Chapter 4

A Scalable Aberration Detection Technique for Smart Energy Meters

In the previous two chapters, we explained software system based solutions that enable the development and deployment of extensible energy management applications. In this chapter, we shift the focus to proposing novel data analytics method for identifying abnormal energy usage for a network of buildings within a neighborhood. The proposed method leverages the opportunistically available aggregate level smart meter readings and metadata attached to the meter identity, as collected by utilities. We proposed an unsupervised aberration detection method using partial context information which reduces false positives. Further, it gives a ranked list of potential abnormal energy usage instances for further inspection by the building owners [1].

4.1 Background and Motivation

There has been rapid growth in energy monitoring infrastructure in several countries. For example, by 2016, 70.8 million Advanced (smart) Metering Infrastructure (AMI) installations are done done by the US electric utilities [110]. Aggregated level electricity usage of both residential and

[1] Please note that, in this chapter, we use the terms aberration, abnormal and anomaly interchangeably, consistent with the literature[67, 88, 114], although their exact definition may differ under different context.

commercial buildings are measured and recorded by the smart meters at a minimum of hourly intervals. Analyzing this massive volume of smart meter data for identifying energy wastage events is an important enabler for reducing the energy usage for a large number of buildings.

Several automated methods have been proposed in the recent literature for identifying abnormal energy usage events from the aggregated level smart energy meter data [67, 88, 114]. One of the important drawbacks of existing anomaly detection algorithms is that various unknown context variables, such as seasonal variations, can affect the energy consumption of buildings in ways that appear anomalous to existing time series based anomaly detection algorithms. Moreover, the primary intuition behind these approaches is that deviation from baseline energy usage bears a one to one relationship with anomalous events. However, this assumption has two major flaws. First, not all deviations from baseline patterns are anomalous; energy consumption can vary significantly based on the day of the week, the time of the year, and many other seasonal and contextual variables. Second, not all anomalous events display markedly different energy consumption; small deviations may still be significant when contrasted with data from other consumers within the neighborhood.

In this chapter, we present a novel method for identifying abnormal energy usage events using partially known context information. Our approach recognizes that sensing every possible context variable (such as occupancy or zone temperature) is technically and economically infeasible. Instead, we use context information which is directly available from meter readings: timestamps and metadata attached to the meter identity. Temporal context information, extracted from timestamps, is used for improving the anomaly detection accuracy by picking only the relevant historic data for baseline estimation. This captures the effect of factors such as operating hours (for commercial buildings) and seasonal changes (such as heating/cooling loads) on the consumption characteristics of the buildings.

Neighborhood information (as derived from available metadata) is used for adjusting the anomaly score to account for unknown context variables that influence historically correlated consumption patterns in the same way. For each building or user, the neighborhood is defined as the set of all other users that have similar characteristics (function, location or demography), and are therefore likely to react and consume energy in a similar way in response to the external conditions. The neighborhood can be predefined based on prior customer information or can be identified through an analysis of historical energy consumption.

The proposed approach consists of three steps:

1. Split the given meter readings into disjoint sets based on available temporal context information.
2. Within each temporal context, run an anomaly detection algorithm separately on each meter's readings.
3. Adjust the anomaly score for individual meter readings based on available neighborhood information.

We validate the effectiveness of the proposed algorithm for both residential and commercial buildings using energy consumption data for a year, thus capturing seasonal trends as well as shorter-term ones. Through experiments, we show the importance of using temporal context and neighborhood information which significantly improves the anomaly detection accuracy in comparison with an existing anomaly detection method proposed in [67].

4.2 Definitions and Assumptions

Electric utilities install smart meters in their consumer's buildings and record the aggregate level electricity usage, typically sampled at a one-hour interval, on a massive scale. Our objective is leveraging such opportunistically available large dataset for detecting abnormal energy usage behavior and provide feedback to the consumers for further action. Note that abnormal energy usage may happen due to several reasons such as 1) a faulty appliance consuming more electricity than usual; 2) an electric component failure leading to high or low power consumption; 3) misusage of devices by humans when not required; or 4) genuine usage (e.g. festival). In this work, we are primarily targeting usage anomalies, any irregular (high or low) energy usage patterns due to human mistake. e.g., lights are left on during non-working hours. Identification of other anomalies may require retrofitting for collecting fine-grained sensor data and operational context, which is not the focus of this work.

Anomaly Types

Given that anomaly events can occur at any time, we broadly characterize the common anomalies in real-world buildings into two categories:

- **Single instance anomaly:** Occurrence of an anomaly event on a single instance in the given time series of measurements. An example of single instance anomaly is a sudden increase in the building's energy consumption on a given day. Figure 4.1 shows the examples of single point anomalies (a few days within the black boxes) in Meter 2 and 4 as the lighting systems were not turned off by mistake during non-operational hours, e.g. night.
- **Sequence anomaly:** A set of consecutive anomalous events over a short period of time. One example of such an anomaly would be increased consumption for a given building for an entire week. Figure 4.1 shows examples of sequence anomalies (red boxes) in Meter 1 and 4 as some devices were left on during night hours for few days.

Temporal context sets

As explained in the previous section, several types of context variables can influence energy consumption for a user or a set of users. In order to avoid using context variables, such as fine-grained occupancy and appliance usage patterns, that can only be measured with expensive sensing infrastructure, we only use temporal context variables in this work. Since energy usage events follow regular temporal patterns, we define *temporal context sets* which split the given meter readings into to N disjoint temporal subsets. Figure 4.1 shows the hourly power consumption of a commercial building complex for a year. It shows several temporal context changeover

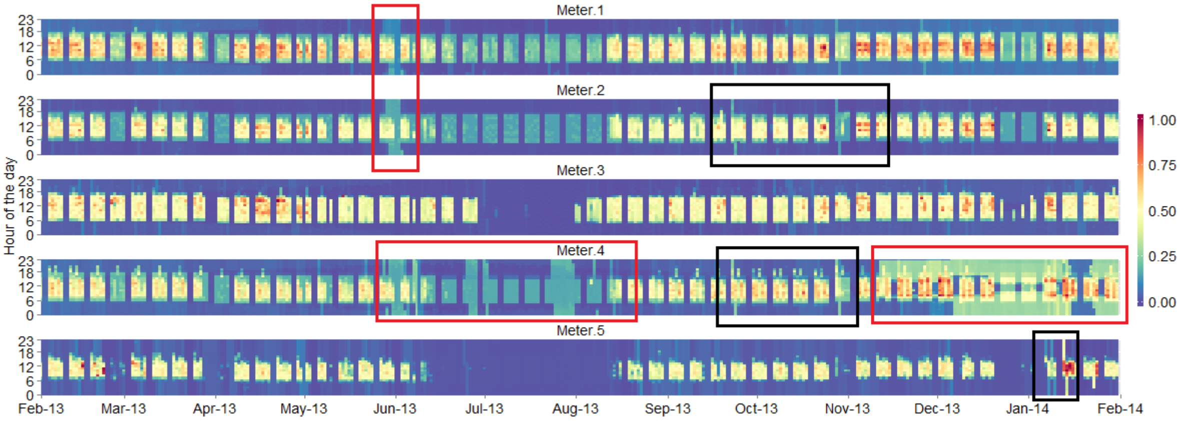


Figure 4.1: Hourly power usage (normalized) of different buildings with in a large commercial building complex (neighborhood) in Sweden for a year from 1st Feb 2013 to 31st Jan 2014. It shows, (a) daily and weekly power usage cycles with seasonal variations during summer, winter and holidays, (b) examples of single point anomaly (marked in black), and (c) examples of sequence anomaly (marked in red). X-axis denotes the day of the year while Y-axis is hour of the day.

events including daily cycle (working and non-working hours), weekly cycle (week days and weekends) and seasonal variations due to summer holidays, weather change and different operational modes of a building. As an example, each meter reading can be classified into one of three temporal context sets, based on the development presented in Section 4.3.1:

- **WorkingDay-BusinessHours:** This set contains meter readings taken during business hours (8 am to 5 pm) on working days (Monday to Friday).
- **WorkingDay-NonBusinessHours:** This set contains meter readings taken outside business hours (5 pm to 8 am) on working days.
- **Weekend:** This set contains meter readings taken during weekends, when commercial premises are unoccupied throughout the day and night, while residential buildings are likely to have increased occupancy.

