

Submitted by
Samuel Tim
Student Number 307636

samuel.tim200@yahoo.de

B

C

Bachelor Thesis
Energy Anomaly Detection with
Machine Learning

S

Konstanz, 31st December 2025

Bachelor Thesis

Energy Anomaly Detection with Machine Learning

by

Samuel Tim

in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science

in Applied Computer Science

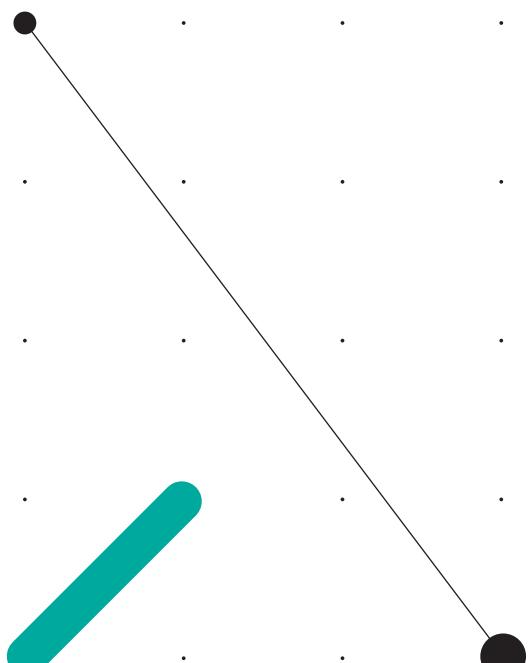
at the Hochschule Konstanz University of Applied Sciences,

Student Number: 307636

Date of Submission: 31st December 2025

Supervisor: **Prof. Dr. Marko Boger**

Second Examiner: **Dipl.-Inf. Björn Erb**



An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Abstract...

Contents

1

Introduction

1.1. Motivation and Problem Statement

Energy efficiency and resource conservation are paramount challenges in the context of global climate change and rising operational costs. In both industrial and residential settings, energy consumption data is being collected at an unprecedented scale, largely due to the widespread adoption of smart meters and Internet of Things (IoT) devices [SomeSmartMeterPaper2023].

While this data holds immense value, it also presents a significant challenge: identifying consumption patterns that deviate from the norm. These *anomalies* can represent critical information, such as:

- Equipment malfunction or failure,
- Inefficient operational processes,
- Data integrity issues from faulty sensors, or
- Potential for energy savings and process optimization.

The manual analysis of these vast, high-velocity time series datasets is impractical. Automated methods are required to detect these anomalies reliably and in near real-time. However, defining a "normal" energy profile is complex due to factors like seasonality (daily, weekly, yearly cycles), weather dependencies, and stochastic user behavior. This complexity makes simple threshold-based detection methods ineffective, leading to a high rate of false positives or missed detections.

1.2. Research Goal and Objectives

The primary goal of this thesis is to design, implement, and evaluate a robust system for anomaly detection in energy consumption time series data. This work aims to compare different algorithmic approaches to identify the most suitable method for a given energy dataset.

To achieve this goal, the following key objectives are defined:

1. **Literature Review:** To investigate the fundamentals of time series anomaly detection and review the current state-of-the-art (Related Work) specific to energy data.
2. **Data Preprocessing:** To select a suitable energy dataset and develop a preprocessing pipeline to handle missing values, normalize data, and engineer relevant features.
3. **Model Implementation:** To implement and train several detection models, likely including a statistical baseline (e.g., ARIMA) and one or more machine learning models (e.g., Isolation Forest, Autoencoder).
4. **Evaluation:** To define and apply appropriate evaluation metrics (e.g., Precision, Recall, F1-Score) to systematically compare the performance of the implemented models.

1.3. Scope and Delimitations

This thesis focuses on **unsupervised anomaly detection**, as labeled anomaly data is rare and expensive to obtain in real-world energy scenarios. The primary input will be univariate energy consumption time series. While external factors like weather or building occupancy are acknowledged as influential, their integration as exogenous variables is considered outside the primary scope of this work but will be discussed as potential future work.

Furthermore, this work is concerned with the **detection** of anomalies, not their **diagnosis**. The system will flag a data point or sequence as anomalous, but it will not perform a root-cause analysis of the anomaly's origin.

