optimize building operations, such as the research of Lu et al. from 2020 [13], that is discussed in Section 2.3.

## 2.2 Sensor Data

In the heart of any cyber-physical system, but especially Digital Twins, lies the data, that is provided near real-time by the thousands of sensors of the Lab42 building. That data needs to be analyzed and potentially cleaned [1], while maintaining the near-real time characteristic of it. However, the data provided by the university does not require cleaning, as the results show, in the case of the study, these are deemed irrelevant, but it is important to mention for generalizability purposes.

There are a few requirements collected by Zhang M, Tao F and Huang B et al. [21], that consider data gathering, interaction, universality, knowledge mining, data fusion, optimization and the requirements on convenient data usage. The authors also summarized the methodologies regarding the handling of Digital Twin data, that applies for the requirements. These were only applied in a limited manner, as the university data source (API) cannot be changed in the case study, and no additional data gathering is required.

While most Digital Twins have a bidirectional data-flow [7], for the purposes of this study, only one direction is required, which meets the current technological possibilities of the data retrieval system, provided by the university, and it falls into the 'supervisory' category defined by Somers et. al [18]. A unidirectional data-flow allows to build the near real-time model of the building that can be used for analysis and monitoring purposes.

## 2.3 Anomaly detection

There are studies that focus on anomaly detection using Digital Twins, such as the work of Lu et al. from 2020, where the authors are using historical and real-time data to compare early signs of anomalies. These anomalies can be identified if there is a large amount of data available with all the different sensors, that can help the system differentiate between causality and correlation. Since most of the available data from the building is restricted to Heating, Ventilation and Air Conditioning (HVAC), the challenge of this research is to confirm if such data is enough to correctly identify anomalies, and if the results suggest that it is not, then what kind of sensors or data are required to do so.

### *2.3.1 ATTAIN method.*

In 2021 Qinghua Xu, Shaukat Ali and Tao Yue [20] has presented a novel approach called ATTAIN for detecting anomalies using the Digital Twin technology. Their approach uses sensor and actuator values, from historical and real-time data acquired from an operational cyber-physical system, that is processed by a learning digital twin. Their anomaly detector (the DT) is trained on real-time data only, with a Generative Adversarial Network (GAN) in its backbone framework. While they focus on cyber attacks, the methodology could be interesting for physical changes, as their approach is to take samples from the data, generate ground truth of the sample, and then predict the DT's current state and compare it to the actual state, considering certain pre-defined thresholds. This logic returns if the current state of the twin is normal or not.

### *2.3.2 NAIA method.*

Another novel approach by Izaskun et. al [14] has been first published in 2022, "for the detection of anomalous energy consumption patterns in industrial cyber-physical systems", where they used the 'NAIA' methodology to detect unsupervised anomalies. The logic of this methodology is similar to the one mentioned before: determining a baseline (nominal performance) and detect outliers based on daily energy consumption patterns. The data for their experimental setup was derived from energy consumption on production and auxiliary lines. For the latter they used similar data that is also available from the Lab42 building, such as lightning and air conditioning data.

### *2.3.3 LSTM-autoencoder approach.*

In 2023, Wei Hu, Xin Wang, Khery Tan and Yiyu Cai [10] wrote a research paper on predictive maintenance for indoor climate, especially focusing on fault detection in the air conditioning systems. They used a Digital Twin enabled predictive maintenance framework using Long short-term memory (LSTM) and Autoencoder technology. "LSTM is a specialized recurrent neural network (RNN) architecture designed to address the challenges of modeling long-term dependencies in sequential data" [10]. This technology enables contextual knowledge over previous time frames, which is effective in recognizing temporal patterns.

"The autoencoder is a fundamental neural network architecture used in unsupervised learning tasks which aims to capture the most salient features and learn the meaningful representation of the input data" [10].

Their method requires a trained model on the historical data, and real-time data to capture if there are any discrepancies with the expected values. Since the authors used a similar dataset to what is available from Lab42, their method seems highly relevant for the project, and it is worth further investigation and replication.

#### 2.3.4 *Overview of inspected methods.*

The above discussed methods are summarized in Table 1 to show their strengths and weaknesses in the context of the Lab42 solution.

As the table shows, there is no directly applicable solution for the Lab42 sensor data and cyber-physical system, nevertheless the NAIA and the LSTM-Autoencoder technology shows enough similarity, that their combination can fit the requirements and capabilities of Lab42. Their implementation and evaluation is described in the following sections. Sater and Hamza [16] has also reached the conclusion in their study, that an LSTM based system yields the best results, with the highest precision. Their results are used for benchmarking purposes, and the results of this research are compared to theirs in Section 6.5.

## 3 METHODOLOGY

To answer the research questions, different approaches will be used. First of all, the data will be evaluated and cleaned if necessary, based on the established pipeline and considered if the available data is sufficient to support the features determined by the prototype.

Then to answer the question how the anomalies can be detected, a detailed research has been performed – as described in Section 2.3, and algorithms and methods were implemented, to determine if the system contains anomalies or not. Wickramasinghe et al. [19] has researched the topic with similar methods in 2023, and concluded that they are effective in anomaly detection. This approach requires a determined baseline, that models a properly functioning building; and the model of the current system. Then a comparison needs to be done, that shows the deviations, which are then evaluated based on a predefined threshold, that determines if the deviation is considered an anomaly. This can be done by retrieving change logs and the analysis of their effects. The latter is the approach to answer the question if the anomaly requires human interaction or further investigation or not, however that is outside of the scope of the thesis and is mentioned in Sub-section 7.1.4.

To answer the secondary research question (RQ2), it needs to be evaluated if the anomaly detection model uses any building specific configuration, or can be easily adapted or generalized, so if used with datasets from other buildings, it would work, or it would need extensive tailoring. This is determined by analyzing the steps of the work done regarding the first research question.

| Method | ATTAIN | NAIA | LSTM - Autoencoder |
|---|---|---|---|
| **Core Methodology** | Generative Adversarial Network (GAN) | Unsupervised clustering (k-means) | Long Short-Term Memory (LSTM) and autoencoder |
| **Data Requirements** | Historical and real-time sensor and actuator data | Historical and real-time sensor data | Historical and real-time sensor data |
| **Example use in the referenced literature** | Detecting cyber attacks | Using energy consumption data on different power lines | Using air quality data |
| **Strengths** | Precise pattern recognition of causality | Precise and early recognition of the effects of the main line malfunction | Contextual knowledge of temporal patterns |
| **Weaknesses** | Requires many data sources | Uses energy consumption data, that is more indicative for small changes in the data | More focused on long-term effects |
| **Applicability to Lab42** | The available sensor data is not enough for this method in the case of Lab42 | The kind of information that is available is different, however, with some modification the method could be applied for Lab42 | Similar data and use case as Lab42, however the discrepancies in the available historical data can cause issues in the contextual knowledge generation |

**Table 1: Comparison of the methods and use cases**

### 3.1 What is an anomaly?

First and foremost in order to identify anomalies, a definition of 'anomaly' is needed. In other publications a definition of anomaly has been referenced from Chandola et. al [6] as "Anomalies are patterns in data that do not conform to a well defined notion of normal behavior." Within the scope of this thesis, it is defined as follows: an event that happens over a certain period of time, that is not the expected outcome of the inspected parameters, but a result of an unknown factor.

In subject to the system, which observes a set of values that are received from the sensors, any inconsistency in their relations over time, can be considered an anomaly if there is no correction observed.

This fits the previously referenced definition, with the additional mention of parameters, that further clarifies what defines the normal behavior, to which the data conforms to.

## 3.2 The NAIA and LSTM-Autoencoder methods

*3.2.1 NAIA.* The NAIA method, by Izaskun et al. [14] relies on unsupervised machine learning on the sensor data – in their use case, data of the power lines and the auxiliary lines of factory production lines. Even though the data is different, the method can be applied in the four phases:

- Definition of the nominal performance
  - Loading and aggregating daily profiles with hourly resolution
  - Iterative clustering using K-means and Principal Component Analysis (PCA)
  - IQR based outlier detection within the clusters
- Validation of the trained model
  - Applying new data on the model, sorting to the existing clusters
  - Checking cluster-radius and linear model consistency
- Discovering reasons of anomalous behaviour
- Knowledge management of the nominal performance

The first two phases of the method are applied in the system for Lab42, as they focus on the detection of the anomaly in data, but the last two phases that focus on the building maintenance and management of physical anomalies fall outside of the scope of the current thesis, however it can provide the basis of the future work.

*3.2.2 LSTM-Autoencoder.* This method relies on the long short-term memory (LSTM) and autoencoder methods [10]. While LSTM can predict failure based on labeled historical data (labels of the past failures of the system), the autoencoder can detect inconsistencies using unsupervised learning. It is important to use these two in comparison with each other, as each of them alone cannot sufficiently distinguish between an anomaly and manual intervention.

The LSTM method learns how sequences of conditions lead to system degradation or failure, but requires the labeled data to know when a failure or degradation occurs. The labels are generated from the air quality values as described in Table 2, based on the "$CO_2$ Guideline Concentrations" from Lowther et al. [12].

| $CO_2$ppm | Label |
|---|---|
| 0 - 500 | Excellent |
| 501 - 1000 | Good |
| 1001 - 1500 | Moderate |
| 1501 - 2000 | Poor |

**Table 2: Air quality labels**

The autoencoder method uses the reconstruction method, which means, that based on the past values, it tries to reconstruct the expected values. These are then compared with the actual real-time values, and if they breach a certain threshold, they are considered an anomaly, and they are reported accordingly, with a certain warning period. The warning period in the works of Hu et al. [10] is set to 30 minutes which was applied in the paper as a best practice. This plays a crucial role if the system is working with real-time data, as increasing this value can allow more time to act upon these anomalies, however they can increase the amount of false positives as well. Since the proof of concept system in this study does not work with real-time data, it does not have practical reasons to increase the warning period, therefore it is kept at 30 minutes.