Multiple temporal context sets can also be defined for each meter reading based on the operating characteristics of a building e.g. residential vs commercial.

Definition of neighborhood

The *neighborhood* of a given building (or residence) is defined as the set of buildings (or residences) that are expected to be influenced similarly by the same context variables (known or unknown). This definition is based on available metadata. For example, the neighborhood of a commercial building may be defined to be the set of all other buildings within the same commercial complex (*administrative* neighbors). Alternatively, the neighborhood of a school building may be defined to be the set of all other school buildings in the same geographical area (*functional* neighbors).

Neighborhood information can be predefined by a domain expert based on prior customer information or can be identified through an analysis of historical energy consumption and available metadata. As an example, authors in [120] proposed a framework for grouping the consumers based on several contextual dimensions such as locations, communities, seasons, weather patterns, and holidays. In this work, we assume that identification of neighborhood information is a prior step of applying the proposed anomaly detection method.

Our objective is to identify potential abnormal energy usage events using aggregate (building level) smart meter data as the input. The *abnormality* is quantified by computing anomaly scores for each time period. The anomaly score computation combines metadata based neighborhood definitions with the historical correlation between each pair of time series, in order to compute the pairwise influence of neighbors on each other's individual anomaly scores. In the absence of any metadata, the default neighborhood is the set of all users, and pairwise influences are computed purely using historical correlation.

4.3 Anomaly Detection Algorithm

In this section, we describe the algorithm developed for computing anomaly scores for energy consumption data, and also for flagging potential events of interest. The anomaly score, as it

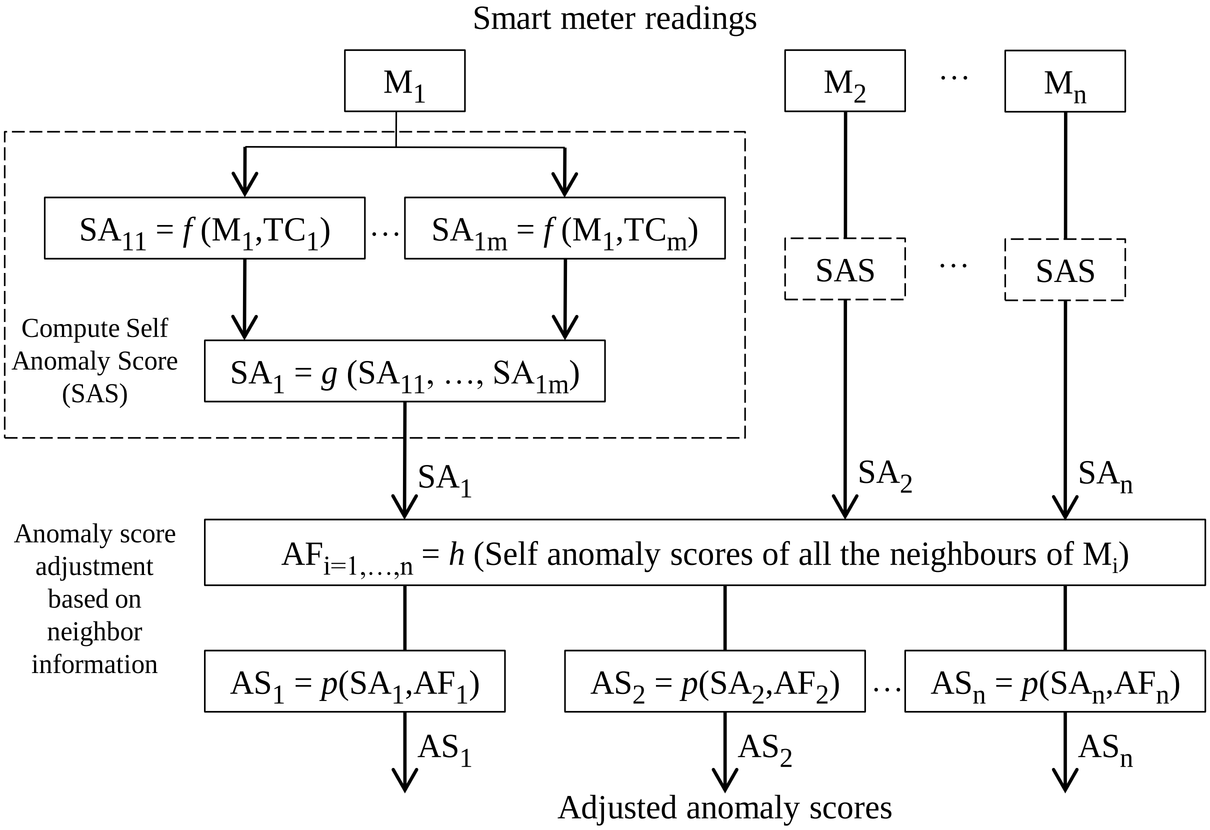

Smart meter readings
M1
M2
...
Mn
SA11 = f (M1,TC1)
...
SA1m = f (M1,TCm)
SAS
...
SAS
Compute Self Anomaly Score (SAS)
SA1 = g (SA11, ..., SA1m)
SA1
SA2
SAn
Anomaly score adjustment based on neighbor information
AFi=1,...,n = h (Self anomaly scores of all the neighbours of Mi)
AS1 = p(SA1,AF1)
AS2 = p(SA2,AF2)
...
ASn = p(SAn,AFn)
AS1
AS2
ASn
Adjusted anomaly scores

Figure 4.2: Logical flowchart of the proposed anomaly detection algorithm. The function f computes self anomaly score for each meter and for each temporal context set separately followed by function g concatenates them. Function p computes the adjusted anomaly score for each meter data based on the available neighborhood information.

applies to this work, is a scalar in the range $[0, 1]$ (low to high severity) that denotes the likelihood of a given data instance being anomalous. The proposed method consists of three steps for computing this score: (1) split the individual meter readings into disjoint sets based on available temporal context information (Section 4.3.1); (2) apply an anomaly detection algorithm separately on each context set and compute an initial anomaly score (Section 4.3.2); (3) adjust the anomaly score for individual meter readings based on available neighborhood information (Section 4.3.3). Figure 4.2 illustrates the workflow of the proposed anomaly detection method.

