CHALMERS
UNIVERSITY OF TECHNOLOGY

LoCoVolt: Distributed Detection of Broken Meters in Smart Grids through Stream Processing

Downloaded from: https://research.chalmers.se, 2025-06-18 03:14 UTC

Citation for the original published paper (version of record):

Van Rooij, J., Gulisano, V., Papatriantafilou, M. (2018). LoCoVolt: Distributed Detection of Broken Meters in Smart Grids through Stream Processing. DEBS 2018 - Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems: 171-182.
http://dx.doi.org/10.1145/3210284.3210298

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Industry Paper: LoCoVolt: Distributed Detection of Broken Meters in Smart Grids through Stream Processing

Joris van Rooij
Chalmers University of Technology
and Göteborg Energi
Gothenburg, Sweden
jorisv@chalmers.se;joris.vanrooij@goteborgenergi.se

Vincenzo Gulisano
Chalmers University of Technology
Gothenburg, Sweden
vincenzo.gulisano@chalmers.se

Marina Papatriantafilou
Chalmers University of Technology
Gothenburg, Sweden
ptrianta@chalmers.se

ABSTRACT

Smart Grids and Advanced Metering Infrastructures are rapidly replacing traditional energy grids. The cumulative computational power of their IT devices, which can be leveraged to continuously monitor the state of the grid, is nonetheless vastly underused.

This paper provides evidence of the potential of streaming analysis run at smart grid devices. We propose a structural component, which we name LoCoVolt (Local Comparison of Voltages), that is able to detect in a distributed fashion malfunctioning smart meters, which report erroneous information about the power quality. This is achieved by comparing the voltage readings of meters that, because of their proximity in the network, are expected to report readings following similar trends. Having this information can allow utilities to react promptly and thus increase timeliness, quality and safety of their services to society and, implicitly, their business value. As we show, based on our implementation on Apache Flink and the evaluation conducted with resource-constrained hardware (i.e., with capacity similar to that of hardware in smart grids) and data from a real-world network, the streaming paradigm can deliver efficient and effective monitoring tools and thus achieve the desired goals with almost no additional computational cost.

CCS CONCEPTS

• **Information systems** → **Data streams**; • **Hardware** → **Energy metering**; **Smart grid**;

KEYWORDS

Data Streaming, Advanced Metering Infrastructure, Smart Meter

1 INTRODUCTION

Smart Grids, in which communication-enabled IT devices can share information with energy utilities, are replacing traditional energy grids. At the lower-voltage distribution tiers (i.e., at the level where energy is distributed to private customers and businesses), this transformation has been enabled by Advanced Metering Infrastructures (AMIs).

These consist of Smart Meters (SMs) and a communication infrastructure to communicate with the energy utilities' data center. A common communication infrastructure consists of Concentrator Units (CUs) that use either a wireless radio network or power line communication to communicate with the SMs [20].

In any large system continuous monitoring is needed to detect faulty components. This is especially important in an AMI because of safety-related, economic and administrative implications. To our advantage, AMIs make available the cumulative distributed computational power of their devices, which can be used to efficiently monitor the state of an AMI in a continuous and distributed fashion. In this context, data streaming fits well thanks to its inherently distributed, parallel and low-latency analysis properties.

In this paper, we show this for the detection of broken SMs that report incorrect voltage readings. Incorrect voltage readings can indicate that the SM is broken and will influence the amount of energy billed to the customers. Furthermore, early detection of broken SMs boosts safety, since undetected high voltages can cause damage to electric equipment or injury to persons. We present LoCoVolt (Local Comparison of Voltages). With LoCoVolt the differences of voltage readings between any pair of close-in-space SMs are continuously monitored by each CU. Based on the observed differences (instantaneous and average) and the correlations between voltage readings from each SM connected to the same CU, an SM is reported as broken when its number of suspicious readings exceeds a given threshold (as we explain in detail in § 4) . LoCoVolt can function in a manner that is agnostic of the precise topology of the underlying electricity network and this property makes its approach applicable in a variety of deployments.

We implemented LoCoVolt on top of Apache Flink [7] and tested it with real-world electricity network data from a deployed network. Our evaluation, conducted on resource-constrained hardware whose capacity resembles AMI hardware capacity, shows that the streaming paradigm delivers efficient and effective monitoring tools.

The rest of the paper is organized as follows. § 2 overviews preliminary concepts. § 3 describes the problem in more detail, explaining also the bigger challenges. LoCoVolt's technique to detect broken SMs is presented in § 4 and evaluated in § 5. § 6 discusses related work while § 7 concludes the paper.

2 PRELIMINARIES

In this section we overview introductory and background information about data streaming applications, AMIs and voltage monitoring, as well as correlation measures between data streams.

2.1 Data streaming processing applications

Data streaming processing applications are designed as graphs composed by *streams* of data and *operators*.

Each *stream* carries tuples sharing the schema $\langle ts, A_1, \ldots, A_n \rangle$, where ts is the tuple's creation timestamp and $A_1, \ldots, A_n$ are application-related attributes. In a DAG, streams specify how tuples flow from the data sources through the operators and, eventually, to the data sinks (delivering results to analysts or other applications).

Operators are provided by the Stream Processing Engine (SPE) being used to run the application. Despite the fact that each SPE provides its own definition (and implementation) of basic streaming operators, a common subset of the operators provided by different SPEs includes *Aggregate*, *Join*, *Stateless* and *Merge* operators [9].

Aggregate operators apply aggregation functions over *sliding windows* of tuples. Windows are defined by their size, their advance and, optionally, by a group-by parameter referring to one or more of the input tuples' attributes when the aggregation function is applied independently to each group of tuples sharing such attributes. The Join operator matches tuples from two streams (keeping a sliding window for each stream) and forwards the pairs for which a given predicate holds. Stateless operators, as the name suggests, do not maintain a state evolving with the tuples being processed, and can produce zero, one or more output tuples for each input tuple, applying a user-defined function that specifies the input tuples' attributes to be copied to the output tuples and the functions to be applied to them. Finally, Merge operators allow to merge multiple streams into a single one. As we discuss in Section 4, these basic operators can be composed to implement LoCoVolt's analysis.