## 3.3 Novel method for anomaly detection in Lab42

The novelty lies in applying a mix of the two methods, that provides a more reliable short and long term anomaly detection tool, by creating a pipeline that starts with the data retrieval, and ends with the visualized reports of anomalies. Since the NAIA and LSTM-AE methods both rely on different and more diverse data sources, the challenge, and additional novelty is in using them in a way that identifies valid anomalies and degradation on less saturated and change sensitive data.

The proposed pipeline is described in Figure 1, that shows the four stages of the pipeline and the data flows between the elements of the architecture. All the data is stored in different tables of one database.

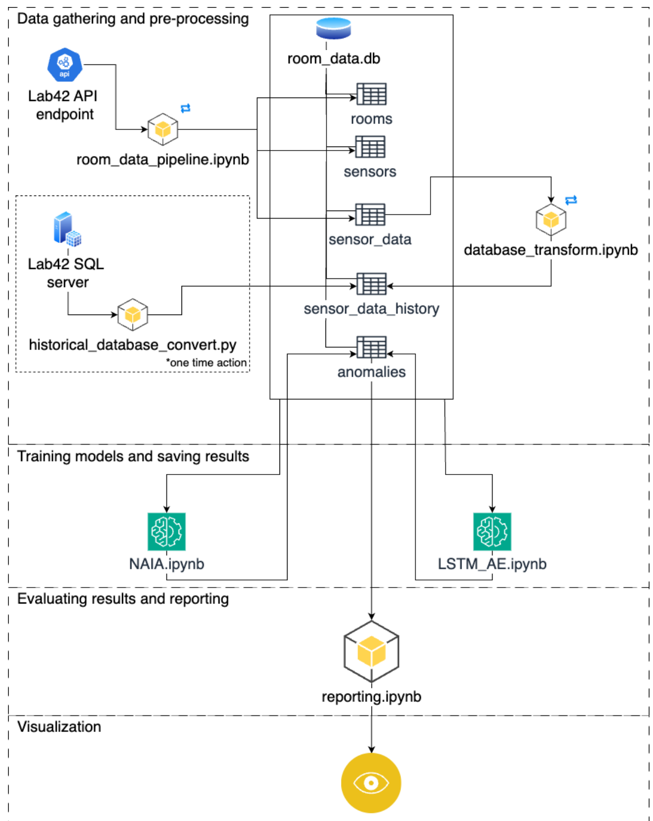


**Figure 1: Pipeline architecture**

Once the system is validated, as described in Section 3.4, Figure 1 provides the basis for the answer for the second research subquestion (SRQ2), on how can anomalies be detected.

*3.3.1 Data.* Data is gathered using two methods: to get the large amount of historical data, a one-time connection to Lab42's SQL server was opened, and the selected data was copied into the sensor_data_history table, grouped by the room_id and the timestamps, to minimize the database size and simplify access to the data by the models.

To get the recent data from the building, the room_data_pipeline script is run daily that connects to the API endpoint of the server in Lab42, and therefore keeps the data up-to-date. The new data is stored in the sensor_data table as it is received from the server, without any grouping. This is required to adhere to the limitations of the Lab42 server, while keeping the time duration of the data retrieval at minimum.

The contents of the sensor_data table are grouped and moved to the sensor_data_history table to keep the data uniform and processable by the models.

*3.3.2 Training models.* The NAIA and the LSTM_AE models are trained on the data, and the values marked anomalous by any of the models are saved in the database's anomalies table, with the additional information of which method deems it as an anomaly.

*3.3.3 Comparing results.* The results are analyzed and the anomalies reported by the different methods are compared to avoid reporting false positives or duplicated reports of detection.

## 3.4 Validation

The system is validated by feeding it with manually created anomalous data, and checking if the system captures the anomalies or not, serving as a confirmation that the system works as expected.

The validation starts with determining the baseline on a set of data, which shows what anomalies are detected in the data without any hindering. João Gama et al. in A Survey on Concept Drift Adaptation [8] has classified changes in the mean of data as sudden/abrupt, incremental, gradual, reoccurring concepts and outliers. An illustration of the data changes can be found in Appendix C, from the original publication of Gama et al. The experiments aim to replicate these changes both to showcase what kind of anomalies are detected, and validate that the system works as expected.

# 4 EXPERIMENTAL SETUP

An experimental setup is created for the use case of the Lab42 building, as described in the following subsections:

## 4.1 Data

While the answer to the first research sub-question (SRQ1), is clear after evaluating the results, this subsection hints at the answer of what sort of data is required to recognize anomalies. The system uses the temperature, air quality and lights (natural and artificial) data which is available for every minute. There are some gaps in the data for a number of rooms that can be marked as less accurate depending on the gap, or if the data for a specific sensor is missing altogether, it can be excluded from the proof of concept.

Due to the large amount of data available, the time period of one year was chosen for the historical data collection, from 01-05-2024, with continuous collection of near-real-time data.

Rooms without sufficient amount of data, such as missing a certain sensor reading altogether had to be excluded from the sample, which means the system does not detect any anomalies in those rooms. The brief statistics of the number of rooms with sufficient data for the system is described in Table 3.

| | |
|---|---|
| Number of rooms | 267 |
| Number of rooms that are missing at least one sensor data | 221 |
| Number of rooms with data from all sensors | 46 |

**Table 3: Available room data**

The rooms have different purposes, properties and usage patterns, that can affect the data patterns and the success rate of the anomaly detections. For example the air quality sensors are less indicative in open study spaces or lecture halls, while in study and meeting rooms the system can be expected to perform better. Table 4 shows the distribution of rooms that have data:

| Room function | Number of rooms with available data |
|---|---|
| Study room | 21 |
| First aid and lactation room | 1 |
| Open Study space | 3 |
| Lab | 3 |
| Student association room | 1 |
| Lecture hall | 1 |
| Meeting room | 16 |

**Table 4: Room function distribution with available data**

## 4.2 Database

The system requires the historical data to be available in a rapid, highly available structured database. Therefore the historical data is first retrieved from the API endpoint provided by the university.

The database has a simple structure as described in the following figure:

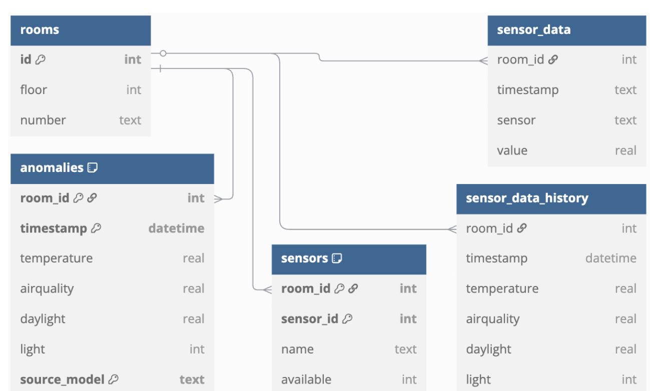


**Figure 2: SQL database structure**

The sensor data is stored in two different tables, one for the historical data, and one to store the new sensor readings. It is needed, since through the API endpoint the data is gathered in a different structure than what is more efficient and easier to process by the ML algorithms. The data from the "sensor_data" table is periodically transformed and transferred to the "sensor_data_history" table.

## 4.3 ML training

The machine learning algorithms are trained on the historical data to determine the baseline for the normal behavior of the system. The code is written in python and uses several scikit-learn and keras model and layer libraries. The code itself can be found in the GitHub repository.

## 4.4 Anomaly detection

The results of the different methods and models are compared to each other to avoid false positives (where a model shows a false anomaly), and validate the outputs.

## 4.5 Visualization

The results of the analysis are visualized in graphs as part of the reporting. These graphs are shown in the results section and the appendices of this study.

# 5 RESULTS

The proposed system has been applied on the historical and real-time dataset, that were retrieved through the API endpoint, as described in the previous sections. For validation purposes, the real-time data was injected with anomalies in different ways, that are described in the experiments sub-section, and the system was run again. These results are presented in this section.

## 5.1 Data

There are continuously gaps in the near real-time data, which in some occasions cause disruptions in training the models, where the main cause of the errors is that there was not enough available data in the previous 24 hours for analysis.

The lack of an entire sensor data for a room prevents the system to build the necessary data frames, and therefore the anomaly detection becomes impossible. As Table 3 shows, only about 17% of rooms in the LAb42 building have sufficient data, therefore only these 46 rooms with sufficient data were included in the system.

## 5.2 Models

The amount of distinct anomalous hours detected by the NAIA method (as of 29th of May 2025) is shown on Figure 3, and the LSTM-AE method returns anomalies as shown in the following figure 4.

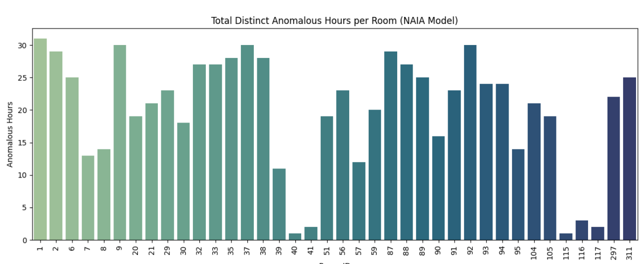


**Figure 3: Anomalous hours detected by NAIA**

Since the LSTM-AE method has only been trained on the past 180 days of data, the NAIA method is also showing anomalies from that

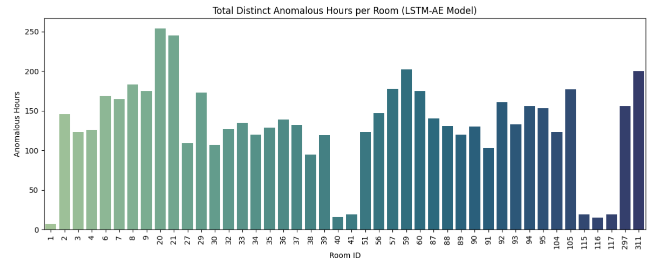


**Figure 4: Anomalous hours detected by LSTM-AE**

time frame to ensure the same conditions, and provide a comparable result in the amount of anomalies found by each method. Both of them show distinct anomalous hours for each room_id.