4.3.1 Splitting data based on temporal context

A suitable period of time in the available data set (for example, the past 60 days) is picked for the analysis described in this section. The proposed algorithm computes anomaly scores for each instance in this time period (for example, for each day in a 60 day period). It can be executed on a rolling basis in order to provide periodic feedback to the end user. The first step in the anomaly detection method involves splitting the given individual meter readings based on the temporal context. As explained in Section 4.2, we use only data timestamps for this classification because they are easily obtainable from existing metering infrastructure. Since human activity typically follows regular temporal patterns, appliance usage is highly correlated to the temporal context. Therefore, computation of anomaly scores based on time-classified data is expected to increase the accuracy of the anomaly detection algorithm.

The specific definitions of temporal context for a physical installation are assumed to be provided by a domain expert. It is assumed that the number of definitions is C, and each definition splits the data set into N disjoint subsets. For example, the building manager for a commercial complex could define a classification with $N = 3$ and a periodicity of one week: *WorkingDay-BusinessHours* [Mon-Fri, 8AM-5PM], *WorkingDay-NonBusinessHours* [Mon-Fri, 5PM-8AM], and *Holidays* [Sat-Sun]. In the absence of temporal context information, it can be obtained automatically by applying existing change point detection methods [77, 116]. The algorithm that we propose thus supports the definition of multiple temporal context sets for the same set of data. For example, the same date/time instance could be classified by the time of the day, day of the week, or season of the year. The self-anomaly detection algorithm described in Section 4.3.2 can be run on each temporal context set separately, and the scores can be merged later.

4.3.2 Self-anomaly score computation

After time series data from each meter has been classified according to a given temporal context definition $c \in \{1, \ldots, C\}$ into N subsets, each subset is processed independently by an anomaly

Algorithm 1: Self anomaly detection algorithm

```
Input: $X^M_{N,L}$: A multivariate time series spanning $D$ days (split into $N$ temporal context
       sets with $L$ slots each) and $M$ meters
Output: $A^m_{n,l}$: Self anomaly score for each time slot
Compute dissimilarity matrix $\Delta^m_n$ using DTW function for all pairs of time slots
  $(x^m_{n,i}, x^m_{n,j})$ within a given temporal class.
Find the optimal number of clusters $P$ in $\Delta^m_n$ using PAM.
Partition $\Delta^m_n$ into clusters $C_p$, $p \in \{1, \ldots, P\}$ using the k-medoid algorithm, compute the
  population of each cluster and save in $\bar{S}^m_n$.
Compute the Euclidean distance vector $\bar{D}^m_n$ from each time slot $x^m_{n,l}$ to the medoid of each
  cluster $C_p$
  forall $x^m_{n,l} \in$ one temporal class and meter $m$ do
      forall $p \in \{1, \ldots, P\}$ do
          $\bar{D}^m_n(p)$ = Euclidean distance ($x^m_{n,l}$, Medoid($C_p$))
      $A^{m*}_{n,l} =< \bar{D}^m_n, \bar{S}^m_n >$
Compute the normalized anomaly score for each $x^m_{n,l}$
```

$$A^m_{n,l} = \frac{A^{m*}_{n,l}}{\max_{i \in tempclass}(A^{m*}_{n,l})}$$

detection algorithm. This initial analysis, referred to as *Self-anomaly detection*, is summarized in Algorithm 1. The input to the algorithm is the time series from one meter and for a single temporal context (e.g. *WorkingDay-BusinessHours*), and the output is an anomaly score for each instance in the time series. Note that the algorithm only requires historical data for this computation and that no neighborhood information has been introduced at this point.

In this work, we denote each series of consecutive meter readings within a single context set as a *power-time cycle* or *time slot*. For example, all meter readings between 0700 and 1800 on a Monday would form one time slot in the *WorkingDay-BusinessHours* set. Let us denote the readings from meter index $m \in \{1, \ldots, M\}$ in a given neighborhood, and within a given time slot, by $x^m_{(n,l,t)}$, where $n \in \{1, \ldots, N\}$ is the index of the context set, $l \in \{1, \ldots, L\}$ is the index of the time slot, and $t \in \{1, \ldots, T\}$ is the time index (minute or hour of the day) within the time slot. We refer to the collective set of all measurements in a single slot $x^m_{n,l}$, to all the time slots within a single context set collectively by the notation $X^M_{n,L}$, and to the entire data set by the notation $X^M_{N,L}$. In this work, we use a classification scheme with $N = 3$ and a single temporal context definition ($C = 1$), as described in Section 4.2.

The algorithm computes the dissimilarity matrix Δ^m_n for all pairs of time slots $(x^m_{n,i}, x^m_{n,j})$, where $i, j \in \{1, \ldots, L\}$ that belong to the same meter and the same temporal class. In order to measure the similarity/dissimilarity between two time slots, we use the *Dynamic Time Warping (DTW)* method [103]. This is a common method for finding the similarity between two time series while accounting for small differences in temporal characteristics. This method is especially useful in the current instance because it compensates for differences in the working hours for two buildings, and also for the shifts caused by daylight saving time (where applicable).

The next logical step is to execute a clustering algorithm on the computed dissimilarity matrix Δ^m_n, thus assigning similar time slots to the same clusters. An unsupervised *k-medoid* clustering algorithm based on Partitioning Around Medoids (PAM) [96] is used for this purpose. The PAM method identifies the optimal number P of clusters in the dissimilarity matrix Δ^m_n, and the k-medoid algorithm subsequently populates the clusters. In contrast with other distance-based clustering algorithms such as k-means, k-medoid algorithm is robust to noise and outliers. Note that the existence of a cluster for a certain set of operating characteristics only implies that these characteristics were observed a significant number of times in the data set. However, the cluster itself may represent an undesirable operating condition from the perspective of a building operator. This issue is discussed further in Section 4.5.

An anomaly score is assigned to each time slot $x^m_{n,l}$ based on the Euclidean distance between this time slot and the medoids of all the clusters, as described in Algorithm 1. The set of Euclidean distances between time slots $x^m_{n,l}$ and each medoid is stored in an P-sized vector $\bar{D}^m_n$. The size (population) of each cluster C_p, $p \in \{1, \ldots, P\}$ is stored in a P-sized vector $\bar{S^m_n}$. Note that $\bar{S}^m_n$ is common to all instances belonging to the same temporal class for a given meter m. The unnormalized anomaly score $A^{m*}_{n,l}$ is then derived by the dot product between $\bar{D}^m_n$ and $\bar{S}^m_n$. In order to compute the relative severity of each anomaly, the final step in the self-anomaly detection algorithm is to normalize by the maximum observed value of $A^{m*}_{n,l}$ within the same temporal

Algorithm 2: Integrated anomaly detection algorithm with neighborhood comparison

```
Input: $X^M_{N,L}$: A multivariate time series spanning $D$ days (split into $N$ temporal context
       sets with $L$ slots each) and $M$ meters
       $A^m_{n,l}$: Self anomaly scores for each instance in $X^M_{N,L}$
Output: $\widehat{A}^m_{n,l}$: neighborhood-adjusted anomaly score for each instance in $X^M_{N,L}$
$A^m_{n,l} = \textsc{ComputeSelfAnomalyScore}(x^m_{n,l})$.
Compute the correlation matrix $\mathbb{C}_n$ of size $M \times L$ between the time series
  $\{x^{m_1}_{n,1}, \ldots, x^{m_1}_{n,L}\}$ and $\{x^{m_2}_{n,1}, \ldots, x^{m_2}_{n,L}\}$ for each pair of meters $(m_1, m_2)$.
Adjust self-anomaly score based on neighborhood
  forall $l \in \{1, \ldots, L\}$ do
      forall $m \in \{1, \ldots, M\}$ do
          $\delta^m_{n,l} = \sum_{k \neq m} \mathbb{C}_n(k, l) A^k_{n,l}$
          $\widehat{A}^m_{n,l} = |A^m_{n,l} - w \times \delta^m_{n,l}|$
```

class for a given meter m. If there are multiple context definitions ($C > 1$), the anomaly scores from each one can be combined to give the collated self-anomaly score $A^m_{n,l}$. The final value of $A^m_{n,l}$ is fed to the neighborhood adjustment algorithm described in the next subsection.