1.4. Structure of the Thesis

This document is organized into six chapters:

Chapter 1 (Introduction) motivates the research topic, defines the core problem, and outlines the goals and structure of this thesis.

Chapter 2 (Foundations and Related Work) discusses the theoretical background of energy time series data and anomaly detection. It reviews existing academic literature (Related Work) to identify current approaches and an existing research gap.

Chapter 3 (Conception and Methodology) presents the conceptual design of the proposed system. It details the selected dataset, the data preprocessing pipeline, and the theoretical foundation of the chosen detection algorithms.

Chapter 4 (Implementation) describes the technical realization of the concepts from Chapter 3, including the software stack (e.g., Python, scikit-learn, TensorFlow) and key implementation details.

Chapter 5 (Evaluation and Results) details the experiments conducted to assess the performance of the implemented models. The results are presented, visualized, and critically discussed.

Chapter 6 (Conclusion and Future Work) summarizes the key findings of the thesis, answers the initial research questions, and provides an outlook on potential future research directions.

2

Fundamentals (Background)

2.1. Application Domain: Smart Buildings and Energy Management

The building sector is currently facing a significant transformation, driven by two primary forces: rising public and corporate demand for sustainability (Environmental, Social, and Governance - ESG) and increasingly stringent government legislation aimed at energy monitoring and consumption reduction.

The urgency for this transformation is clear: buildings are responsible for approximately **40% of global energy consumption** and a similar fraction of energy-related CO₂ emissions [**TODO Cite IEA or EU Commission**]. This massive energy footprint represents both a critical challenge and a significant opportunity. Reducing this consumption is not only a legislative necessity but also a powerful lever for companies to meet ESG goals and reduce operational costs.

This trend is enabled by rapid advancements in the Internet of Things (IoT). IoT sensors and edge computing devices are becoming simultaneously more powerful and more cost-effective. This allows modern "Smart Buildings" to move beyond simple automation (like lighting or heating) towards comprehensive, high-frequency data collection. This creates a critical need to monitor all building subsystems, especially energy-related assets like HVAC (Heating, Ventilation, and Air Conditioning), lighting, and plug loads.

The data generated by these systems is finally received and stored in a centralized IoT platform or an **Energy Management Information System (EMIS)**. In systems such as these, the raw time series data is processed, stored in time-series databases,

and made available for visualization, alarming, and advanced analytics—which is the starting point for the work in this thesis.

2.1.1. Data Sources and Acquisition in Smart Buildings

The data in a smart building originates from a diverse set of heterogeneous sources. These systems must bridge the gap between traditional Operational Technology (OT) and modern Information Technology (IT).

- **Building Management Systems (BMS):** This is the central "brain" of the building, often based on established frameworks like **Tridium Niagara** or **openBOS**. It controls and monitors core assets like HVAC (chillers, pumps, air handling units) and lighting. These systems traditionally communicate over building-specific protocols like **BACnet** or **Modbus**.
- **Smart Meters (AMI):** Historically, utility consumption (electricity, gas, water) was tracked by **legacy meters** that required manual, infrequent readings (e.g., annually). A key enabler for smart buildings is the retrofitting or replacement of these with smart meters, which are part of an Advanced Metering Infrastructure (AMI). While these meters *can* provide data at high frequencies (e.g., every second), a **15-minute interval** has become a de-facto standard for energy balancing and building analytics.
- **IoT Sensors:** A third, rapidly growing source is a wide array of granular IoT sensors. These are often wireless and provide high-resolution data on factors that influence energy consumption, such as **CO2** levels, room **occupancy** (PIR sensors), **temperature**, and **humidity**.

Data from these different sources is collected and unified by on-site **edge gateways** (which could be the BMS itself, like a Niagara JACE, or a separate device). These gateways poll the Modbus/BACnet devices, read the smart meters, and subscribe to IoT sensor data. The edge device then aggregates, standardizes, and transmits this data to the central EMIS, typically using lightweight, IT-friendly protocols like **MQTT**.

2.2. Data Foundation: Time Series

Formally, a **time series** is a sequence of data points indexed in time order. In the context of this thesis, every time a sensor or meter in the building is read (e.g., a 15-minute

energy value), a data point is generated. This data point consists of two components: the **timestamp** (the exact time the measurement was taken) and the **value** (the measurement itself).