2.2 AMIs and voltage monitoring

Electricity Network: Private customers and businesses, along with their SMs, are connected to the grid via *transformers*. Each SM connects to exactly one transformer while a single transformer can host multiple SMs. Each SM is connected to a transformer by one, two or three lines as well as a neutral.

Data Network: At the same time, each *Smart Meter* SM is also connected to the utilities' servers, often via a *Concentrator Unit* (CU) that aggregates data from multiple SMs. Each SM connects to exactly one CU (but can change it over time) while a single CU is connected to multiple SMs. SMs that are physically close (e.g., deployed in the same building) have high chances of being connected to the same transformer and CU. In the case of wireless communication between SMs and CU there is nonetheless no guarantee about the overlap between the SMs and transformers topology and the SMs and CUs topology (e.g., two physically close SMs could for instance be connected to different transformers but the same CU). This is illustrated in the schematic overview in Figure 1.

The voltage measured by the SMs depends on the input voltage at the transformer, the length of the connecting cable connecting and local loads in the distribution network. SMs that are not broken and are connected to the same transformer are thus expected to display a high correlation between their voltages time series [17]. Figure 2 illustrates this by showing the voltages measured by two SMs that are physically close and connected to the same transformer.

Among other things, SMs report the voltage readings for each of their lines with tuples composed by attributes $\langle ts, CU, SM, L \rangle$, where *ts* denotes the timestamp for the reading, CU and SM indicate the corresponding device identities and L is a 3-entry vector, containing the root-mean-square voltage readings for the respective lines, i.e. the equivalent steady (DC) value which gives the same effect as the sinusoid signal [4].

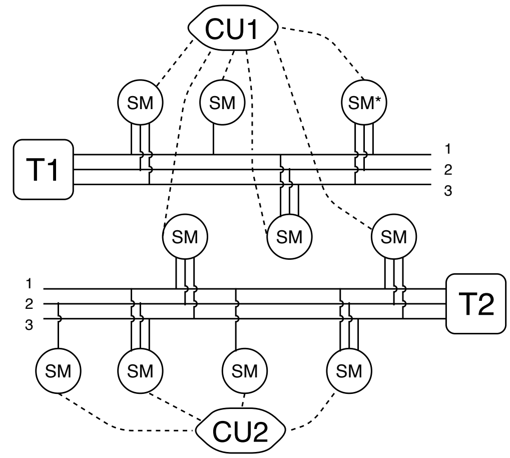

Figure 1: A schematic overview of the SMs connected to two transformers and two CU. Note that the data network overlaps only partially with the electricity network. SMs are connected to the transformer with at least one and at most three lines. The line order at the SMs can differ, e.g. here the line order of SM* is reversed compared to the other SMs. Besides, the set of SMs connected to each CU may vary with time, e.g. depending on conditions that affect the signal strength.

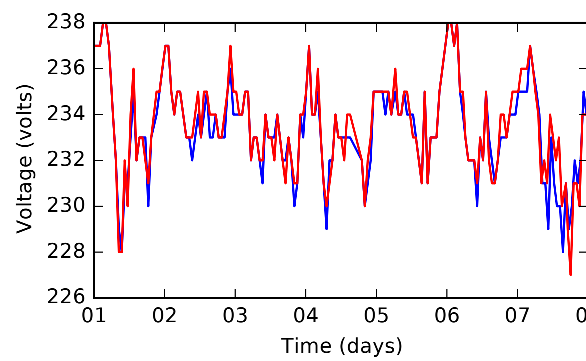

Figure 2: The voltage for two SMs that are physically close and connected to the same transformer during one week. The correlation coefficient for the time series is 0.94.

2.3 Streams correlation

In LoCoVolt, we make use of correlation of time series. Given two time series a and b with n elements, the Pearson correlation coefficient, r, is suitable to be calculated incrementally [21] as:

$$r_{a,b} = \frac{P - \frac{AB}{n}}{\sqrt{A_2 - \frac{A^2}{n}}\sqrt{B_2 - \frac{B^2}{n}}},$$

where $A = \sum_i a_i$, $B = \sum_i b_i$, $A_2 = \sum_i a_i^2$, $B_2 = \sum_i b_i^2$ and $P = \sum_i a_i \cdot b_i$. For incremental calculation only the sums and the number of values need to be stored. The range of r is $[-1, 1]$, where -1 indicates maximum negative correlation and 1 maximum positive correlation. A correlation coefficient of 0 indicates that there is no correlation between the time series.

3 PROBLEM DESCRIPTION

In AMIs, SMs are expected to report consumption and quality measurements periodically. Over time, SMs can break and stop reporting or start reporting measurements that are not correct. An SM is defined to be broken when its measured voltage differs from the actual voltage. Detection of broken SMs is challenging since variations in consumption readings cannot be easily distinguished between customer-dependent variations and variations dependent on broken SMs. Because of the physical properties of electricity, a broken SM can be potentially identified by comparing the voltage readings of its *lines* with the ones of the lines of a working SM connected to the same transformer (as explained in Section 2 they are expected to have readings that follow the same temporal curves) [17].

LoCoVolt aims at detecting broken SMs based on this observation. It should be noticed that several practical aspects of AMI deployments can make this a challenging task:

(1) *Line ordering*: SMs' lines are not connected to transformers in a fixed order. That is, the same physical line connecting two SMs to the same transformer is not necessarily plugged as line 1 (or 2 or 3) in both SMs.
(2) *Asynchronous measurements*: The primary task of SMs is to measure the consumed energy. For this reason SMs don't necessarily take voltage readings in a synchronized fashion, voltage readings can even be measured by the CUs sequentially with a coarse-grained periodicity (e.g. 15 minutes, 1 hour). This implies that we might not have simultaneous measurements at all SMs and hence the differences might be inaccurate. Moreover, due to intermittent connectivity and noise, not all readings made in nearby time-intervals reach the CU in time (i.e. the continuous analysis must tolerate missing values).
(3) *Symmetric differences*: A suspicious difference between the voltage readings of the lines of two SMs indicates that one of the two is potentially broken, but does not give any hint about which one of the two is the broken one.
(4) *Electricity network and communication network topology*: The topology of the electricity network (i.e. which SMs are on the same transformer) might not be known, due to distinctions between administrative domains of the utility or for security reasons. Moreover, wireless communication may imply that the set of SMs connected to the same CU is dynamic. (cf. Figure 1). I.e. we are looking for a method that is agnostic to the transformer - SM network, as mentioned in the introduction.