4.3.3 Neighborhood based adjustment

After computing the initial (self) anomaly scores for each time slot $x^m_{n,l}$ from individual meters m, the final step in the algorithm adjusts these scores based on contemporary self-anomaly scores within the neighborhood of m. This adjustment accounts for those unknown contextual factors that influence energy consumption for all meters within a neighborhood in a similar way. An adjustment $\delta^m_{n,l}$ for each instance $x^m_{n,l}$ is calculated based on the original self-anomaly score $A^m_{n,l}$ and the baseline correlation of power consumption between individual members of a neighborhood. The baseline correlation itself is computed using historical comparisons between each pair of meters. As described in Algorithm 2, $\delta^m_{n,l}$ is given by the correlation-weighted anomaly scores from the same time slot (n, l), for all meters within the neighborhood of m. The neighborhood adjusted anomaly score $\widehat{A}^m_{n,l}$ is defined to be the absolute value of the difference between the self-anomaly score $A^m_{n,l}$ and the weighted adjustment factor $w \times \delta^m_{n,l}$. The parameter w is chosen between 0 and 1 which decides how much importance to be for the neighborhood for seasonal adjustments. The optimal value for w can be chosen by a domain expert or calculated empirically, as discussed in Section 4.5.

Intuitively, information from other meters within the neighborhood should help the anomaly detection algorithm to differentiate between the effects of unknown contextual factors and anomalous behavior for a single meter. Unknown contextual factors are expected to produce an effect on several meters within the neighborhood. Therefore, the severity of an observed anomaly should be reduced if multiple other meters also report high self-anomaly scores. On the other hand, the anomalous behavior observed for a given meter but not for other meters within the neighborhood (or vice versa) should result in high anomaly scores. Finally, low self-anomaly scores throughout the neighborhood should result in low anomaly scores overall. We note that the definition of $\widehat{A}^m_{n,l}$ given in Algorithm 2 satisfies all of these requirements.

4.4 Datasets

We evaluate the proposed anomaly detection method using energy meter readings collected from two different geographical locations. The dataset was collected from commercial and residential buildings located in Sweden and India respectively. Thus, these two data sets represent energy meter readings collected from buildings with different operational characteristics along with influence from diverse context factors such as weather conditions.

4.4.1 Commercial buildings

The commercial building data set used for our experiments was collected from a public school campus in Sweden. The school consists of 10 buildings for classrooms and office spaces. These buildings are operated on fairly regular schedules with fixed working days, holidays and daily fixed hours work cycle. The aggregated energy usage of each building within the campus was

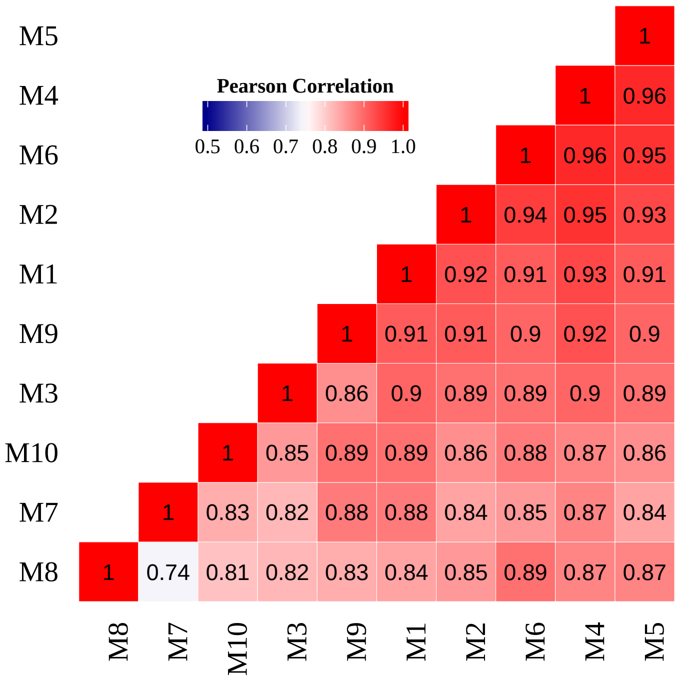


Figure 4.3: The baseline correlation between 10 buildings for a year in the Sweden commercial building data set. Meters are arranged using hierarchical clustering algorithm.

measured separately by smart energy meters. Meters were installed by a third party energy data analytics company for real-time energy monitoring. Aggregated meter readings were sampled at 1 minute to 15 minutes interval and stored in a cloud-based data collection system for more than a year. Figure 4.3 shows the baseline correlation among these buildings for a year.

4.4.2 Residential buildings

This data set contains meter readings from a multi-floor residential complex in India. This building complex consists of 8 floors with 3 apartments in each floor, with a total of 24 apartments in a single building. Most residents in these apartments are working professionals who follow regular office hours. Hence the apartments are typically not occupied during daytime but display high activity during morning and evening. These apartments are equipped with common home appliances such as lighting systems, refrigerator, heaters and air-conditioners. A smart meter was

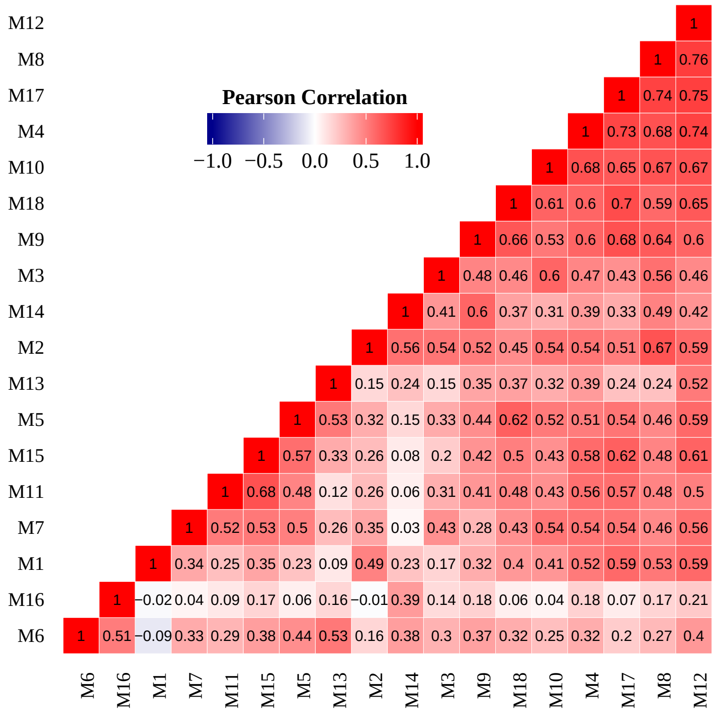


Figure 4.4: The baseline correlation between 18 apartments for a year in the Indian residential buildings dataset. Meters are arranged using hierarchical clustering algorithm.

installed for each apartment separately for energy monitoring. Energy meter readings are sampled at 30 seconds interval and stored in an open-source meter data aggregation system. Among the 24 apartments, we selected 18 for our analysis because there was a lot missing readings for the remaining 6 apartments. Similar to commercial building dataset, we use data spanning one year for our experiments, between August 2013 and July 2014, to account for the seasonal variations. Figure 4.4 shows the baseline correlation among these apartments for a year.