This sequential, time-ordered nature is the defining characteristic of time series data. It implies that data points are not independent; rather, each point has a temporal relationship with its predecessors and successors. This structure is fundamental to the analysis, as patterns (like trends and seasonality) and anomalies are only revealed by observing the data in its correct sequence.

2.2.1. Characteristics of Energy Time Series

Energy time series data exhibits several key characteristics:

- **Seasonality:** Daily (e.g., building operation hours), weekly (weekends vs. weekdays), and annual (summer vs. winter) cycles.
- **Trend:** (e.g., increasing consumption due to expansion, or decreasing due to efficiency measures).
- **Autocorrelation:** The value at time t is highly dependent on the value at time $t - 1$.
- **Noise:** Inherent randomness and measurement errors.

2.2.2. Common Data Faults and Challenges

While time series analysis is powerful, data from real-world building systems is notoriously "dirty" and presents significant challenges. The detection of these data faults is a prerequisite for any meaningful anomaly detection, as one must first distinguish between a *data error* and a true *process anomaly*.

A primary source of data faults comes from how consumption is calculated. Many utility meters (e.g., for electricity or gas) provide a **cumulative meter reading**, which is a monotonically increasing value. The interval consumption is then calculated by the edge gateway or platform as the difference between the current reading and the previous one.

This calculation is highly vulnerable to **connection errors**. If the edge gateway loses connection to the meter for a period (e.g., one hour), it cannot record the intermediate values. When the connection is re-established, it reads the new, higher meter value. The system then calculates the *entire* consumption that occurred during the

downtime and logs it as a single data point. This creates a large, artificial **data spike** in the consumption time series, which does not represent a real surge in energy use but rather a data-logging artifact.

Other common data problems in this domain include:

- **Missing Values:** Gaps in the data (often shown as ‘null’ or ‘NaN’) due to connection loss, device power failure, or database errors.
- **Stuck Values:** The sensor or meter “freezes” and reports the exact same value repeatedly over a long period, which is physically improbable for dynamic values like power consumption or temperature.
- **Sensor Drift:** A gradual, systematic error where a sensor’s accuracy degrades over time (e.g., a CO₂ sensor requires recalibration).
- **Out-of-Bounds Values:** Data points that are physically impossible, such as negative energy consumption (for a non-producing asset) or an indoor temperature of 1000 °C, which are clearly sensor failures.

2.2.3. Common Data Preprocessing Techniques

Before a reliable anomaly detection model can be trained, the data faults described in Section ?? must be addressed. Raw building data is almost never suitable for direct analysis. The goal of preprocessing is to clean, standardize, and transform the data into a high-quality format suitable for machine learning models.

Key preprocessing steps include:

- **Handling Missing Values (Imputation):** Gaps in data (e.g., ‘NaN’ values from connection loss) must be filled.
 - **Simple Methods:** Techniques like **Linear Interpolation** (drawing a straight line between two known points) or **Forward Fill** (carrying the last known value forward) are fast but often inaccurate.
 - **Statistical Methods:** More advanced models, like ARIMA, can be used to forecast and fill in small gaps based on statistical properties.
 - **Deep Learning & Foundation Models:** For complex, seasonal data, modern approaches use deep learning models (like LSTMs or Transformers) trained on similar, complete data to predict the missing values. Recent **time series foundation models** (FMs) are particularly powerful, as they can be used for “zero-shot” imputation—filling gaps by leveraging patterns learned from vast, unrelated datasets [TODO Cite FM Imputation Paper].

- **Correcting Faulty Data:**
 - **Cumulative Meter Spikes:** The most common fault—a large spike from a re-connection—is typically handled by a specific algorithm. This involves detecting the spike, replacing it and the preceding gap with ‘NaN’, and then using an imputation method (like linear interpolation or back-distribution) to evenly spread the total consumption across the gap period.
 - **Stuck Values:** These are often detected by finding runs of identical values (e.g., ‘value(t) == value(t-1)’ for N consecutive steps) and replacing them with ‘NaN’ to be imputed.
 - **Out-of-Bounds Values:** Simple clipping (Winsorizing) or removal of physically impossible data (e.g., negative consumption, 1000°C) is a necessary first step.
- **Resampling:** Since data can arrive at inconsistent intervals (e.g., 5 min, 15 min, 1h), all time series must be resampled to a uniform frequency, such as the **15-minute** interval common in energy management. This is typically done using an aggregation (e.g., ‘mean()’, ‘sum()’) or interpolation.