From an implementation perspective, an additional challenge is to employ methods that are intuitive to communicate with the utility system experts, as unnecessary complexity introduces risks of errors due to misunderstandings. For that reason, we propose to analyze the problem so as to use standard SPE operators, usable by engineers with common programming skills.

4 LOCOVOLT

LoCoVolt addresses the challenges described in the previous section, through continuous monitoring of voltage readings of SMs at the CU level. Following LoCoVolt's rules, SMs can "accuse" each other if their voltage readings are suspiciously far. On a CU containing some broken SMs, this will result in many-to-few accusations (from the working SMs to the broken ones) and few-to-many accusations (from the broken SMs to the working ones). Accusations made by an SM M_i are weighed by the number of accusations it receives, introducing a reliability measure. The weighted accusations for M_i are then summed and compared with the number of SMs expected to accuse M_i is the latter is broken. In the following, we discuss LoCoVolt's semantics. We present the stream operators (Figure 3) that implement them and an example in the following subsections (subsections' names correspond to the boxes in the figure, the stream operators are also referenced in the text).

When do SMs accuse each other? The difference measured by two SMs fluctuates over time due to the reasons mentioned in Section 2. Hence, rather than having SMs accuse each other when their readings are arbitrarily distant, we base accusations on the difference between their instantaneous (ΔV) and average ($\overline{\Delta V}$) difference. Furthermore, we weigh such accusations by the correlation (c) observed between them, so that accusations from SMs observing similar differences over time count more than those from SMs observing fluctuating differences. Two SMs accuse each other when:

$$|\Delta V - \overline{\Delta V}| \cdot c_{a,b} > \theta$$

where (i) θ is a parameter related to the measurement resolution, the accuracy and local load changes occurring between the readings of a pair of SMs; and (ii) $c_{a,b} = (r_{a,b} + 1)/2$, i.e. a mapping of the correlation of formula 2.3. This mapping of the correlation preserves the monotonic relation between the correlation and $\overline{\Delta V}$. The mapping also allows uncorrelated and inversely correlated SMs to accuse each other if $|\Delta V - \overline{\Delta V}|$ grows large enough. Systems experts maintaining the smart grid used in our evaluation identified a suitable value for θ from a set of working SMs. Since SMs can have up to three lines, $\Delta V, \overline{\Delta V}$ as well as c are 3x3 matrices. Therefore, the number of accusations between two SMs, M_i and M_j, is an integer in the range [0,9]. Notice that accusations are symmetric, i.e. if M_i accuses M_j, then M_j accuses M_i too.

How to prevent accusations by SMs from different transformers from affecting broken SMs detection? When M_i and M_j are not connected to the same transformer, their voltage reading difference can fluctuate more than if they share the same transformer (independently of whether any of the two is broken). When two big-enough groups of SMs are connected to the same CU but different transformers, this can result in many-to-many accusations (because of the legit different voltage readings across groups). To mitigate this, we normalize accusations weighting them by:

$$w_i = \frac{1}{\sum_{S_j} acc(j,i)}$$

where w_i is the weight for M_i, S_j is the set of SMs accusing M_i, and $acc(j,i)$ is the number of accusations between M_j and M_i. The value of w_i is small for SMs that receive a large number of accusations, indicating that it is less trustworthy. The weighted accusation received by M_i from M_j is then defined as:

$$wacc(i,j) = w_j \cdot acc(i,j).$$

Each $wacc()$ is a real number in the range [0,1]. M_i weighted accusation from M_j is 1 when M_i is the only SM accusing M_j. Note that while the accusations are symmetric between M_i and M_j, this does not necessarily hold for the weighted accusations. The total amount of weighted accusations can now be calculated for all SMs by:

$$J_i = \sum_S wacc(i, j)$$

where S is the set of SMs whose readings are compared with M_i.

If an SM breaks, how many other SMs at its CU will accuse it? Value J_i depends not only on how much M_i's readings deviate from the other SMs, but also on the number of SMs that the reading is compared with, as well as how correlated these SMs are with M_i (i.e., how likely they will trigger an accusation). When M_i breaks and starts reporting inaccurate values, the probability of receiving an accusation from M_j will be related to the correlation between M_i and M_j, with an increasing probability for increasing correlations.

We estimate E_i, the expected number of SMs accusing M_i, by the likelihood of receiving at least one accusation from another SM, which we approximate through the correlation matrix entries $c_{i,j}$. More concretely, E_i is defined as:

$$E_i = \sum_S max(c_{i,j})$$

Notice that each SMs in S contributes to the sum with value 1 if its correlation with M_i is 1. A weight θ' is introduced to specify which portion of E_i is sufficient for an accused SM to be reported as broken. The value of the weight influences the number of simultaneous broken SMs that can be detected since a larger number of broken SMs will cause *wacc* to decrease. The weight is set by a system expert to match the expected maximum number of simultaneous broken SMs. An alert is eventually triggered for M_i if $J_i > \theta' E_i$.

The resulting graph of streams and operators implementing the method is presented in detail in Figure 3 as well as in the following sections. The names of the sections correspond to the boxes in the figure and the stream operators in the figure are referenced in the text with (OperatorID). A running example in the text is used to illustrate the query. The example consists of three SMs, A, B and C on CU X, where C is accused by A and B. For simplicity, all SMs have a single line.

4.1 Input

As described in Section 2.2, the highest correlation between voltage time series will be between SMs that are close in the distribution network. Therefore LoCoVolt compares SM readings that are within a certain time-window and connect to the same CU. This can be implemented by joining the stream of measurements onto itself in a window. The size of the window (w_s), and advance (w_a), is a trade off between the number of pairs that can be compared and the correlation between the readings. Here it is set to 10 minutes with the help of a system expert and the window is evaluated for every incoming tuple.