4.4.3 Preprocessing and anomaly injection

The two data sets represent smart meter readings collected from different geographical locations, weather conditions, daily/weekly usage cycles. We down sampled the meter readings to 1-hour resolution (averaged over 1-hour window) for each data set. Because utility companies generally collect meter readings as low as 1-hour resolution and for a valid comparison of the proposed method with an existing method (See Section 4.5.1) which also used 1-hour resolution data. Further, days which contained more than 10% of missing values were excluded from our analysis. For the rest of the missing data (1% for commercial and 15% for residential data), the missing values were replaced by a weighted average of the rest of the day.

Verification procedure for anomalies

Since the analysis presented in this work is based on real historical data, ground truth information about actual anomalies is not readily available. Discussions with the owners of the commercial data set were used to confirm the veracity of certain anomalies in the Swedish data. These confirmed cases are presented in the next section. For residential data, our analysis is restricted to identifying a particular abnormal energy usage event where one or more appliances are operational outside usual hours. An example of such an anomaly would be the operation of heating/cooling loads outside office hours in commercial spaces.

In summary, ground truth comparison in the Swedish data set was carried out through discussion of suspicious events (shown in Figure 4.1) with the facility managers. For the Indian residential building data set, we surveyed the apartment owners and collected the details about all the appliances being used and their power ratings. We manually matched the energy usage of some of the high-power consuming appliances with changes in the raw power meter readings. This exercise was used to model the energy ratings of the appliances. Subsequently, we injected the signatures of 'anomalous usage' of the appliances (left on for few hours to few days) into the raw power readings and marked it as ground truth for our analysis. Table 4.1 summarizes the details about the injected anomaly events. As we use three context sets with a minimum time interval of 12 hours (Section 4.3.1), the required minimum duration of a single anomaly should be 12 hours. To cover the worst case scenario, which is below 12 hours, we injected single anomaly for 6-24 hours. As sequence anomaly occurs in consecutive time slots, we injected the sequence anomaly for 2-3 days (required minimum no of days is two).

Table 4.1: Details about the injected abnormal energy usage events into the residential building dataset.

Appliance	Power (kW)	Anomaly type	No. of instances	Duration
Air conditioner	1.8	single	5	6 - 24 hours
Air conditioner	2.0	sequence	2	2 - 3 days
Room heater	2.2	single	4	6 - 24 hours
Room heater	2.2	sequence	3	2 - 3 days

4.5 Experimental Results

One of the major challenges with evaluating anomaly detection methods is the unavailability of fine-grained ground truth data about the actual anomalies from real-world buildings. Hence we adopt a similar evaluation method presented in [88], and analyze the computed anomaly score case by case for known abnormal energy usage events, such as lights are not turned off during non-working hours. Further, we employed a visual analytics method, similar to [94], for visualizing the actual data and anomaly score in a convenient manner, and to highlight the potential anomalies to the building owners. Therefore, our experimental results are focusing on analyzing the signature of some known anomalies with respect to the temporal context sets and neighborhood information.

4.5.1 Baseline Methods

We compare the performance of the proposed anomaly detection method with an existing algorithm described for commercial building. Additionally, we also compare the performance of the integrated algorithm with two simpler versions that use only self-anomaly detection method.

Self Anomaly - No Context (SANC): This is the self-anomaly detection method described in Section 4.3.2 but without using any temporal context information. Energy meter readings were directly fed into Algorithm (1) without splitting them into temporal context sets. We selected this method to show the significance of using available temporal context information for improving the anomaly scores for known anomaly events.

Self Anomaly, but using Temporal Context (SATC): This is also the self-anomaly detection method described in Section 4.3.2, but now using all the available temporal context information described in Section 4.3.1. However, it does not adjust the computed anomaly score using available neighborhood information. We selected this method to show the significance of using available neighborhood information for adjusting the anomaly score to factor the influence of unknown contexts for a year.

Self Anomaly - HP (SAHP): This anomaly detection method was proposed by Bellala et al from HP for identifying daily anomalous events for commercial buildings [67]. We chose this method because it shares a similar idea with the self-anomaly method but without using any temporal or neighborhood information.

The baseline and the proposed anomaly detection algorithms were implemented in R using its built-in package libraries. Also, we developed a R-Shiny web application and released the code in open-source [40]. We present the identified abnormal energy usage events and compare their performance in the below sections.

4.5.2 Analysis of commercial building data

We executed the proposed anomaly detection method on the Sweden commercial building dataset. We used a temporal context set, as discussed in Section 4.3.1, as the building exhibits fixed daily and weekly cycle. A brief discussion with the data owner revealed that all the buildings are from a single administrative neighborhood as the baseline correlation among these buildings is high, as shown in Figure 4.3. Figure 4.5 illustrates the computed anomaly score of the proposed algorithm along with all the baseline methods for a single smart meter. We selected this particular meter data as it contains many instances of single and sequence anomalies (as shown in Figure 4.1) as compared to other meters.