2.2.4. Univariate vs. Multivariate Time Series

Time series data can be classified based on the number of variables recorded at each timestamp.

A **univariate** time series measures only a single variable over time. An example would be a dataset tracking **only** the total electricity consumption of a building. While simple, this approach ignores all external factors.

In contrast, a **multivariate** time series measures multiple variables simultaneously. In this case, the dataset would contain parallel, time-aligned sequences, such as electricity consumption, outside air temperature, and building occupancy.

Energy data is, by its very nature, a prime example of a multivariate problem. The energy consumption of a building is not an independent variable; it is heavily influenced by a multitude of other factors (**covariates**), such as weather, occupancy, and operational schedules. By capturing these rich dependencies, a multivariate model can learn a far more accurate and robust definition of “normal” behavior.

This thesis focuses on multivariate data specifically because this relationship between variables is the key to finding complex **contextual anomalies**—for exam-

ple, identifying high energy use that would be normal on a cold day but is highly anomalous on a warm day.

2.3. Problem Definition: Anomaly Detection

This section defines what an anomaly is and applies this definition to the specific context of building energy data.

Globally, an anomaly is a data point, or a sequence of points, that deviates significantly from the majority of the data or a predefined notion of "normal" behavior [TODO Chandola2009]. In the context of this thesis, this definition requires further refinement. But let's first explore the different types of anomalies.

2.3.1. Types of Anomalies

An anomaly is a data point, or a sequence of points, that deviates significantly from the majority of the data or a predefined notion of "normal" behavior [TODO Chandola2009]. Following the taxonomy from [TODO Paparrizos2025], we distinguish:

Point Anomalies: A single data instance that is anomalous with respect to the rest of the data. In energy data, this could be a true consumption spike (e.g., a large, unscheduled device activation) or a data fault (e.g., a sensor failure).

Contextual Anomalies: An instance that is anomalous only in a specific context. This is the most critical anomaly type for energy management (e.g., high heating energy consumption in summer is anomalous, but normal in winter). This highlights why multivariate data (providing context like weather) is essential.

Collective (Sequence) Anomalies: A collection of related data instances that are anomalous as a group, even if individual points are not (e.g., a subtle but persistent unusual vibration pattern in an HVAC pump).

A critical distinction must be made between anomalies in the *data logging process* (data faults, as discussed in Section ??) and anomalies in the *physical energy consumption*. For example, a spike caused by a temporary connection issue to a cumulative meter is a data anomaly, but not a real process anomaly. This thesis

seeks to identify anomalies in the *real energy consumption*, not data pipeline errors.

Furthermore, not all statistical deviations are true anomalies. A high consumption spike every Monday morning caused by a scheduled AI-workstation training session is not an anomaly; it is a recurring, explainable pattern. A robust detection model must learn this context and correctly identify only *unexpected* deviations.

Therefore, this thesis will primarily focus on detecting **Point** and **Contextual Anomalies** that signify real, unexplained deviations in energy usage. **Collective (Sequence) Anomalies** are not a primary focus. While they might occur (e.g., an HVAC unit cycling more rapidly than usual), the main consequence of such inefficiency is typically a higher-than-expected energy consumption, which will be detected as a point or contextual anomaly anyway. This marks a key difference from other domains like medical ECG analysis, where the precise *shape* of a sequence is paramount.

2.4. Anomaly Scoring

Most anomaly detection algorithms do not output a binary "normal" or "anomalous" label. Instead, they assign a continuous numerical value to each data point, known as an **anomaly score**. This score quantifies how "unusual" a data point is, given the learned model of normality.

By convention, this score is often normalized to a range between **0 and 1**.

- A score of **0** (or close to 0) indicates that the data point is perfectly normal.
- A score of **1** (or close to 1) indicates that the data point is highly anomalous.

This approach provides more flexibility than a simple binary classification. To generate a concrete alarm, a human operator or a downstream process must set a **threshold** (e.g., 0.95). Any data point with a score exceeding this threshold is then flagged as an anomaly. The following methodologies describe different ways to generate this anomaly score.