Since an SM can report values for 1, 2 or 3 lines, the readings are stored in an array of size 3. If there are fewer than three values the corresponding place in the array will be set to a predefined null value.

The example starts with the following three readings:

$$< ts_A, X, A, 230 >, < ts_B, X, B, 231 >, < ts_C, X, C, 219 >,$$

where $ts_A < ts_B < ts_C$. When all timestamps are within the window for the Join operator, the following tuples will be produced:

$$< ts_B, X, A, B, 230, 231 >, < ts_C, X, A, C, 230, 219 >,$$
$$< ts_C, X, B, C, 231, 219 > .$$

4.2 Statistics

The statistics required for LoCoVolt are the correlation (c) and the average difference ($\overline{\Delta V}$) between all line pairs of all SMs. This can be implemented with an aggregation operator that incrementally calculates the average for the difference as well as the correlation in a window. The window size should be large so that $\overline{\Delta V}$ and c are stable values. The window size for LoCoVolt is set to 28 days with a window advance of 14 days.

Since the readings are arrays with length three, the average difference, as well as the correlation between all possible line pairs are stored in two 3x3 matrices. (A1)

For the example we assume that the correlation between SMs A and B is 0.91 while the average difference is 1.2. Values for the other pairs can be found below.

$$< ts, X, A, B, 0.91, 1.2 >, < ts, X, A, C, 0.87, 2.3 >,$$
$$< ts, X, B, C, 0.89, 2.8 > .$$

4.3 Accusations

The calculation of the accusations is accomplished by an operator that calculates ΔV for all line pairs (S1). This stream is then joined with the statistics stream with a window size and advance that equals the window advance for the statistics operator. The join function can now calculate the value of $|\Delta V - \overline{\Delta V}| \cdot c$. (J2)

$$< ts_B, X, A, B, 0.182 >, < ts_C, X, A, C, 7.569 >,$$
$$< ts_C, X, B, C, 8.188 > .$$

The threshold θ can now be applied by a stateless operator. (S2) θ is set to 3, which for the example results in the following tuples:

$$< ts_B, X, A, B, 0 >, < ts_C, X, A, C, 1 >, < ts_C, X, B, C, 1 > .$$

The number of accusations between any pair of SMs is symmetric and can be processed in a single tuple containing the IDs for both SMs as well as the number of accusations. This is a number between 0 and 9, depending on the number of lines for the SMs and the result from the previous equation. In order to get the sum of the accusations per SM, two tuples with the number of accusations are created, one for each SM (S3).

$$< ts_B, X, A, 0 >, < ts_B, X, B, 0 >,$$
$$< ts_C, X, A, 1 >, < ts_C, X, C, 1 >,$$
$$< ts_C, X, B, 1 >, < ts_C, X, C, 1 > .$$

These tuples are aggregated in a tumbling window and counted (A2). The window size is chosen to match the frequency of the readings which in our case are hourly.

$$< ts_C, X, A, 1 >, < ts_C, X, B, 1 >, < ts_C, X, C, 2 > .$$

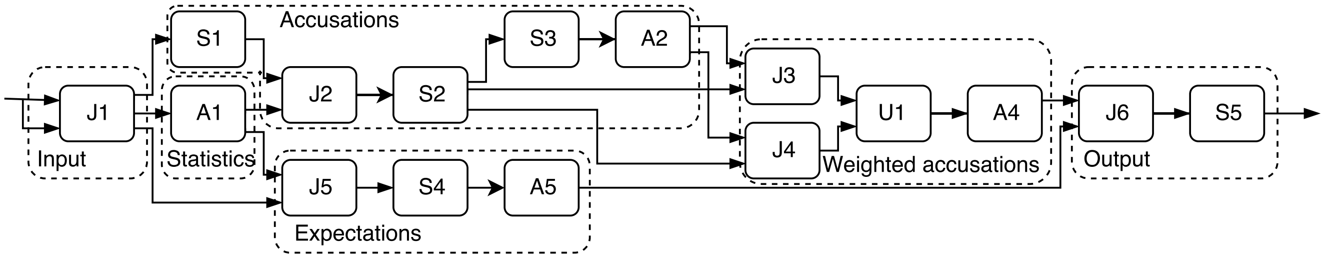

Figure 3: The graph for LoCoVolt. The letter in the operator name refers to the type of operator: J **for Join,** A **for Aggregate,** S **for Stateless and** U **for Union.**
The operators are arranged in dashed logical blocks. The function and details of the logical blocks are described under the equally named headings in Section 4. Details for the individual operators can be found in Table 1.

Operator	Description	Tuple schema
J_1	Match any pair of tuples from two different SMs that share the same CU over a sliding window of 10 minutes, outputting the SMs' IDs and their voltage readings per line (3X1 matrices L_1 and L_2).	$\langle ts, CU, m_1, m_2, L_1[3], L_2[3]\rangle$
S_1	Produce a 3X3 matrix (ΔV) with the difference between each pair of lines between two SMs joined by J_1.	$\langle ts, CU, M_1, M_2, \Delta V[3X3]\rangle$
A_1	Produce a 3X3 matrix (c) with the scaled correlation (Pearson coefficient) as well as a 3X3 matrix ($\overline{\Delta V}$) with the average difference observed for each pair of lines between two SMs joined by J_1 over a sliding window of 28 days with an advance of 14 days.	$\langle ts, CU, M_1, M_2, c[3X3], \overline{\Delta V}[3X3]\rangle$
J_2	Match tuples referring to the same pair of SMs and produce a 3X3 matrix (D) carrying the value $c \circ \lvert\Delta V - \overline{\Delta V}\rvert$, where $\circ$ denotes the entrywise product.	$\langle ts, CU, M_1, M_2, D[3X3]\rangle$
S_2	Applies a threshold to the elements in D, setting the element to 1 if the value exceeds the threshold and 0 otherwise.	$\langle ts, CU, m_1, m_2, D[3X3]\rangle$
S_3	Produce the sum of the elements in D that equal 1 (accusations, acc) and output this in two tuples, one for each SM in the input tuple.	$\langle ts, CU, M, acc\rangle$
A_2	Produce the sum of acc for all SMs on all CUs in a 1 hour tumbling window.	$\langle ts, CU, M, \sum acc\rangle$
J_3	Match tuples where M from A_2 equals M_1 from S_2. Produces a tuple for M_2 containing weighted accusations $wacc$.	$\langle ts, CU, M_2, wacc\rangle$
J_4	Similar to J_3 but matches tuples where M equals M_2. Produces a tuple for M_1.	$< ts, CU, M_1, wacc >$
U_1	Union of the outputs from J_3 and J_4.	$\langle ts, CU, M, wacc\rangle$
A_4	Produce the sum of $wacc$, (J) for all SMs in a 1 hour tumbling window.	$\langle ts, CU, M, J\rangle$
J_5	Matches tuples referring to the same pair of SMs in a sliding window of 14 days, removes $\overline{\Delta V}$ from the tuple..	$\langle ts, CU, M_1, M_2, c[3X3]\rangle$
S_4	Splits the input tuple into two, one for each SM and keeps containing the maximum element in c.	$\langle ts, CU, M, max(c)\rangle$
A_5	Produces the expected number of accusations (E), in a 1 hour tumbling window.	$\langle ts, CU, M, E\rangle$
J_6	Join A_4 and A_5 for each SM.	$\langle ts, CU, M, J, E\rangle$
S_5	Report suspicious SMs for which the number of J exceeds $\theta' E$.	$\langle ts, CU, M\rangle$