## 5.3 Reported anomalies

The reported anomalies are investigated and the results of the methods are compared to report only anomalies detected by both methods. Due to the construction of the NAIA method, since it uses daily patterns to check for anomalies in the historical data, hourly comparison is not possible with the LSTM-AE method, only the number of distinct anomalies. The daily comparison however shows that there are days where both methods reported anomalies. The number of distinctive anomalous days detected by both methods are shown on Figure 5:

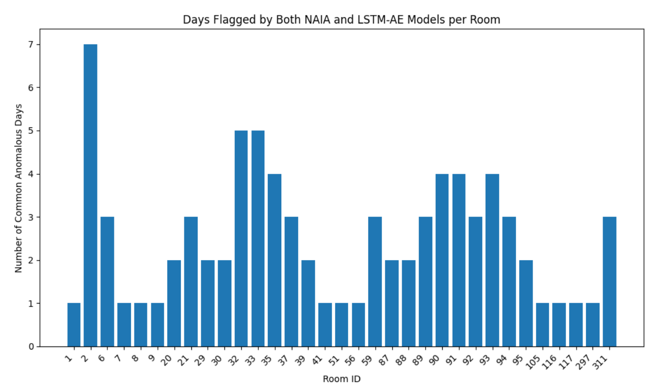


**Figure 5: Anomalous days detected by both methods**

The number of distinct anomalous hours detected by either of the methods is shown on Figure 6. This shows the most important product of the system: a chart that shows all of the detected distinct anomalies, which the experiments confirm to be highly accurate. Important to note, that the resolution of the anomalies are hourly, which does not provide information on its intensity. Due to the large amount of data, it is necessary to group the anomalies by the hour, since it can be assumed that anomalies are correlated – especially in the case of anomalies detected by the LSTM-AE method as that uses sequences of data. This grouping provides additional information, as it shows the distinct anomalous events, spanning between 1 and 60 minutes, instead of a simple sum of the anomalous minutes, that does not take their proximity into account.

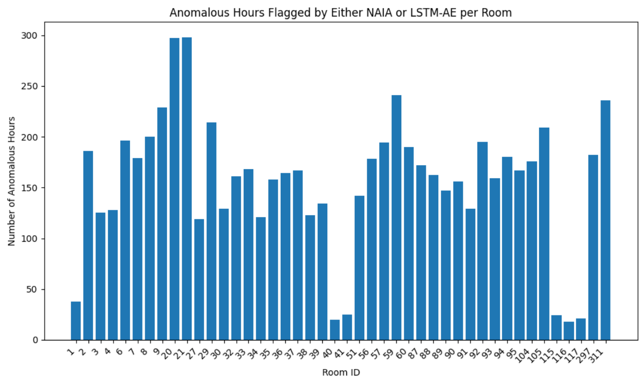


**Figure 6: Anomalous hours detected by either methods**

Since the data gathered from the Lab42 building is unlabeled, the accuracy, recall and precision of these results cannot be determined, as it is unknown how many anomalies are in the data, and what percentage of these were detected. That is the reason for the experiments described in the following section.

## 5.4 Experiments

Several experiments have been done to validate that the system can detect anomalies as expected. Different methods of data manipulation were applied to a real data-set of one day. The validation set contains 90 days of data for training the models, and a manipulated set of one day of data to test the anomaly detection. Each of the experiments are based on a different data manipulation principle described in the following subsections.

For each experiment, every decision on which rooms, sensors or timestamps to modify was entirely randomized to ensure that the selection of rooms, sensors and types were unbiased. Only rooms with insufficient data to train the models were removed.

*5.4.1 Sudden change in values for any sensors.* The data was manipulated, so at a certain point of time, there was a sudden increase in the selected values (by 10 units). After the first modified value, all the subsequent values were offset by the same value. This manipulation was repeated for different rooms with different sensors.

*5.4.2 Incremental change in values for any sensors.* The data was manipulated the following way: At a random point in time one of the sensor values starts to slowly increase over the course of three hours, by equal steps. After the three hours increase period, the subsequent values were changed by the total offset. This manipulation was repeated for different rooms with different sensors. The maximum offset for the values were 10 degree Celsius for the temperature, 200 $CO_2$ppm for air quality and 50 lumen for daylight and light.

*5.4.3 Gradual change in values for any sensors.* The data was manipulated, so there are random periods of time when the values are changed by a certain sensor value, lasting between 30 and 90 minutes, and after the last change of the day, the values plateau at the increased value. The offsets were 10 degree Celsius for the temperature, 200 $CO_2$ppm for air quality and 50 lumen for daylight and light.

*5.4.4 Reoccurring changes in values for any sensors.* The data was manipulated so there are reoccurring anomalies in the data that affect more than one sensor, but these changes are not lasting longer than a few minutes, after the change occurs, the data remains unedited.

*5.4.5 Outliers in values for any sensors.* Random values in the database were changed to extreme values, which would never occur normally (temperature and air quality at -1000, and light and daylight values at 9999).

*5.4.6 Experiment results.* The results of the experiments are summarized in Table 5.

False positives are instances that were detected after the data injection as anomalies, but are not in the baseline or the injected anomalies.

Precision quantifies the proportion of detected anomalies that are indeed truly an anomaly. It is calculated as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

In the "Overall precision of the system", both methods were used with an 'OR' relation, to measure true and false positives, however, since the NAIA method had 0 false positives throughout the experiments, it also represents the proportion of the false positives detected by the LSTM-AE method against all of the found anomalies.

The false negatives are the anomalous hour instances, that are either in the injected anomalies, or detected in the baseline, but were not identified after running the system on the modified data. These are also referenced as "Undetected anomalous hours" in the results.

The recall metric shows "the percentage of positive instances correctly classified, and indicates how often a classifier misses a positive prediction" [16], and it is calculated as the following:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

These experiment results can be implied on the original data, meaning that similar types of anomalies (sudden, incremental, gradual and recurring change, and outliers) were detected with similar precision and recall.

| Experiment / Metric | Sudden change (D.2) | Incremental change (D.3) | Gradual change (D.4) | Reoccurring change (D.5) | Outliers injection (D.6) |
|---|---|---|---|---|---|
| Baseline anomalous hours detected by NAIA **or** LSTM-AE | 143 | 154 | 154 | 120 | 137 |
| Anomalous hours injected | 115 | 134 | 161 | 74 | 10 |
| Injected anomalies detected by NAIA | 2 (1.74%) | 6 (4.48%) | 3 (1.86%) | 2 (2.70%) | 0 (0.00%) |
| Injected anomalies detected by LSTM-AE | 110 (95.65%) | 118 (88.06%) | 157 (97.52%) | 66 (89.19%) | 6 (60.00%) |
| Injected anomalies detected by NAIA **and** LSTM-AE | 1 (0.87%) | 4 (2.99%) | 2 (1.24%) | 1 (1.35%) | 0 (0.00%) |
| Injected anomalies detected by NAIA **or** LSTM-AE | 111 (96.52%) | 120 (89.55%) | 158 (98.14%) | 67 (90.54%) | 6 (60.00%) |
| False positive anomalous hours by NAIA | 0 (0.00%) | 0 (0.00%) | 0 (0.00%) | 0 (0.00%) | 0 (0.00%) |
| False positive anomalous hours by LSTM-AE | 0 (0.00%) | 1 (0.65%) | 2 (1.12%) | 13 (9.49%) | 5 (4.31%) |
| Overall false positive hours by either method | 0 (0.00%) | 1 (0.57%) | 2 (1.00%) | 13 (8.33%) | 5 (3.68%) |
| Overall precision of the system | 100% | 99.17% | 98.75% | 83.75% | 54.55% |
| Undetected anomalous hours (false negatives) | 4 (3.48%) | 14 (10.45%) | 3 (1.86%) | 7 (9.46%) | 4 (40.00%) |
| Recall of the system (on the injected data) | 96.52% | 89.55% | 98.14% | 90.54% | 60.00% |

**Table 5: Experiment results**

# 6 DISCUSSION

To answer the research questions and sub-questions in detail, the system needs to be analyzed from a few different aspects. These are the anomaly detection results of the models, the specificity of the data the system relies on, and the validity of the results.

## 6.1 Models

The models have proved to find different type of anomalies, which, contrary to the initial assumption, complement each other, rather than confirm the detection of anomalies.

*6.1.1 The NAIA method.* mostly identifies the outliers and inconsistencies in the data patterns, but proved insufficient at identifying new anomalous patterns. In the study of Izaskun et al. [14], the difference in the dataset was the stronger causational relationship between the different sensors, since the authors worked on power consumption data, that is less prone to external factors compared to the data available for the Lab42, where there is a delay in the different sensor changes. As an example, in the use case of Izaskun et al. [14], if the power consumption changes in the auxiliary lines, there is an immediate change in the HVAC sensor system, while in the Lab42, if the light values are changed, which suggests presence in the room, the air quality readings will only change slowly over time (during average use). However, this method is great in supplementing the LSTM-AE method, as the false positive detections made by the NAIA method is 0 across all experiments.

The results of the outlier detection experiments show that the NAIA method did not detect any outliers. That is because during the definition of the nominal performance phase of the model, the outliers are filtered out, as the anomaly is not in the behavior of the system, but in the reading of the sensor, which the method is not aiming to detect, but filter. The results do not confirm if the filtering was successful, as it does not exclude the possibility of leaving the reading undetected, but confirms that the method does not identify it as an anomaly in the building.