2.5. Taxonomy of Detection Methodologies

Detection methods can be broadly grouped into three main families, primarily differing in how they model "normal" data and, consequently, how they calculate the anomaly score [TODO Paparrizos2025]:

Distance-based: Assumes that normal data points are "close" to their neighbors.

The anomaly score is a function of the distance of a point to its nearest neighbors (e.g., k-NN).

Density-based: Assumes that normal data points are located in dense regions.

The anomaly score is derived from the local data density (e.g., Isolation Forest calculates a score based on how easily a point is "isolated").

Prediction-based: Assumes that normal data is predictable. A model learns the

temporal patterns of the data. The anomaly score is then based on the deviation between the model's expectation and the actual observed data.

2.5.1. Focus: Prediction-Based Methodologies

Given that energy data is a multivariate time series where context is critical, prediction-based methods are a natural fit. This family itself can be split into two main approaches based on how the model "predicts" the data:

- **Reconstruction-based:** The model (e.g., an Autoencoder) is trained to compress and then reconstruct its own input. It learns a compact representation of "normal" data. The anomaly score is the **reconstruction error**—the difference between the original data point and its reconstructed version. The assumption is that the model will be very good at reconstructing normal patterns but will fail to reconstruct anomalous ones, resulting in a high error.
- **Forecasting-based:** The model is trained to predict the next time step(s) based on past data. The anomaly score is the **prediction error** (or **residual**)—the difference between the model's forecast and the actual value that occurs.

A more advanced version of the forecasting-based approach, and the focus of this thesis, is **Probabilistic Forecasting**. This method provides a key advantage over simple "point" predictions:

- **Point Forecasting:** The model predicts a single value (e.g., 21.5°C). The anomaly score is the residual (e.g., $|actual - predicted|$). This can be noisy if the model is uncertain but predicts a single "wrong" value.
- **Probabilistic Forecasting:** The model predicts a full probability distribution, often represented by a prediction interval (e.g., "The value will be between 21.0°C and 22.0°C with 95% confidence").

This thesis will leverage probabilistic forecasting, defining an anomaly as a real observation that falls **outside** of the predicted confidence interval. This is a more robust approach as it accounts for the model's own uncertainty.

2.6. Fundamentals of Evaluation

To evaluate the performance of an anomaly detection model, we require labeled data (ground truth) and robust evaluation metrics. While large, public benchmark datasets for anomaly detection exist, such as TSB-UAD [[paparizos2022tsb](#)] and the more recent TSB-AD [[liu2024elephant](#)], only a small fraction of these datasets consist of multivariate energy data that exhibits complex contextual anomalies and the inherent chaotic nature of building operations. Therefore, to validate the methods in this thesis, a domain-specific, labeled dataset of multivariate energy time series is essential.

Given a labeled dataset and an anomaly score (as defined in Section ??), a binary decision is made by applying a **threshold**. Any score above this threshold is classified as "anomalous," and any score below it as "normal." This classification allows for the creation of a **Confusion Matrix**, which forms the basis of most standard metrics.

2.6.1. Point-based Metrics

These metrics evaluate each time stamp independently, treating the entire time series as a collection of independent classification problems. The Confusion Matrix consists of:

- **True Positives (TP):** An anomaly occurred, and the model correctly flagged it.

- **True Negatives (TN):** The data was normal, and the model correctly ignored it.
- **False Positives (FP):** The data was normal, but the model incorrectly flagged an anomaly (Type I Error).
- **False Negatives (FN):** An anomaly occurred, but the model failed to detect it (Type II Error).

From this matrix, metrics like **Precision** ($\frac{TP}{TP+FP}$) and **Recall** ($\frac{TP}{TP+FN}$) are derived. [cite_{start}]The **F1-Score** ($2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$) is the harmonic mean of Precision and Recall, providing a single score that balances the trade-off between FPs and FNs [Tutorial`Advances in Time-Series Anomaly Detection.pdf].

Since choosing a single threshold is problematic, threshold-independent metrics are often preferred. These metrics evaluate the model's performance across *all* possible thresholds:

[cite_{start}]

- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** Plots the True Positive Rate (Recall) against the False Positive Rate (FPR) [Tutorial`Advances in Time-Series Anomaly Detection.pdf].
- **AUC-PR (Area Under the Precision-Recall Curve):** Plots Precision against Recall. [cite_{start}]This is often considered more informative than AUC-ROC for highly imbalanced datasets.