Table 1: Detailed description of the Join, Aggregate, Stateless and Union operators used in LoCoVolt's query.

4.4 Weighted accusations

The weighted accusations are obtained by combining the accusations between the SMs with the sum of the accusations per SM. Two join operators are required to accomplish this since the accusations between the SMs is stored in a single tuple for every pair of SMs, while the total number of accusations received has one tuple per SM. (J3,J4) The window size and advance of the join operators match the frequency of the readings. The union of these streams results in a single stream with all *wacc* values. (U1)

The tuples at this point in the query contain *wacc* as shown in the example:

$$< ts_B, X, A, 0/1 >, < ts_B, X, B, 0/1 >,$$
$$< ts_C, X, A, 1/2 >, < ts_C, X, C, 1/1 >,$$
$$< ts_C, X, B, 1/2 >, < ts_C, X, C, 1/1 > .$$

The tuples now need to be aggregated in a window and counted in order to obtain J (A4). The window size and advance match the values chosen for the join operators in this block.

$$< ts_C, X, A, 1/2 >, < ts_C, X, B, 1/2 >, < ts_C, X, C, 2 > .$$

4.5 Expectations

The expected amount of accusations (E) for a broken SM is estimated in the query by joining the tuples containing the readings that are being compared with the statistics. (J5). The window size and advance equal the window advance for the statistics aggregator. A stateless operator selects the maximum correlation value for every pair of SMs and outputs a tuple containing this value for each SM in the pair (S4). These tuples can then be aggregated in order to obtain the sum per SM, ie. the value of E (A5).

The expected number of accusations for the SMs in our example is given by:

$$< ts_C, X, A, 1.78 >, < ts_C, X, B, 1.80 >, < ts_C, X, C, 1.76 > .$$

4.6 Output

The final step in the query is then to combine the expected number of accusations with the total number of weighted accusations (J6) and output an alert if the number of weighted accusations exceeds the expected number weighted by θ'. (S5) The window size and advance for the join operator match the frequency of the readings.

θ' is set to 0.5 (majority voting) which renders the following final output in the example:

$$< ts_C, X, C >$$

5 EVALUATION

In this section, we present LoCoVolt's evaluation. We first introduce the evaluation criteria followed by the evaluation setup. Subsequently, we discuss how we simulated broken SMs based on real cases observed at the energy company. Finally, we evaluate LoCoVolt effectiveness in detecting such broken SMs and LoCoVolt's performance.

Evaluation criteria

In order to evaluate LoCoVolt's detection capabilities, the metrics we take into account are (1) the *detected percentage of manipulated readings*; (2) the *number of true positive (TP) and false positive (FP) alarms*; (3) the *detection time* for the different voltage manipulation rates. TP alarms are alarms that are generated for the manipulated SM after the manipulation has started, while FP alarms are alarms for SMs that have not been manipulated. We also evaluate(4) the *precision*, *recall* and *accuracy* which are common measures in classification problems [22]. These metrics are defined with TP and FP as well as true negative (TN) and false negative (FN) alarms. FN is the number of manipulated readings that did not trigger an alarm, while TN are the readings that neither were manipulated nor triggered an alarm. Precision is now defined as the quotient of TP by TP+FP, recall as the quotient of TP by TP+FN and accuracy is defined as the quotient of TP+TN by TP+TN+FP+FN.

In the case where there are multiple broken SMs simultaneously, we investigate (5) the percentage of detected broken SMs.

Finally (6) the performance in terms of *processing throughput* and *latency* is evaluated, in order to assess the possibility to run LoCoVolt on the hardware available in the AMI.

The detection time is especially important in order to minimize the duration and impact of the problems described in Section 3. The number of FP alarms should be as low as possible to minimize the manpower needed to investigate the alarms, while a high number of TP alarms generated by a large percentage detected readings helps to ascertain that a TP alarm truly is TP.

The results are compared with a baseline consisting of the current situation at the utility where SMs are investigated manually by system experts when the SM readings are outside of a predefined interval.

Evaluation setup

We evaluated LoCoVolt with data collected from 939 SMs during a period of 9 months. Each SM reports the voltage observed for each of its lines every hour, for a total of 4 million readings (about 1.5 million readings are missing). SMs connect to 26 Concentrator Units (CUs). The average number of SMs per CU is 36 while the CU with the largest set of SMs connects 152 and the CU with the smallest set connects 7 SMs. The SMs are connected to the grid by 30 different transformers. 26 CUs contain SMs that are connected to a single transformer. Two CUs contain SMs connected to two transformers while a single CU has SMs that are connected to three different transformers. Three kinds of SMs exist in the dataset, types 1 and 2 are three line SMs while type 3 is a single line SM. 366 SMs are of type 1 with a voltage resolution of 1 volt, 361 of type 2 with a resolution of 0.1 volts and finally 212 SMs of type 3 with a 0.1 volts resolution. The data is sanitized by removing values that would have been ignored based on existing validation rules.