*6.1.2 The LSTM-AE method.* detects the majority of the anomalies in the system, as the experiment results show. Apart from the outlier injection experiment, for every test, the method detected more than 88% of the injected anomalies. The model performed extremely well, detecting more than 98% of the gradual change injections, and similarly good (95.65%) of the sudden change injections. These were the experiments that injected the most out-of-pattern data, that the machine learning model can identify easier. The gradual change experiment also revealed, that the model does not learn the anomalous data to an extent, where the non-anomalous data is flagged, resulting in false positives – only 1.12% of the anomalies detected by the model proved to be false.

## 6.2 Data

The data available for the Lab42 building proved to be sufficient for anomaly detection, in the cases of the rooms where all four sensor readings were available. Based on the experiments of anomalous data injection the anomalies can be recognized with a 90% certainty, that is discussed in a later section.

During the data gathering phase of the pipeline there is no need for extensive data cleaning or modifications, therefore in that phase the system can be applied on any building providing the four type of sensor data in similar resolution. The only case specific part of the data retrieval is in connecting to the API, that also determines the initial database structure and the possibilities regarding retrieval and processing speed.

The important aspect of the data is that it needs to be available per room, since there can be significant differences between different types of rooms. The models are trained per room to identify anomalies., therefore they do not affect each other.

## 6.3 Validity

The goal of the experiments is to confirm that the models are valid, that they indeed find the anomalies. As the results show, the different experiments yielded similar results (apart from the outlier injection), however, there are certain patterns that are easier to identify, as discussed in the previous subsections.

## 6.4 Limitations

The main limitation of the system stems in the data availability. As the case of the university building shows, where out of the total 267 rooms, only 46 rooms had all four types of sensor data, and some of those had insufficient daily data, due to gaps in the data stream. While it does not affect the results for the rooms that has the data, in scope of the building, it can be considered as a great limitation.

A limitation in anomaly detection is that the outliers – the extreme and invalid data points – are not properly detected, only 60% are caught by the LSTM-AE method. However, this can be improved, as the NAIA method had successfully disregarded those. As a future work, a feature can be added to the system to handle these, which is discussed in the next section of this study.
As the system solely focuses on the detection of the anomalies in the data, it does not provide any insights on the underlying causes. Therefore, in the current state of the system it cannot determine if the anomaly occurs due to a technical issue, or human behavior.

An additional limitation is that the results are shown in number of anomalous hours, which omits the intensity property, therefore it is not clear if the anomaly is an outlier in the hour, or every minute of the hour is considered anomalous. Since this is a limitation of the representation of the data and how the system simplifies the data for better understanding of the results, it is addressed in the future work, in Section 7.1.4, how it could be improved, along with identifying the underlying cause of the anomaly, that falls outside of the scope of the research.

## 6.5 Comparison to the state of the art

Baldán and García-Gil [5] has created a benchmark of the current methods of anomaly detection, which takes the datasets from the UCR Time Series Anomaly Archive. The difference however is that the datasets used in their study are containing labeled data, and only one anomaly per dataset which they aim to identify. Their results show, that the best method they investigated (DeepSVDD) identified 47.2% of the anomalies, which is a much lower score than what the experiments in this study showed. An important difference in the datasets however is the amount of data-points available: their datasets varied between 6684 and 900 000 data points, while the historical data available in this research were higher, varying between 196 962 and 1 710 590 data points, as shown in Appendix E, which can be the cause of the much higher accuracy.

Sater and Hamza [16] created an LSTM based algorithm that uses similar datasets, with more sensor types, that resulted in overall 88-90% of correct identification of anomalies, with 4-9% of false positives, compared to this study, with correct detections varying between 89.55% and 98.14%, with 0-8.33% false positives (excluding the outlier injection experiment). The outlier injection experiment is excluded from these results, as they can be sorted out with the NAIA method, and removed from the dataset, with minor changes in the system.

# 7 CONCLUSION

The research primarily aimed to answer the question: "How can a cyber-physical system of UvA's Lab42 building be used to monitor and report anomalies in the building?" (RQ1). The results confirm that using the combination of the NAIA and LSTM-AE methods explored in the related work and methodology section can provide a highly accurate anomaly detection tool that monitors and reports anomalies.

The first sub-question was regarding the data: "What type of data is required to confidently recognize anomalies in the building?" (SRQ1). Section 4.1 describes that the amount of data is sufficient in the case of 46 rooms in the building, therefore the answer to the sub-question is: the four different types of sensors are sufficient, if enough data is available for the LSTM-AE method to build 30 data point sequences in hourly resolution.

The second sub-question was "How can anomalies be detected with great confidence, relying solely on the data provided by the HVAC sensors?" (SRQ2). The answer is visually represented in Figure 1, and consists of four phases: data gathering and pre-processing, training models and saving results, evaluating results and reporting, and the visualization of the results.

The secondary question "What building specific configuration is built into the proposed system?" (RQ2), aimed to understand if the system can be easily used for other buildings, or would it need a lot of modification to do so. Since the system relies on the data, gathered in a specific way, set by the API endpoint, some modification is inherently necessary, however, once the data is collected – assuming it is the same or very similar type of sensor readings and resolution – the only specification is the labeling requirement for the LSTM method, that is described in Table 2.

The third sub-question, clarifying the secondary question was "What building specific data cleaning, manipulation or organization is necessary for the anomaly detection system, that could raise difficulties in applying it to other buildings?" (SRQ3). The answer has been partially discussed in regards of the labeling. While it is not building specific, it is also important to mention, that in order to run the system, it must be confirmed, that all sensors, with the correct resolution are available, and it must be organized in the structure described in section 4.2.

The fourth sub-question was "What sort of data quality aspects can prevent anomaly detection in the proposed system?" (SRQ4). It has already been mentioned that the insufficient sensor data or resolution prevents the anomaly detection.

The research addressed the gap in anomaly detection, and it can be the basis of an information system that analyzes the effects and consequences of the anomalies, which is shortly discussed in Sub-section 7.1.4.

## 7.1 Future work

There are several ways the research can continue, and these are outlined in the following subsections.

*7.1.1 Outlier detection.* First of all, the system can be enhanced, by focusing on the experiment results, especially on the outlier detection. This can be done by creating an additional layer in the system, where the outliers are detected, removed from the dataset and reported as sensor failure.

*7.1.2 Additional validation on the benchmark datasets.* Secondly, an extra validation would be a more thorough comparison to the referenced benchmarks, by running the system on the datasets the authors used. This would confirm the generalizability, and also provide a better comparison of accuracies.

*7.1.3 Enlarging the dataset.* The historical dataset for the system was limited for about a year of data, which was enough to learn seasonal effects, but still not too large to require extra computational resources. Enlarging the dataset can lead to even better results and more precise anomaly detection.

*7.1.4 Failure detection.* In its current state, the system can detect anomalies with great confidence, but it only generates information. To move upwards in the data-information-knowledge-wisdom pyramid [15], the root cause of the anomalies must be understood, that would make the system to be able to identify failure from other circumstances that would justify the out-of-pattern behaviour, such as events. The most straightforward way would be to add more data, such as presence information, which can be derived from the existing sensors' behaviour or by installing new sensors. The current resolution of the anomaly detection is hourly, as mentioned in the previous sections, which can be increased to minutely, that would help the system to identify the intensity of the anomalies, and provide useful information in understanding the root cause, whether it is an outlier or a reoccurring issue.

*7.1.5 Visualization.* In its current state, the visualization of the results only consists of graphs representing the data output of the system. This can be enhanced, by connecting it to a visual, two or three dimensional digital representation of the building, where the past anomalous hours, and any current anomalies are shown. Since the anomalies are stored in a separated database table (as shown on Figure 1), these additional visualizations can be easily implemented without changing the current system.

## REFERENCES


[1] Puneet Agarwal and Scot McNeill. 2019. Real-Time Cleaning of Time-Series Data for a Floating System Digital Twin. OnePetro. https://doi.org/10.4043/29642-MS

[2] Rasim Alguliyev, Yadigar Imamverdiyev, and Lyudmila Sukhostat. 2018. Cyber-physical systems and their security issues. *Computers in Industry* 100 (Sept. 2018), 212–223. https://doi.org/10.1016/j.compind.2018.04.017

[3] Arva Arsiwala, Faris Elghaish, and Mohammed Zoher. 2023. Digital twin with Machine learning for predictive monitoring of CO2 equivalent from existing buildings. *Energy and Buildings* 284 (April 2023), 112851. https://doi.org/10.1016/j.enbuild.2023.112851

[4] Mohsen Attaran and Bilge Gokhan Celik. 2023. Digital Twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal* 6 (March 2023), 100165. https://doi.org/10.1016/j.dajour.2023.100165

[5] Francisco J. Baldán and Diego García-Gil. 2025. Benchmarking Anomaly Detection Methods: Insights From the UCR Time Series Anomaly Archive. *Expert Systems* 42, 2 (2025), e13767. https://doi.org/10.1111/exsy.13767 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.13767.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *Comput. Surveys* 41, 3 (July 2009), 1–58. https://doi.org/10.1145/1541880.1541882

[7] Md Shezad Dihan, Anwar Islam Akash, Zinat Tasneem, Prangon Das, Sajal Kumar Das, Md Robiul Islam, Md Manirul Islam, Faisal R. Badal, Md Firoj Ali, Md Hafiz Ahamed, Sarafat Hussain Abhi, Subrata Kumar Sarker, and Md Mehedi Hasan. 2024. Digital twin: Data exploration, architecture, implementation and future. *Heliyon* 10, 5 (March 2024). https://doi.org/10.1016/j.heliyon.2024.e26503 Publisher: Elsevier.