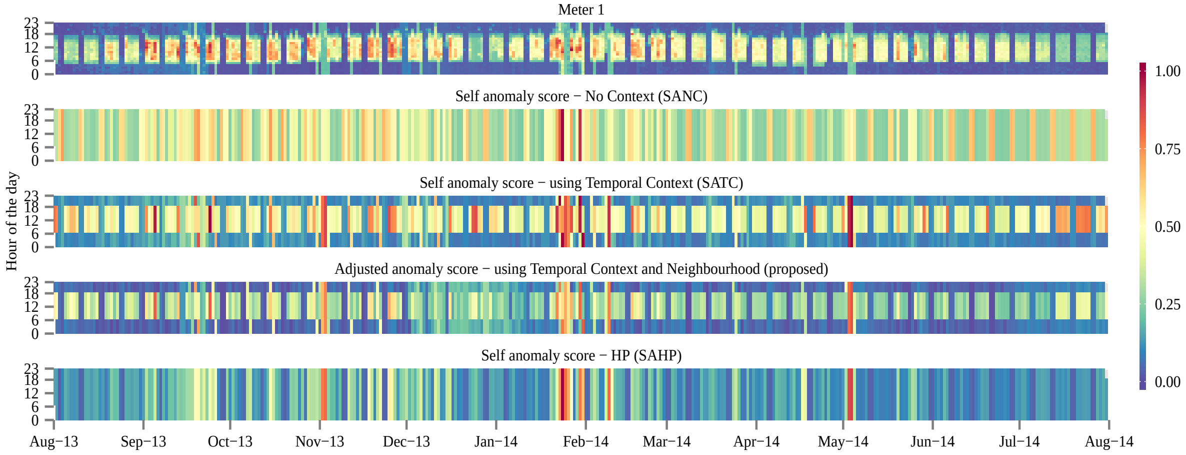


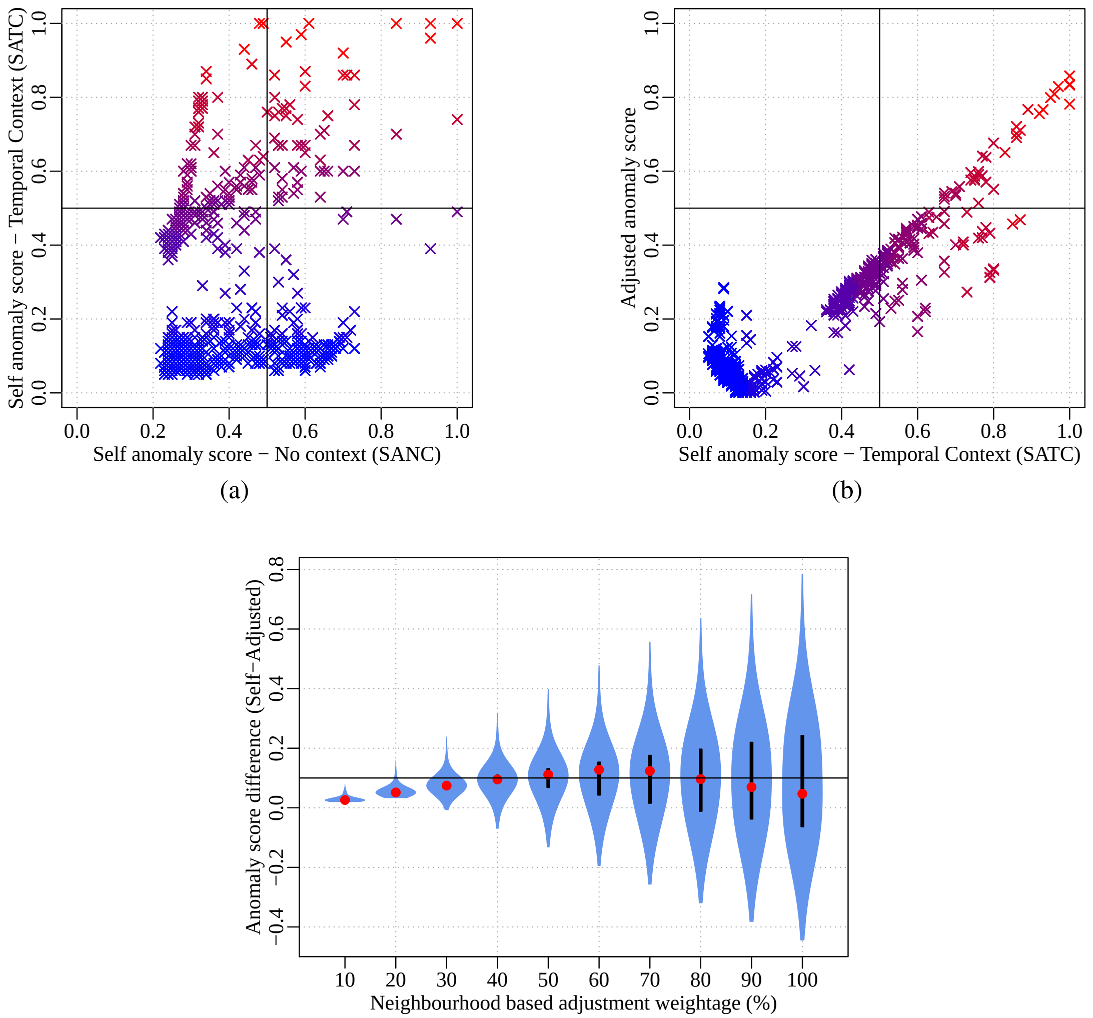
Figure 4.5: Hourly meter readings of a Sweden commercial building with computed anomaly score by different baseline and proposed anomaly detection methods. It shows several instances of point and sequence anomalies and how the computed anomaly score differs using the temporal and neighborhood information.

Single point anomaly

We consider three instances of noticeable single point anomalies in the raw power readings which occurred on September 25, October 4 and November 14, as shown in Figure 4.5. Both SANC and SAHP were not able to assign a high score for them, as the (total) power usage of those anomalous days, although there was anomalous usage during night hours, is similar to other days. However, SATC is able to assign higher score as it splits the power usage of those days into different context sets, *WorkingDay - BusinessHours* and *WorkingDay - NonBusinessHours*. Using the neighborhood information, proposed algorithm decreases the computed anomaly score as the similar usage was observed in some of the other meters as well.

Sequence anomaly

We consider three instances of sequence anomalies (abnormal power consumption during weekends) which occurred during November, February, and May, for evaluating the performance of the proposed anomaly detection method. We can observe that, without using any temporal context information, SAHP outperforms SANC as it assigns higher anomaly score for anomalies



(a)

(b)

(c)

Figure 4.6: Anomaly score comparison of (a) self anomaly score without using any context verses self anomaly score using only the temporal context, (b) self-anomaly score only using the temporal context verses using available neighborhood information, and (c) a violin plot (a combination of box and density plot) shows the differences between anomaly scores (self minus adjusted), by using different adjustment weights for the Sweden commercial building dataset.

happened during November and May. Using the temporal context information, SATC is able to assign a higher score for all the three sequence anomalies. However, after adjustment, it assigns a lower score for the anomaly event that happened in Feb.

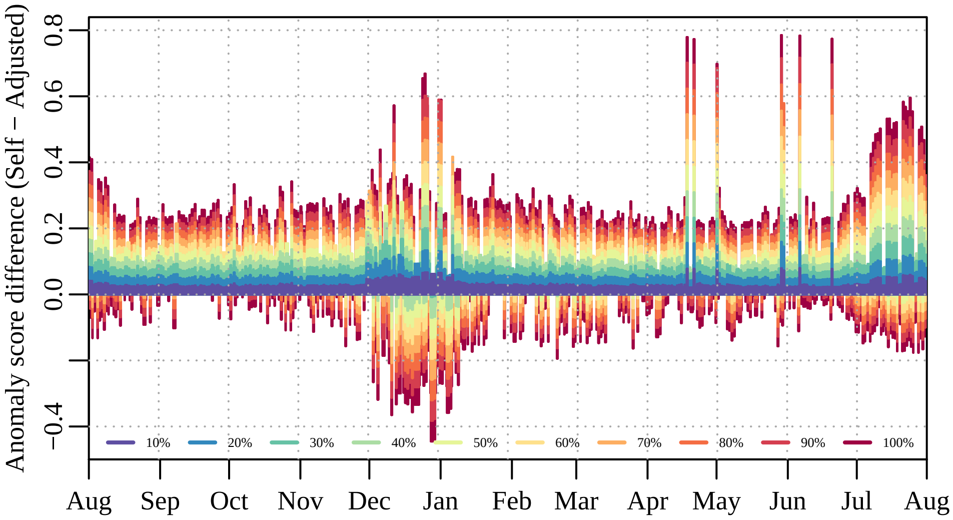


Figure 4.7: Adjusted anomaly score difference for different weights over time for the Swedish commercial building data set. The curve with the smallest magnitude corresponds to a weight of 10% and the one with the highest magnitude corresponds to a weight of 100%. Positive values indicate a reduction in the anomaly score after neighborhood comparison and vice versa.