We implemented LoCoVolt on top of Apache Flink [7] version 1.4.0. In order to test LoCoVolt's performance when potentially deployed at CUs, we run the performance evaluation experiments using a single-board device called Odroid-XU4 [14] (or simply Odroid in the remainder), equipped with a Samsung Exynos5422 Cortex-A15 2Ghz and Cortex-A7 Octa core CPUs and with 2 GB of memory. All other experiments were run on a standard off-the-shelf laptop computer.

Simulation of a broken smart meter

To the best of our knowledge, the SMs selected to conduct the evaluation are not broken during the period of time covered by the data. In order to check LoCoVolt's effectiveness in detecting broken SMs, we simulate the latter by manipulating their data, in ways that comply with the experts' description of experienced failures that are studied in a *post-mortem* fashion, i.e. after an SM is known to be broken through e.g. the damage caused by it. More concretely, we simulate two ways for which broken SMs have been observed in the AMI. In case I (*all-line*), we pick a random SM and, starting from a certain reading, we decrease its reported voltage every day with a constant rate. The rates we simulate are 0.005, 0.01, 0.015, 0.02 and 0.025. In case II (*single-line*), once a random SM is picked, we only alter one of its lines decreasing it every day with a constant rate. Also in this case, we simulate different decrease rates: 0.005, 0.01, 0.015, 0.02.

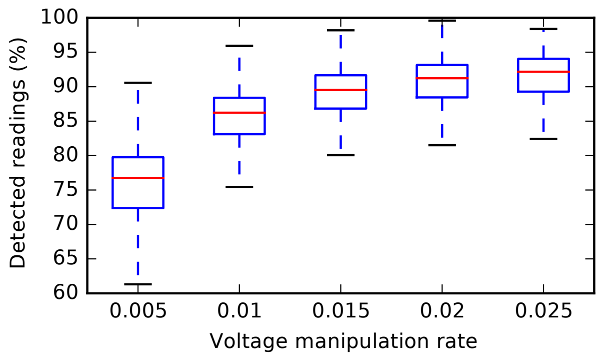

Figure 4: The detected percentage of manipulated readings for different voltage manipulation rates when all lines are manipulated. The largest outliers are typically between 20 and 30%.

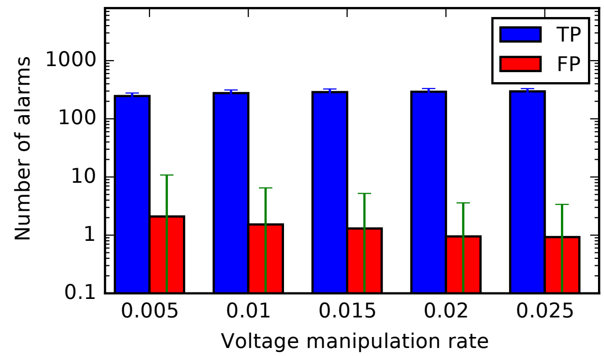

Figure 5: The mean and standard deviation of the number of TP and FP alarms respectively for different voltage manipulation rates when all lines are manipulated. The number of FP alarms is 2 orders of magnitude smaller than the number of TP alarms.

For both cases, we run 1000 experiments for each voltage manipulation rate, picking a random SM and a random starting date each time. The experiments start 28 days before the starting date of the manipulation, to ensure that the window for the statistics operator is filled, and ends after 14 days of manipulation.

Simulation of multiple bad SMs per CU

We also evaluate LoCoVolt when more than one SMs break simultaneously for two different cases. In case III (*multi-SM*) we simulate multiple SMs breaking independently from each other by selecting the correct number of SMs randomly. In case IV (*multi-correlated-SM*) we instead simulate the case where multiple SMs connected to the same transformer break, for instance due to lightning. For this case one SM is selected randomly while the other broken SMs are selected randomly from the set of SMs connected to the same transformer as - and physically close to - the first SM. We try different numbers of broken SMs: 2, 4, 6, 8 and 10 for both case III and IV.. 100 experiments are run for every number of broken SMs on a subset of the data described earlier. For these experiments we use the data from a single CU with 64 SMs that connect to two transformers (45 SMs to first transformer and 19 to the second one). For every experiment we decrease the readings for all lines on all selected SMs with a rate of 0.001 for every following day.

LoCoVolt detection capabilities

The detection capabilities of LoCoVolt are evaluated with the criteria described earlier in this section.

Figures 4, 5 and 6 present the results for the broken SMs in the all-line case. As presented in figure 4, the median percentage of detected manipulated readings grows from approximately 75% (voltage manipulation rate 0.005) to 90% (manipulation rate 0.025). This trend is reflected in Figure 5 showing the mean total number of TP and FP alarms that are detected as well as the standard deviation. This can also be observed in Figure 7, which shows that the accuracy in this case is very close to one independent of the manipulation rate. Both the recall and the precision increase for larger manipulation

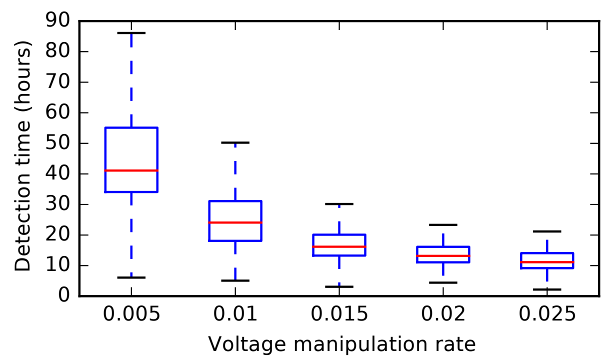

Figure 6: The detection time for different voltage manipulation rates when all lines are manipulated. The largest outliers can be found between 250 and 320 hours for all manipulation rates.

rates, consistent with the larger percentage of detected manipulated readings for larger voltage manipulation rates. The mean number of SMs reporting FP alarms is approximately 2.4 with a standard deviation of 5. The average number of FP alarms per SM reporting such alarms is around 3, but the actual number has quite some variation with a standard deviation of 15. The number of TP alarms is about 250 (voltage manipulation rate 0.005) and increases to 290 (manipulation rate 0.025). This shows that the number of TP alarms detected greatly outnumber the number of FP alarms for every manipulation rate. The detection time of the manipulation depends on the time it takes for the manipulation to become greater than the threshold θ. For a voltage manipulation rate of 0.005 and a normal voltage of 230, this will take approximately 62 hours. Figure 6 shows that the mean detection time at this manipulation rate is some hours faster, and similarly for the larger rates.