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46, 4 (March 2014), 44:1–44:37. https://doi.org/10.1145/2523813

[9] Michael Grieves and John Vickers. 2017. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, Franz-Josef Kahlen, Shannon Flumerfelt, and Anabela Alves (Eds.). Springer International Publishing, Cham, 85–113. https://doi.org/10.1007/978-3-319-38756-7_4

[10] Wei Hu, Xin Wang, Khery Tan, and Yiyu Cai. 2023. Digital twin-enhanced predictive maintenance for indoor climate: A parallel LSTM-autoencoder failure prediction approach. *Energy and Buildings* 301 (Dec. 2023), 113738. https://doi.org/10.1016/j.enbuild.2023.113738

[11] Edward A. Lee. 2015. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors* 15, 3 (March 2015), 4837–4869. https://doi.org/10.3390/s150304837 Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[12] Scott D. Lowther, Sani Dimitroulopoulou, Kerry Foxall, Clive Shrubsole, Emily Cheek, Britta Gadeberg, and Ovnair Sepai. 2021. Low Level Carbon Dioxide Indoors—A Pollution Indicator or a Pollutant? A Health-Based Perspective. *Environments* 8, 11 (Nov. 2021), 125. https://doi.org/10.3390/environments8110125 Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[13] Qiuchen Lu, Xiang Xie, Ajith Kumar Parlikad, and Jennifer Mary Schooling. 2020. Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. *Automation in Construction* 118 (Oct. 2020), 103277. https://doi.org/10.1016/j.autcon.2020.103277

[14] Izaskun Mendia, Sergio Gil-Lopez, Iñaki Grau, and Javier Del Ser. 2024. A novel approach for the detection of anomalous energy consumption patterns in industrial cyber-physical systems. *Expert Systems* 41, 2 (2024), e12959. https://doi.org/10.1111/exsy.12959 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12959.

[15] Jennifer Rowley. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* 33, 2 (April 2007), 163–180. https://doi.org/10.1177/0165551506070706 Publisher: SAGE Publications Ltd.

[16] Raed Abdel Sater and A. Ben Hamza. 2021. A Federated Learning Approach to Anomaly Detection in Smart Buildings. https://doi.org/10.48550/arXiv.2010.10293 arXiv:2010.10293 [cs].

[17] Maulshree Singh, Evert Fuenmayor, Eoin P. Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. 2021. Digital Twin: Origin to Future. *Applied System Innovation* 4, 2 (June 2021), 36. https://doi.org/10.3390/asi4020036 Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[18] Richard J. Somers, James A. Douthwaite, David J. Wagg, Neil Walkinshaw, and Robert M. Hierons. 2023. Digital-twin-based testing for cyber–physical systems: A systematic literature review. *Information and Software Technology* 156 (April

2023), 107145. https://doi.org/10.1016/j.infsof.2022.107145

[19] Ashani Wickramasinghe, Saman Muthukumarana, Matt Schaubroeck, and Surajith N. Wanasundara. 2023. An anomaly detection method for identifying locations with abnormal behavior of temperature in school buildings. *Scientific Reports* 13, 1 (Dec. 2023), 22930. https://doi.org/10.1038/s41598-023-49903-7 Publisher: Nature Publishing Group.

[20] Qinghua Xu, Shaukat Ali, and Tao Yue. 2021. Digital Twin-based Anomaly Detection in Cyber-physical Systems. In *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*. 205–216. https://doi.org/10.1109/ICST49551.2021.00031 ISSN: 2159-4848.

[21] Meng Zhang, Fei Tao, Biqing Huang, Lihui Wang, Nabil Anwer, and A. Y. C. Nee. 2022. Digital twin data: methods and key technologies | Digital Twin. (2022). https://doi.org/10.12688/digitaltwin.17467.2

# Appendix A ETHICAL CONSIDERATIONS

As the data was gathered by the University of Amsterdam, any legal compliancy has to be met on their side. From an ethical perspective, the data is fully anonymous, there is no personal identification referencing any person, therefore there is no personal data handled in the system in its current form. The rooms that have sufficient data are all commonly used rooms as Table 4 shows, therefore they cannot be tied to one person either. The learned patterns are solely used for the machine learning algorithm to identify deviations, these patterns are not used for any type of identification that could be misused in any ways. While the data does not raise personal privacy issues, it is somewhat sensitive for the building management, and is therefore not sahred with the code in the Github repository.

# Appendix B USE OF GENERATIVE AI

Through the course of the thesis and the development of the proposed system, generative AI was used in support of the coding. ChatGPT and Co-Pilot was used in troubleshooting, during the ideation phase of building the structure of the code and for finding python libraries to use for the machine learning models and creating the SQL scripts for the data retrieval. Prompts such as "how to modify the following code to run it for every room_id in the database" and "write the sql script that lists all distinct room_ids in the anomalies table", etc. were used in creating chunks of code, especially in repetitive processes. The troubleshooting prompts were generally phrased as "help me understand how to fix the following error message".

Generative AI was never used for any data generation or modification neither in the historical or real-time data retrieval, nor during the experiments to validate the result.

# Appendix C GAMA ET AL. - PATTERNS OF CHANGES OVER TIME

Figure 7 was copied from "A Survey on Concept Drift Adaptation" by Gama et al. [8]

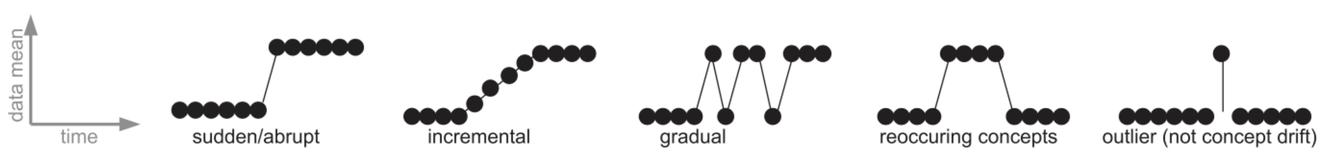


**Figure 7: Gama et al. (2014) - Patterns of changes over time**

# Appendix D EXPERIMENTS RESULTS

## D.1 Baseline

The anomaly detection was performed on the unhindered data to set the baseline and determine if any anomaly was detected for that day. The following charts show the result of the anomaly detection to set the baseline:

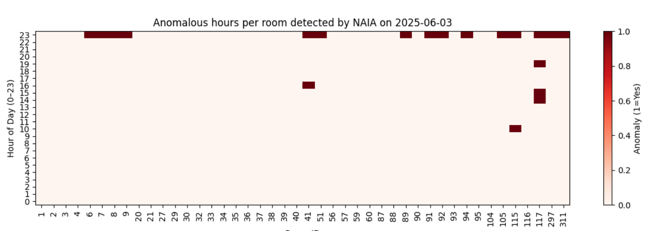


**Figure 8: Anomalies detected by NAIA**

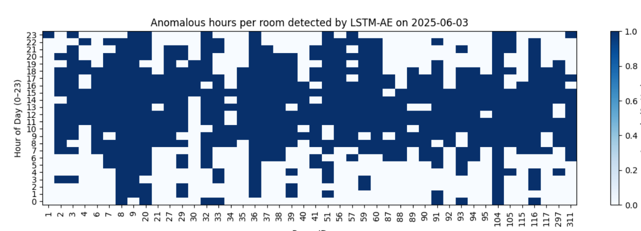


**Figure 9: Anomalies detected by LSTM-AE**

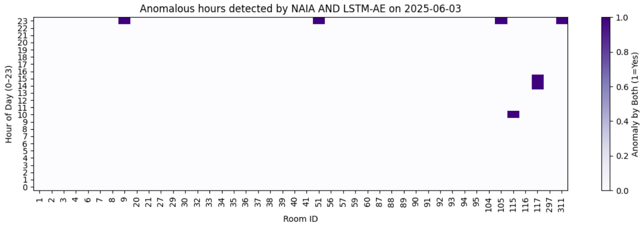


**Figure 10: Anomalies detected by both models**

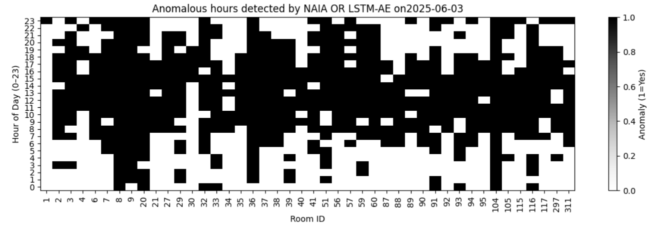


**Figure 11: Anomalies detected by either model**

## D.2 Sudden change experiment

For the sudden change experiment, the data was manipulated, so at a certain point of time, there was a sudden increase in the selected values (by 10). After the first modified value, all the subsequent values were offset by the same value. This manipulation was repeated for different rooms with different sensors.
The results of the experiment are summarized on the figures below.