Anomaly classification

Figure 4.6a shows a scatter plot of self-anomaly scores with and without using temporal context sets. The spread of the scores indicates the influence of the temporal context sets for adjusting anomaly scores based on seasonal changes. Similarly, Figure 4.6b shows a scatter plot of self-anomaly scores using temporal context and neighborhood adjusted scores, for a single meter in the Swedish commercial complex data. This visualization also highlights the unique characteristics of the proposed algorithm, by automatically classifying different classes of anomalies in the data. The lower left corner of the figure represents nominal operating conditions when both the self-anomaly and adjusted anomaly scores are low. The bottom right quadrant represents points that appeared anomalous to the self-anomaly algorithm, but their severity was downgraded after comparison with neighbors. These instances typically represent events such as festival periods or summer vacations (for schools) and are unlikely to be actual anomalies. Instead, these points denote the contribution of the neighborhood comparison step to the reduction of the number of false positives in the anomaly detection algorithm.

The top left quadrant of the figure shows instances that did not appear anomalous at first, but their severity was upgraded after neighborhood comparison. Complementary to the discussion above, these points represent the contribution of the neighborhood comparison towards reducing the false negative rate in the algorithm. Finally, the top right portion of the figure represents points that were deemed to be highly anomalous, both by the self-anomaly detection and after neighborhood comparison. These points represent the most confidently flagged anomalous instances and should be investigated by human supervisors on a high priority basis.

Anomaly score adjustment using neighborhood information

After computing the initial anomaly score, we assigned a different percentage of weights for the neighbors and calculated the adjusted anomaly score. Figure 4.6c illustrates the difference between self and the adjusted anomaly score while using different weights ranging from 10% to 100%. It is observed that up to assigning weights 60% the average difference between the self and adjusted score is increased linearly. After that, it started to decrease for weights higher than 60%, as shown inFigure 4.6c. Figure 4.7 shows how adjustment varies over time for different seasons with different neighborhood weights. We can observe that higher adjustment during January for accounting the seasonal changes. Also, there are some instances of higher adjustment during May to account for the abnormal energy usage events which are also visible in Figure 4.5.

4.5.3 Analysis of residential building data

We executed the proposed anomaly detection algorithm over all the 24 smart meter readings in the Indian residential building data set. We used a temporal context set, as discussed in Section 4.3.1, as the building occupants exhibit regular daily and weekly cycle similar to the commercial buildings. Further, all the buildings were from the same neighborhood. Figure 4.8 illustrates the computed anomaly score of the proposed algorithm and the baseline methods for a single smart meter. We injected the anomalies, shown in Table 4.1, into one of apartment me-

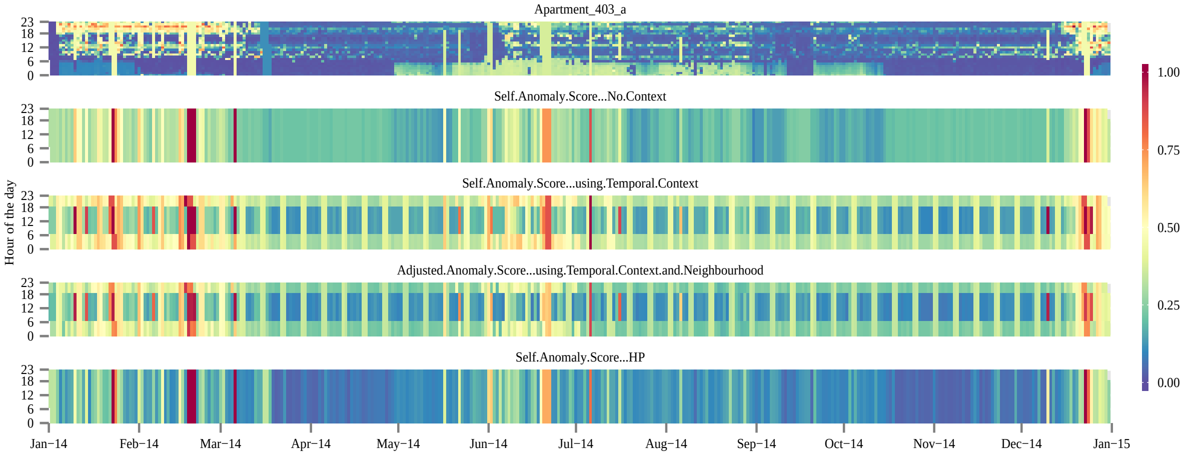


Figure 4.8: Hourly meter readings of an Indian residential building with computed anomaly score by different baseline and proposed anomaly detection methods. It shows several instances of point and sequence anomalies and how the computed anomaly score differs using the temporal and neighborhood information.

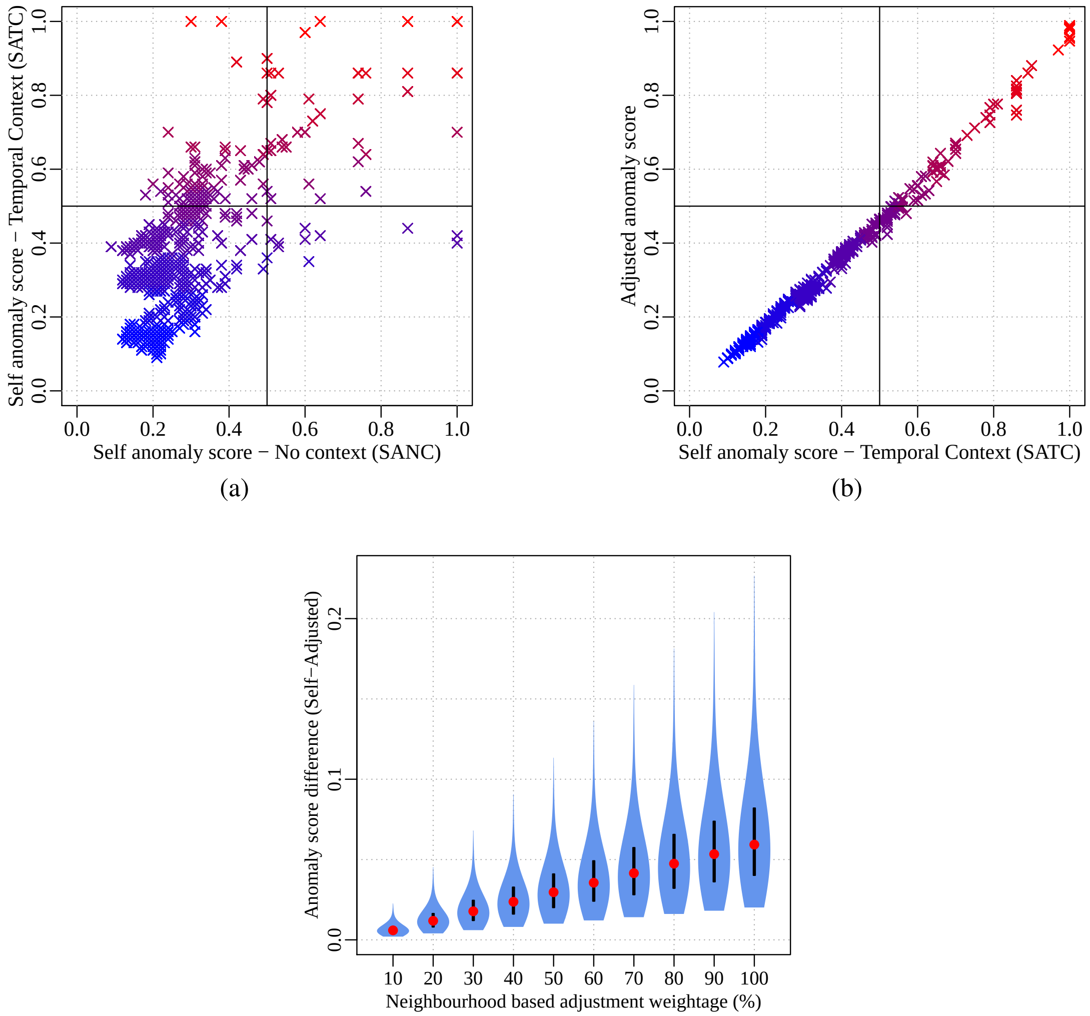
ter data. Also, we applied a temporal context set which is similar to the commercial building experiment to account for the daily/weekly energy usage cycle.