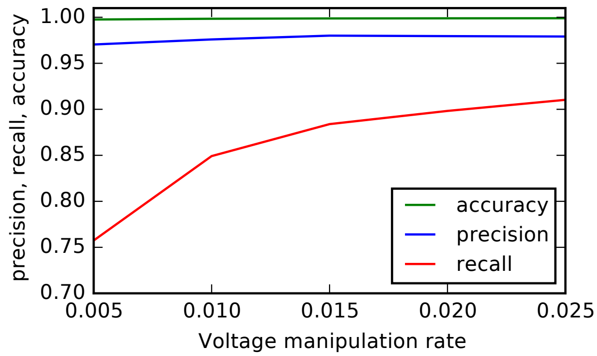

Figure 7: The precision, recall and average for different voltage manipulation rates when all lines are manipulated.

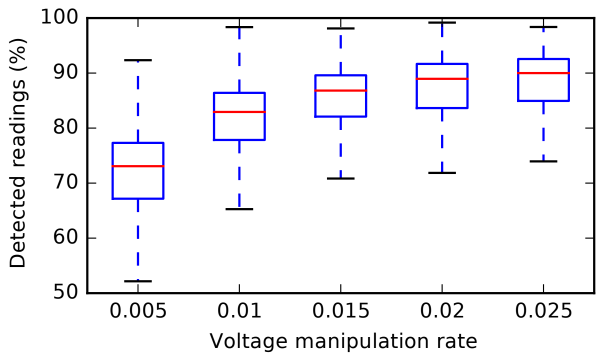

Figure 8: The detected percentage of manipulated readings for different voltage manipulation rates when a single line is manipulated. The largest outliers are typically between 20 and 30%.

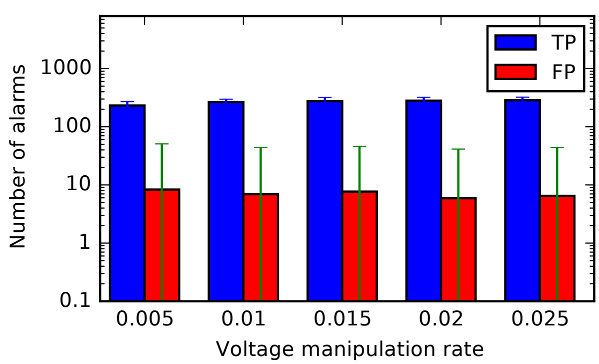

Figure 9: The mean and standard deviation for the number of TP and FP alarms respectively for different voltage manipulation rates when a single line is manipulated.

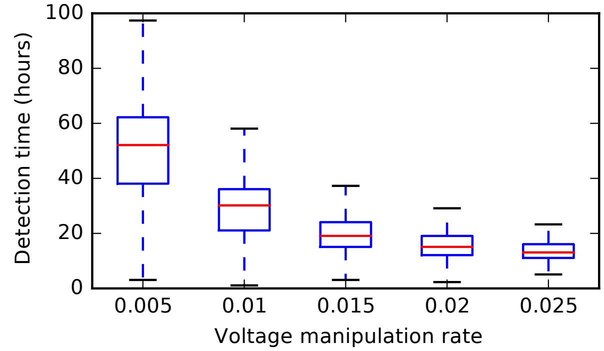

Figure 10: The detection time for different voltage manipulation rates when a single line is manipulated. The largest outliers can be found between 250 and 320 hours for all manipulation rates.

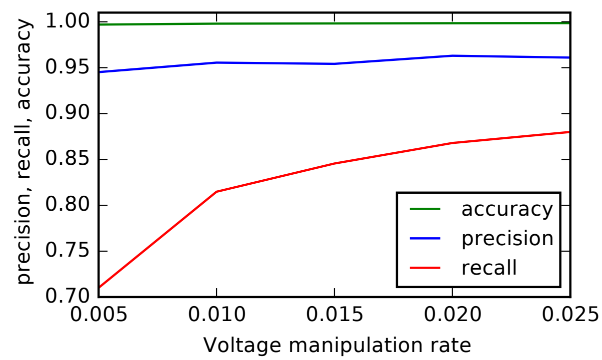

Figure 11: The precision, recall and average for different voltage manipulation rates when a single line is manipulated.

Figures 8, 9, 10 and 11 present the result for the single-line case. The figures show similar trends with the percentage of detected manipulated readings, as well as the number of alarms increasing with the voltage manipulation rate as can be seen in Figures 8 and 9. The absolute amount of alarms is however about 7% lower, with matching results for the percentage of detected readings. This is reflected by the detection time in Figure 10 which is increased by a similar amount. The amount of SMs reporting FP alarms is similar as in the all-line case, but a difference is that the amount of alarms per SM is increased to 7 on average and the standard deviation is increased to 40 as can be seen in Figure 9. The number of TP alarms still greatly outnumber the FP alarms. The recall and precision values follow this trend as can be observed in Figure 11. The accuracy however remains very close to one.

Alarms are triggered both by manipulated SMs as well as SMs that have not been manipulated. However the number of alarms triggered by manipulated SMs is at least one order of magnitude

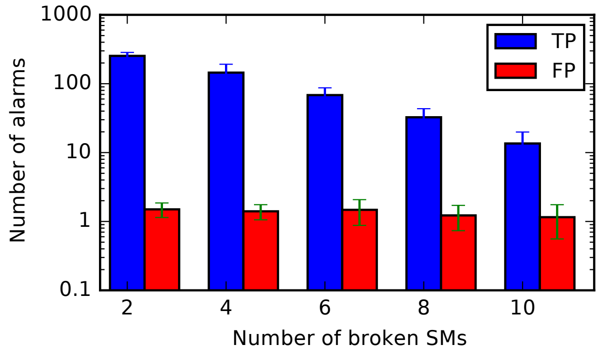

Figure 12: The mean and standard deviation for the number of TP alarms per manipulated SM and FP alarms per SM reporting such alarms respectively for different numbers of manipulated SMs.