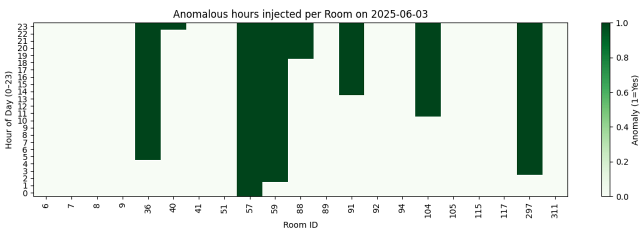


**Figure 12: Injected anomalies**

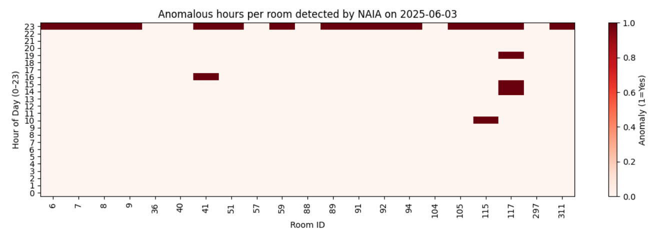


**Figure 13: Injected anomalies detected by NAIA**

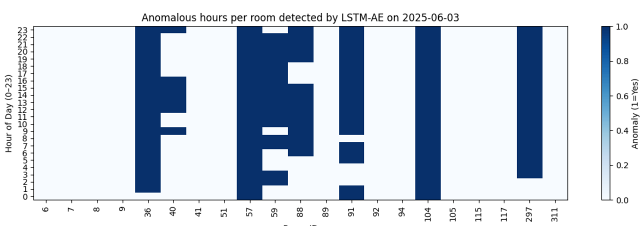


**Figure 14: Injected anomalies detected by LSTM-AE**

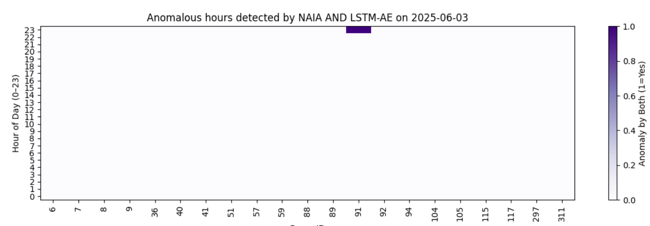


**Figure 15: Injected anomalies detected by both models**

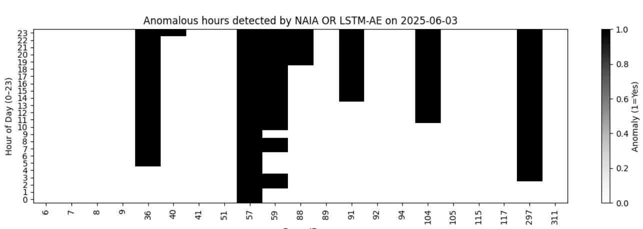


**Figure 16: Injected anomalies detected by either model**

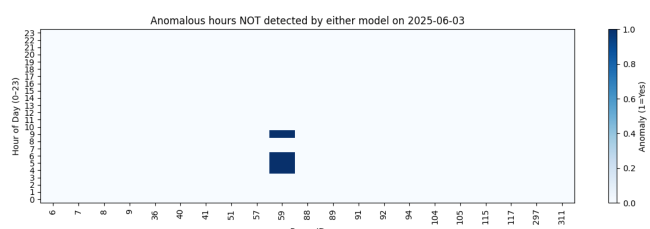


**Figure 17: Injected anomalies undetected by both models**

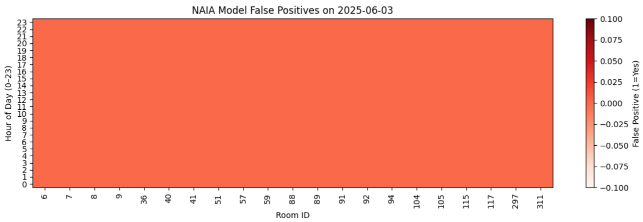


**Figure 18: False positive anomalies detected by NAIA**

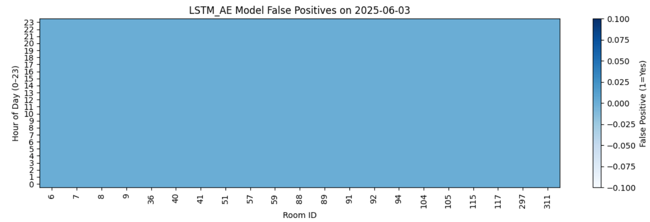


**Figure 19: False positive anomalies detected by LSTM-AE**

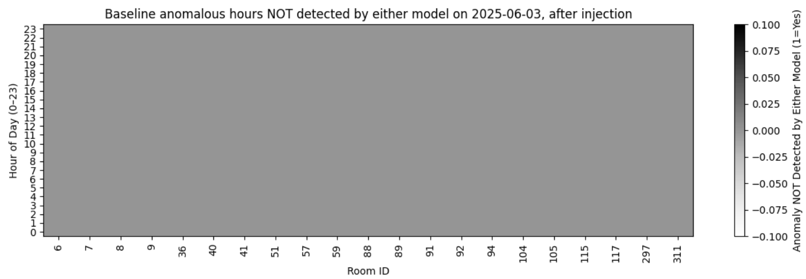


**Figure 20: Baseline anomalies not detected after injection**

## D.3 Incremental change experiment

For the incremental change experiment, the data was manipulated the following way: At a random point in time one of the sensor values starts to slowly increase over the course of three hours, by equal steps. After the three hours increase period, the subsequent values were changed by the total offset. This manipulation was repeated for different rooms with different sensors. The maximum offset for the values were the following:

- temperature: 10 degree Celsius
- air quality: 200 $CO_2$ppm
- light and daylight: 50 lumen

The results of the experiment are summarized on the figures below.

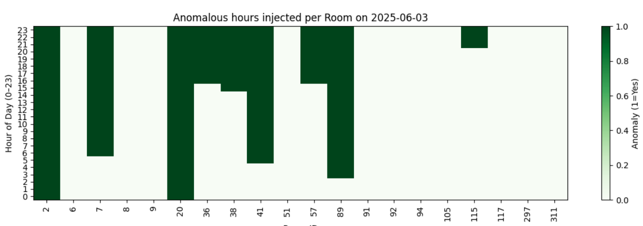


**Figure 21: Injected anomalies**

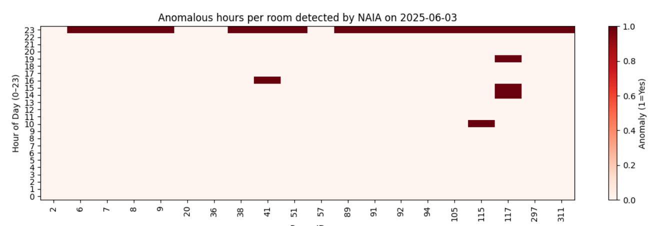


**Figure 22: Injected anomalies detected by NAIA**

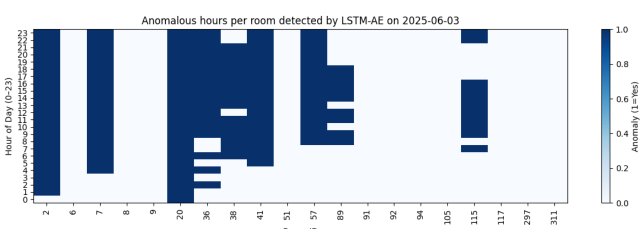


**Figure 23: Injected anomalies detected by LSTM-AE**

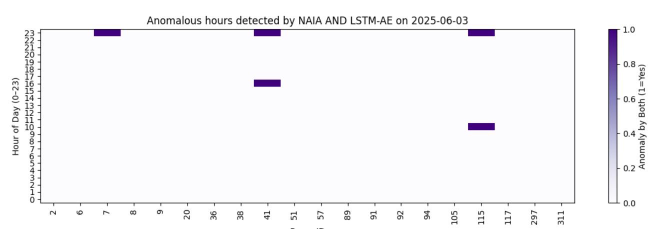


**Figure 24: Injected anomalies detected by both models**

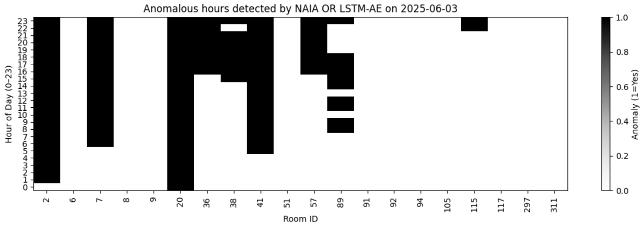


**Figure 25: Injected anomalies detected by either model**

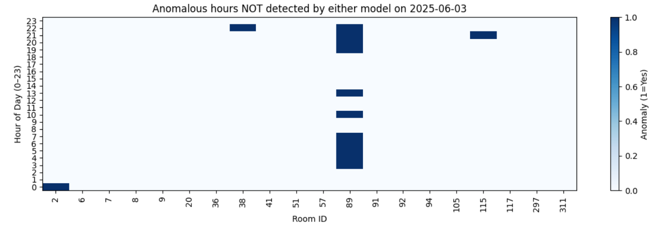


**Figure 26: Injected anomalies undetected by both models**

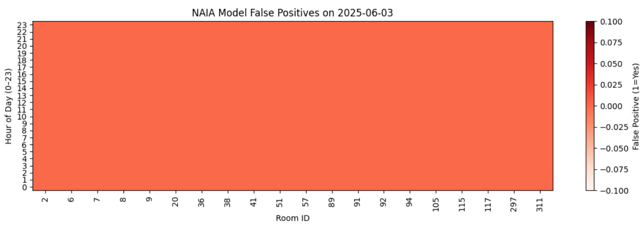


**Figure 27: False positive anomalies detected by NAIA**

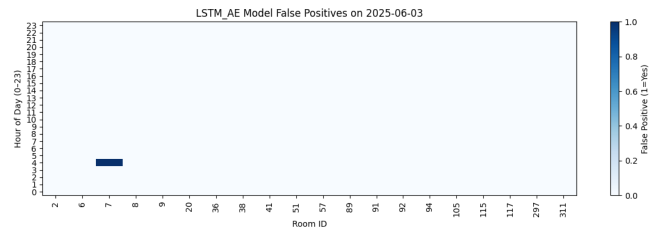


**Figure 28: False positive anomalies detected by LSTM-AE**

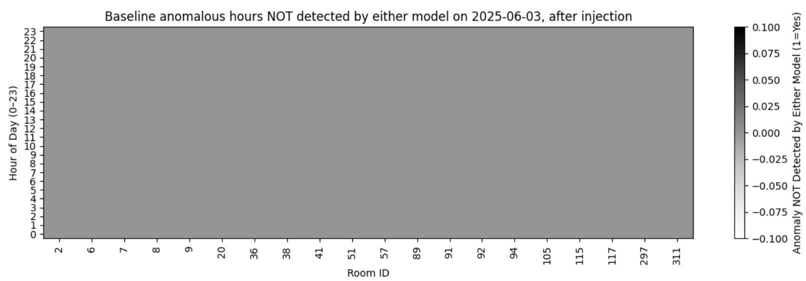


**Figure 29: Baseline anomalies not detected after injection**

## D.4 Gradual change experiment

For the gradual change experiment, the data was manipulated, so there are random periods of time when the values are changed by a certain sensor value, lasting between 30 and 90 minutes, and after the last change of the day, the values plateau at the increased value. The offsets were 10 degree Celsius for the temperature, 200 $CO_2$ppm for air quality and 50 lumen for daylight and light.
The results of the experiment are summarized on the figures below.