2.6.2. Sequence-based Metrics (VUS)

[cite_{start}]Standard point-based metrics like F1 and AUC-ROC have a critical flaw when applied to time series: they are highly sensitive to misalignment or lag [Tutorial`Advances in Time-Series Anomaly Detection.pdf].

An anomaly detection model might correctly identify an anomalous event but flag it a few time steps early or late. [cite_{start}]A point-based metric would severely penalize this, double-counting the error: it would register a **False Positive** (for the misaligned detection) and a **False Negative** (truth point) [Tutorial`Advances in Time-Series Anomaly Detection.pdf].

To address this, **Range-based** or **Event-based** metrics were developed. [cite_{start}]These metrics evaluate events and are more tolerant of minor time shifts, for example by accepting any detection within a "buffer".

A recent and highly robust metric in this category is the **VUS (Volume Under the Surface)** [paparrizos2022volume; boniol2025vus]. The core idea of VUS is to extend the

concept of range-based evaluation. Instead of choosing one fixed buffer size, VUS computes the AUC (e.g., AUC-PR) for *all possible buffer lengths*, from zero up to a predefined maximum. [cite_{start}] This calculation creates a 3D surface (e.g., Precision, Recall, and Buffer Length), an [cite_{start}] VUS has been demonstrated to be significantly more robust to lag, noise, and other real-world data issues compared to traditional metrics [Tutorial: Advances in Time-Series Anomaly Detection.pdf] based detection and its status as the most reliable evaluation measure [TSB-AD-NeurIPS24 (1).pdf], the VUS-PR (VUS-Recall Surface) is selected as the primary evaluation metric for this thesis.

2.7. Learning Paradigms for Anomaly Detection

The methodologies introduced in Section ?? can be further categorized based on the type of data required for training. This "learning paradigm" dictates how a model learns "normal" behavior and is a critical choice in any real-world application.

Supervised: This approach requires a fully labeled dataset containing examples of *both* normal operation and various types of anomalies. The model is trained as a standard classifier (e.g., "normal" vs. "fault type A"). This is common in domains like medical diagnostics, where known anomalous patterns (e.g., specific arrhythmia in an ECG) are well-documented and labeled by experts. However, this is impractical for building energy data, as it is impossible to pre-collect and label all potential future faults.

Unsupervised: The model receives a single dataset with no labels, operating under the assumption that anomalies are "rare and different" from the majority of the data. The model must find these deviations without any prior reference for what is normal or anomalous. Most distance-based and density-based methods are inherently unsupervised.

Semi-supervised: This is the most practical and common approach for time-series anomaly detection. The model is trained *only* on a "healthy" dataset that is assumed to represent a normal, fault-free baseline. The model's sole objective is to learn a comprehensive representation of this normality. When presented with new, unseen data, any point that deviates significantly from this learned baseline is flagged as an anomaly.

In the domain of building energy management, we have a distinct advantage that aligns perfectly with the **semi-supervised paradigm**. Buildings often have months or even

years of historical consumption data. While this data is unlabeled, it can be "curated"—for instance, by selecting a recent, representative period of operation—and be *assumed* to be a "mostly normal" baseline.

Therefore, this thesis will focus on **semi-supervised anomaly detection**. This approach allows us to leverage vast historical data to train robust models. However, it also introduces two specific challenges that this work will address:

- **Baseline Contamination:** What if the historical "baseline" data already contains anomalies? The model might incorrectly learn these faults as "normal" behavior.
- **Concept Drift:** How do we detect anomalies when the baseline itself changes (e.g., due to equipment retrofits, new tenants, or seasonal changes) and the original "healthy" data is no longer representative?