Self anomaly score

In contrast with commercial buildings, variation in the energy usage patterns are high in residential buildings. As shown in Figure 4.4, the baseline correlation between residential apartments was diverse. Specific to India, as shown in Figure 4.8, there is higher power consumption during summer and winter due to the extreme weather conditions. After calculating the anomaly scores, we set a threshold of selecting top 10% of anomalies for the analysis. Among those 14 anomalies that we injected, SATC was able to assign a higher score for all of them, whereas SAHP identified only 12 (missed those single point anomalies happened during July and August).

Anomaly score adjustment using neighborhood information

Similar to the commercial building, we assigned a different percentage of weights for the neighbors and calculated the adjusted anomaly score. Figure 4.9c plots the difference between self-anomaly score and adjusted using different percent of weights ranging from 10% to 100%. In



(a)

(b)

(c)

Figure 4.9: Anomaly score comparison of (a) self anomaly score without using any context verses self anomaly score using only the temporal context, (b) self-anomaly score only using the temporal context and using available neighborhood information, and (c) violin plot (a combination of box and density plot) shows the differences between anomaly scores (self minus adjusted), by using different adjustment weights for the Indian residential building dataset.

contrast with commercial buildings, the baseline correlation between the residential apartments is diverse, as shown in Figure 4.4. Due to that the maximum adjustment factor was 0.2. It is observed that the difference between the self and adjusted score is increased linearly with respect to neighborhood weights. Figure 4.10 shows how the adjustment varies over time for different seasons while using different neighborhood weights. We can observe that increase in the adjustment during July to account for the seasonal changes during summer.

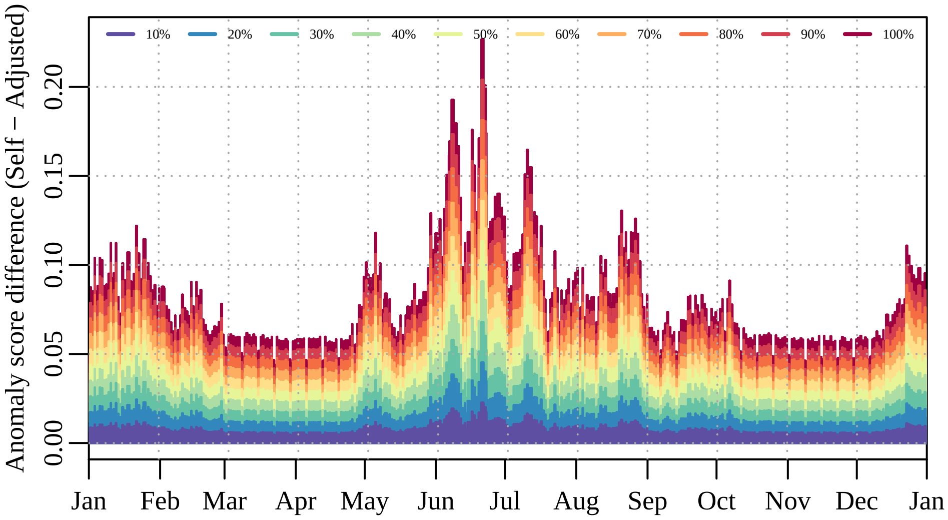


Figure 4.10: Adjusted anomaly score difference for different percentage of weights over time for the Indian residential building data set. The curve with the smallest magnitude corresponds to a weight of 10% and the one with the highest magnitude corresponds to a weight of 100%. Positive values indicate a reduction in the anomaly score after neighborhood comparison.

Similar to the anomaly characterization for the commercial building data set, Figure 4.9a shows a scatter plot of self-anomaly scores with and without using temporal context sets. Similarly, Figure 4.9b plots the self-anomaly score against the adjusted anomaly score which shows the classification of anomaly events. In contrast with the commercial building, the scatter plot looks narrow because of the higher variation in the baseline correlation in residential building users.

4.6 Related Work

Several anomaly detection methods for different domains have been proposed in the literature for identifying potential anomalous events [73]. Authors in [114] proposed an unsupervised anomaly detection method for identifying potential abnormal energy usage days. They used robust statistical methods based on the variability of mean and standard deviation in the power usage. Similar to that, authors in [70] proposed a generalized extreme studentized deviate method based on the variation of mean and standard deviation of the measured data. However, these methods are limited to identify the single point anomaly events.

Authors in [74] proposed a framework for detecting energy usage outliers for smart buildings. They used a suffix tree representation of the energy usage activities for grouping the subsequence energy usage events and cluster them for identifying anomalous events. A similar method but without using clustering algorithm was proposed in [67]. They use a multi-dimensional scaling method for reducing the dimension of the dissimilarity matrix and assign density of a point using the kNN algorithm.

Several multivariate anomaly methods have been proposed in the literature. Authors in [78] presented a graph-based algorithm for detecting and characterizing the anomalies. Since many real-world events are interrelated, Granger causality methods have also been used in the literature for time series anomaly detection [109]. However, all these methods do not account the context factors which influence the occurrence of anomaly events. An anomaly detection framework for monitoring the Hadoop Map-reduce tasks using system-level context information was proposed in [91]. Our proposed self-anomaly detection algorithm shares similar idea with them for using the context information. However, the adjustments of self-anomaly scores across the meters in a neighborhood are essential for energy domain for accounting the seasonal changes.

In contrast with all the existing methods, the proposed method uses the context information which is collected as part of the meter data collection system. Further, we evaluate the proposed method using a year-long data set from both commercial and residential buildings.

4.7 Summary

In this chapter, we described an anomaly detection methodology for energy consumption data time series, that used information from neighboring meters to qualify its output. The neighborhood for a meter (the source of each time series) could be defined a prior or could be identified directly from the data. We showed that incorporating such information was an effective safeguard against the identification of spurious anomalies (false positives), as well as against the omission of real anomalies (false negatives). The generic nature of the algorithm ensures that it is effective for a wide-range of applications, including residential and commercial complexes.