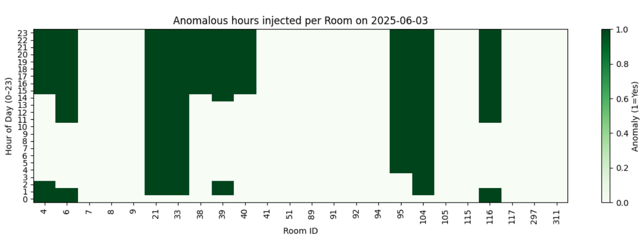


**Figure 30: Injected anomalies**

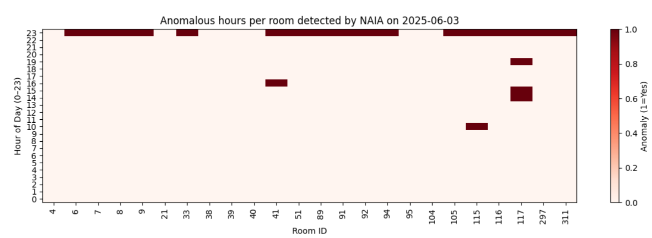


**Figure 31: Injected anomalies detected by NAIA**

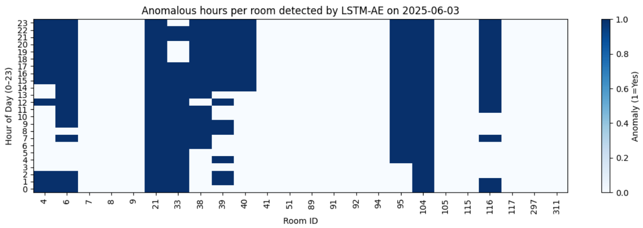


**Figure 32: Injected anomalies detected by LSTM-AE**

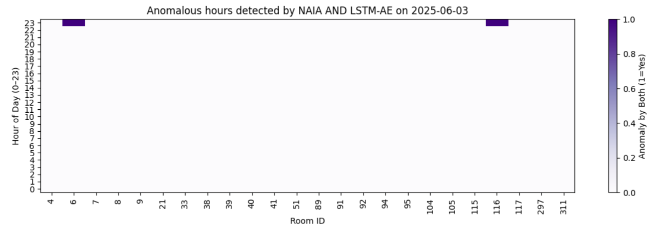


**Figure 33: Injected anomalies detected by both models**

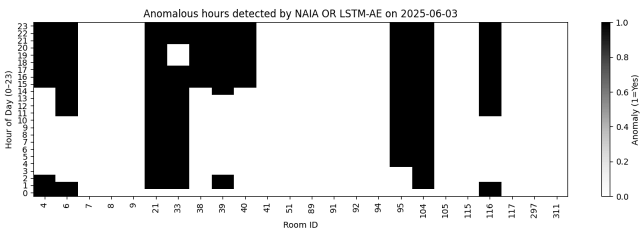


**Figure 34: Injected anomalies detected by either model**

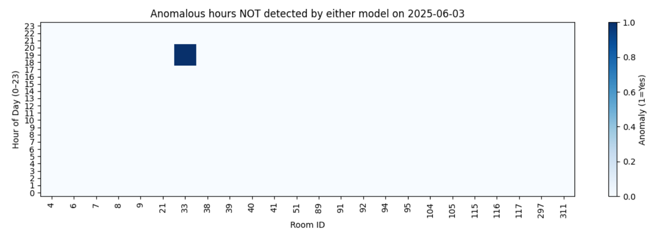


**Figure 35: Injected anomalies undetected by both models**

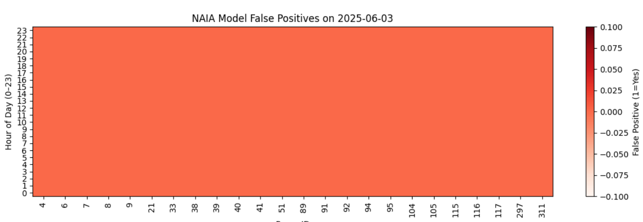


**Figure 36: False positive anomalies detected by NAIA**

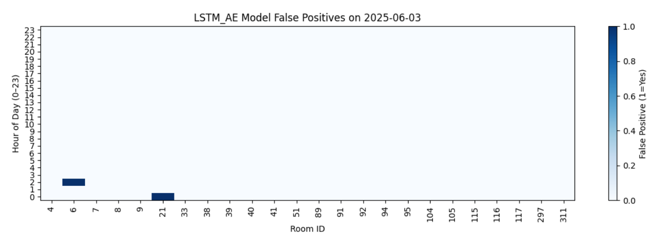


**Figure 37: False positive anomalies detected by LSTM-AE**

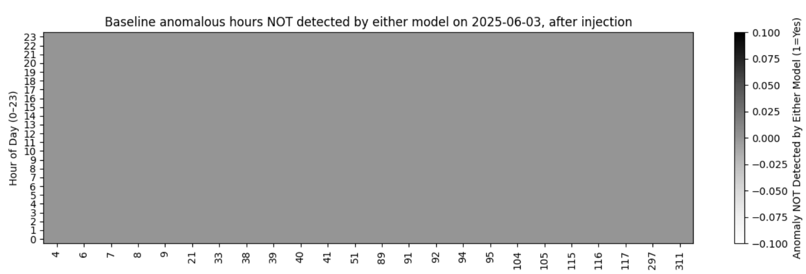


**Figure 38: Baseline anomalies not detected after injection**

## D.5 Reoccurring change experiment

For the reoccurring change experiment the data was manipulated so there are reoccurring anomalies in the data that affect more than one sensors, but these changes are not lasting longer than a few minutes, after the change occurs, the data remains unedited.
The results of the experiment are summarized on the figures below.