The following model architectures are particularly well-suited for this semi-supervised, prediction-based approach:

- **Isolation Forest (IForest):** While traditionally unsupervised, IForest can be effectively used in a semi-supervised manner by training its trees *only* on the "healthy" baseline data. Anomalous points, which were not seen during training, will require fewer random "cuts" to be isolated, resulting in a high anomaly score.
- **Autoencoders (AE):** As discussed in Section ??, this is a quintessential semi-supervised model. The AE is trained to reconstruct *only* the normal baseline data. When it is fed an anomalous data point, it will fail to reconstruct it accurately, leading to a high reconstruction error (the anomaly score).
- **Recurrent Neural Networks (RNN/LSTM):** When used as a forecaster, an RNN/LSTM model is trained on sequences of normal data to predict the next time step. Its anomaly score is the prediction error. If an anomaly occurs, the model's prediction (based on normal patterns) will differ significantly from the anomalous reality.
- **Mixture Density Models (MDM):** These models, often combined with RNNs, do not just predict a single value. Instead, they learn a full *probability distribution* of "normal" data (a "mixture" of multiple distributions). The anomaly score is the negative log-likelihood; if an observed data point is highly improbable under the learned normal distribution, it receives a high anomaly score.

- **Foundation Models (FM):** Large-scale models (e.g., [TODO`Moirai2024; TODO`Chronos2024]) pre-trained on diverse public datasets can be fine-tuned on our *specific* baseline energy data. This "few-shot learning" adapts the model's general understanding of time series patterns to our building's specific "normal" behavior. Anomalies are then detected as deviations from this fine-tuned baseline.

3

State of the Art (Related Work)

3.1. Implementations of Detection Algorithms

3.1.1. Statistical and Classical Machine Learning Methods

These methods are often used as efficient baselines.

- **ARIMA/SARIMA:** (Autoregressive Integrated Moving Average). Models the time series and flags points with high residuals as anomalous [[ExampleARIMAPaper](#)].
- **Isolation Forest (IF):** An ensemble method that "isolates" outliers. It is very efficient and scales well [[Liu2008`IF](#)].
- **One-Class SVM (OC-SVM):** A kernel-based method that learns a boundary around the "normal" data [[Scholkopf2001`OCSVM](#)].
- **DBSCAN:** A density-based clustering algorithm that identifies points in low-density regions as outliers [[Ester1996`DBSCAN](#)].

3.1.2. Deep Learning Methods

Deep learning is powerful for capturing complex, non-linear patterns.

- **LSTM-based Autoencoders (LSTM-AE):** This architecture combines the sequential power of LSTMs with the reconstruction-based anomaly detection of an Autoencoder. It is a very common approach for time series [[Malhotra2016`LSTM`AD](#)].

- **LSTM-based Prediction:** Simple LSTM networks are trained to predict the next time step; a large prediction error indicates an anomaly.

3.1.3. Foundation Models for Time Series

A recent paradigm shift involves using large, pre-trained Foundation Models.

- **Probabilistic Forecasting Models:** Models like **Moirai** [Moirai2024] and **Chronos** [Chronos2024] are pre-trained on vast public datasets (including energy data). They can provide probabilistic (quantile) forecasts in a zero-shot or few-shot (finetuned) manner, making them ideal for prediction-based anomaly detection.
- **General-Purpose Models:** **MOMENT** [Goswami2024] **MOMENT** is a foundation model pre-trained for multiple tasks, including explicit anomaly detection.

3.2. Applications in Energy Anomaly Detection

3.2.1. Applications in Commercial and Residential Buildings

For example: [ExamplePaperBuilding2021] used an LSTM-based Autoencoder to detect anomalies in HVAC systems of a university building. They found... Another study by [ExamplePaperSmartHome2022] compared Isolation Forest and OC-SVM for smart home data and found that... Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

3.2.2. Applications in Industrial and Grid-Level Data

For example: [ExamplePaperGrid2020] focused on anomaly detection in smart grid phase measurements (PMU data) using... Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

3.2.3. Comparative Studies and Benchmarks

A recent benchmark by [ExampleBenchmarkPaper2023] evaluated five different algorithms on a public energy dataset. Their findings indicate that... The TSB-UAD benchmark [Paparrizos2022' TSB] also provides extensive comparison, although...

3.3. Research Gap and Contribution

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

This thesis aims to fill this gap by... (e.g., "...by designing, implementing, and evaluating an anomaly detection system integrated into the [Firma X] IoT platform. It will systematically compare a state-of-the-art Foundation Model (Moirai) using probabilistic forecasting against a highly efficient classical model (Isolation Forest)...") This leads directly to the methodological choices in the next chapter.