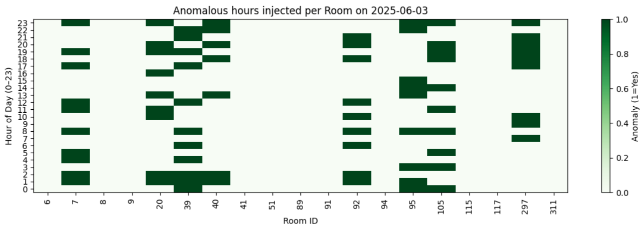


**Figure 39: Injected anomalies**

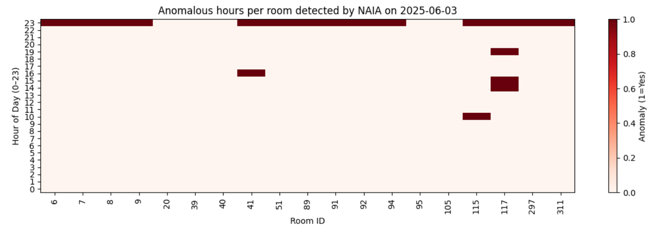


**Figure 40: Injected anomalies detected by NAIA**

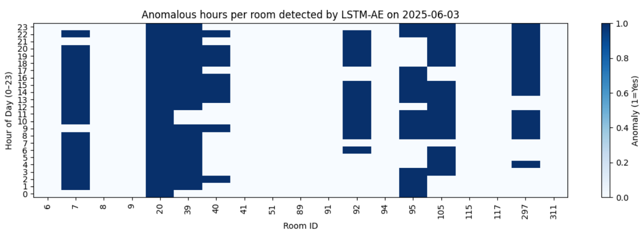


**Figure 41: Injected anomalies detected by LSTM-AE**

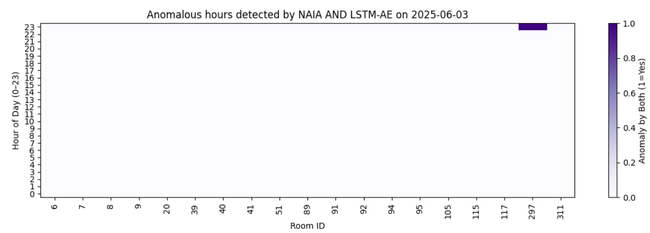


**Figure 42: Injected anomalies detected by both models**

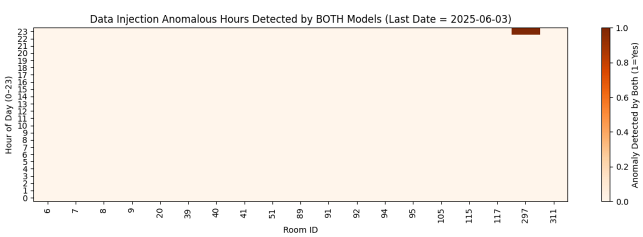


**Figure 43: Injected anomalies detected by either model**

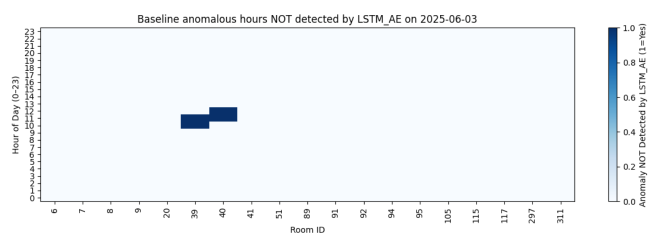


**Figure 44: Injected anomalies undetected by both models**

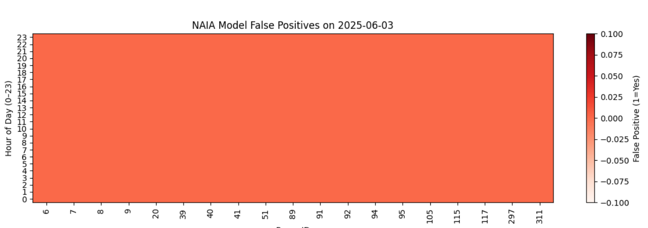


**Figure 45: False positive anomalies detected by NAIA**

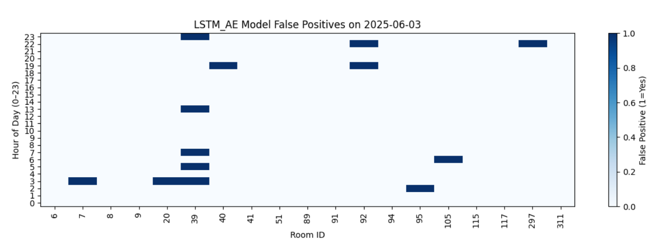


**Figure 46: False positive anomalies detected by LSTM-AE**

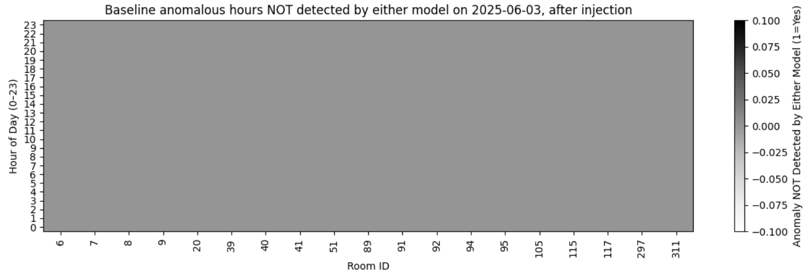


**Figure 47: Baseline anomalies not detected after injection**

## D.6 Outliers experiment

For the outliers experiment, random values in the database were changed to extreme values, which would never occur normally (temperature and air quality at -1000, and light and daylight values at 9999).
The results of the experiment are summarized on the figures below.

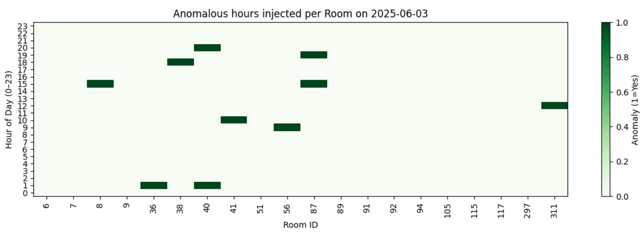


**Figure 48: Injected anomalies**

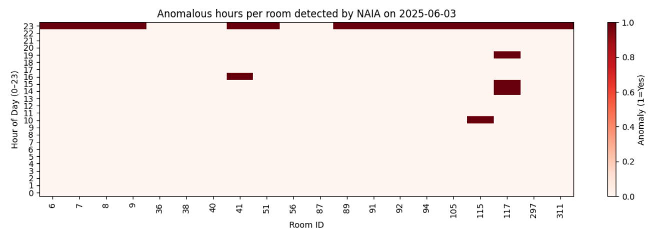


**Figure 49: Injected anomalies detected by NAIA**

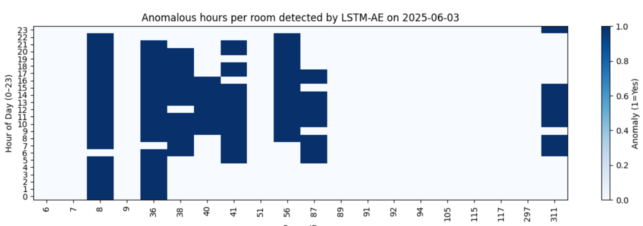


**Figure 50: Injected anomalies detected by LSTM-AE**

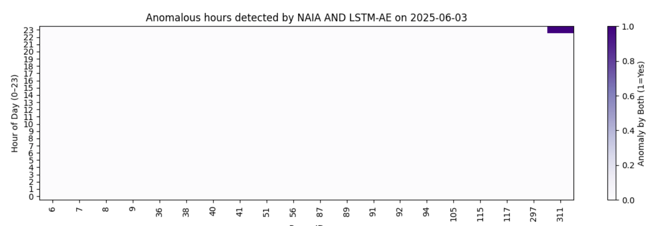


**Figure 51: Injected anomalies detected by both models**

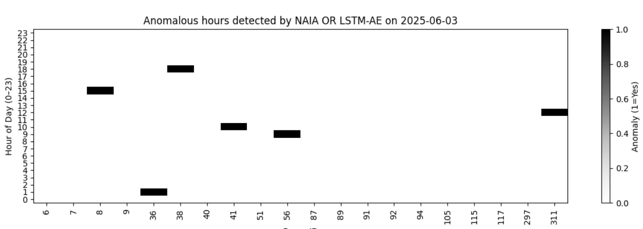


**Figure 52: Injected anomalies detected by either model**

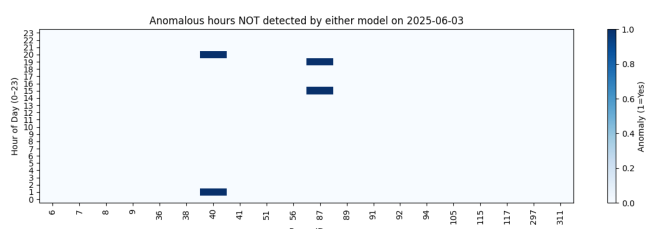


**Figure 53: Injected anomalies undetected by both models**

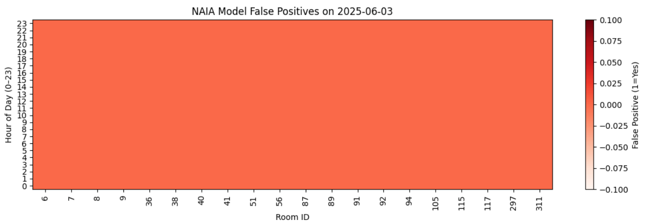


Figure 54: False positive anomalies detected by NAIA

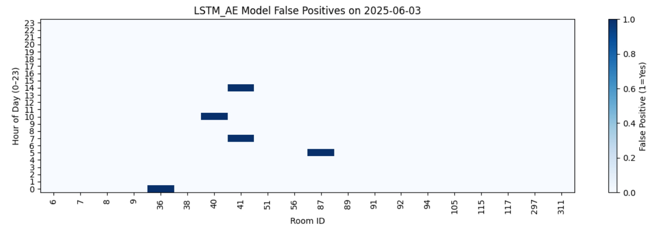


Figure 55: False positive anomalies detected by LSTM-AE

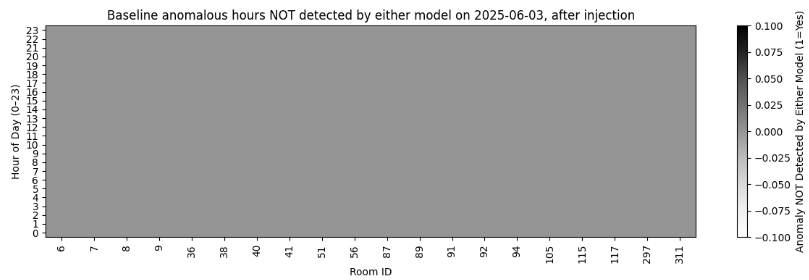


Figure 56: Baseline anomalies not detected after injection

# Appendix E AMOUNT OF HISTORICAL DATA PER ROOM

| Room | Rows | Room | Rows | Room | Rows | Room | Rows | Room | Rows |
|---|---|---|---|---|---|---|---|---|---|
| Room 1 | 1 382 273 | Room 2 | 1 382 336 | Room 3 | 1 367 152 | Room 4 | 1 367 167 | Room 6 | 1 367 137 |
| Room 7 | 1 225 516 | Room 8 | 1 225 287 | Room 9 | 1 225 519 | Room 20 | 1 355 914 | Room 21 | 1 423 660 |
| Room 27 | 1 659 520 | Room 28 | 197 094 | Room 29 | 775 281 | Room 30 | 775 253 | Room 32 | 775 282 |
| Room 33 | 774 312 | Room 34 | 774 312 | Room 35 | 774 303 | Room 36 | 774 428 | Room 37 | 772 380 |
| Room 38 | 772 380 | Room 39 | 773 734 | Room 40 | 197 074 | Room 41 | 197 074 | Room 51 | 774 473 |
| Room 56 | 772 574 | Room 57 | 774 427 | Room 59 | 777 849 | Room 60 | 777 809 | Room 87 | 775 020 |
| Room 88 | 775 351 | Room 89 | 775 353 | Room 90 | 774 608 | Room 91 | 774 609 | Room 92 | 774 608 |
| Room 93 | 776 497 | Room 94 | 776 496 | Room 95 | 776 495 | Room 104 | 777 373 | Room 105 | 777 133 |
| Room 115 | 196 962 | Room 116 | 196 962 | Room 117 | 196 962 | Room 261 | 196 988 | Room 297 | 773 964 |
| Room 311 | 1 710 590 | | | | | | | | |

**Total rows:** 38 997 491 **Average rows per room:** 847 771.54 **Median rows per room:** 775 267 **Max rows (Room 311):** 1 710 590 **Min rows (Room 115):** 196 962

Table 6: The amount of historical data per room with sufficient (four sensor) data in number of rows in the database