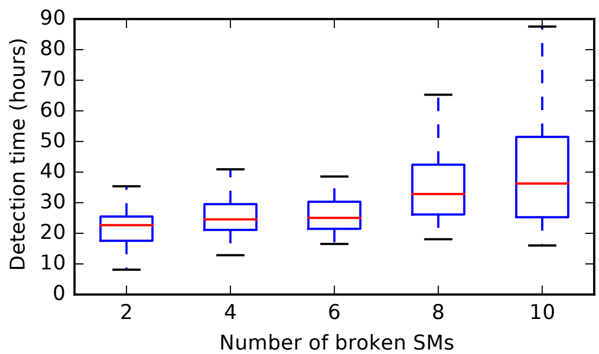

Figure 13: The detection time for the detected manipulated SMs for different numbers of manipulated SMs.

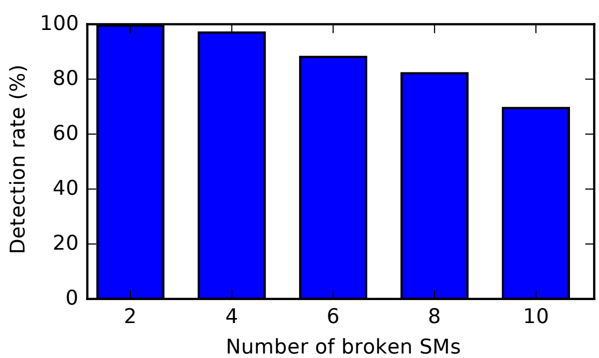

Figure 14: LoCoVolt's detection rate for different numbers of simultaneous manipulated SMs.

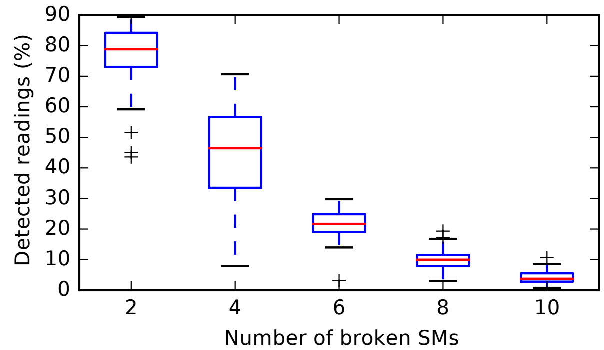

Figure 15: The detected percentage of manipulated readings for different numbers of simultaneous manipulated SMs.

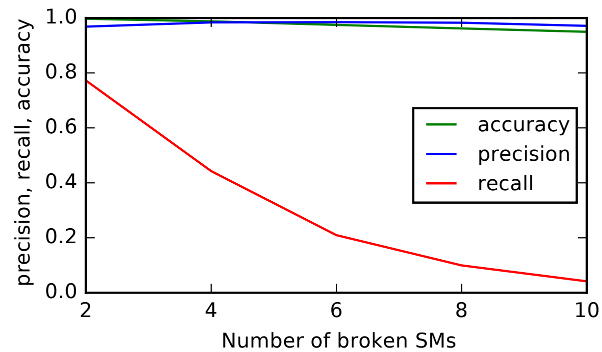

Figure 16: The precision, recall and average for different numbers of simultaneous manipulated SMs.

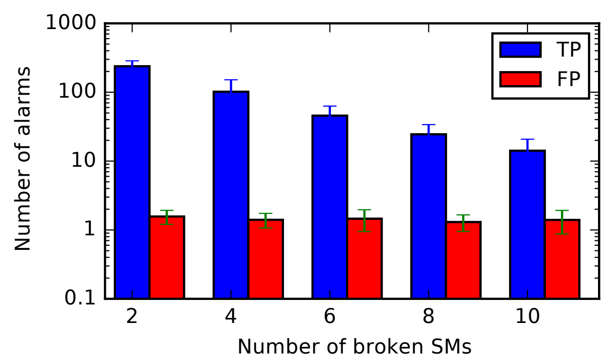

Figure 17: The mean and standard deviation for the number of TP alarms per manipulated SM and FP alarms per SM reporting such alarms respectively for different numbers of manipulated correlated SMs.

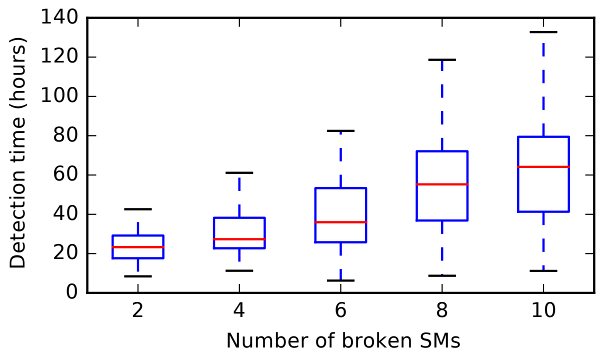

Figure 18: The detection time for the detected manipulated SMs for different numbers of manipulated correlated SMs.

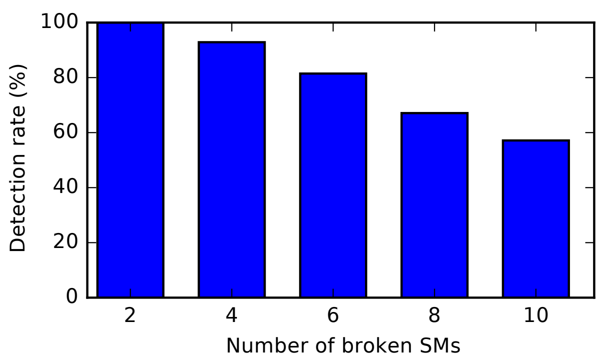

Figure 19: LoCoVolt's detection rate for different numbers of simultaneous manipulated correlated SMs.

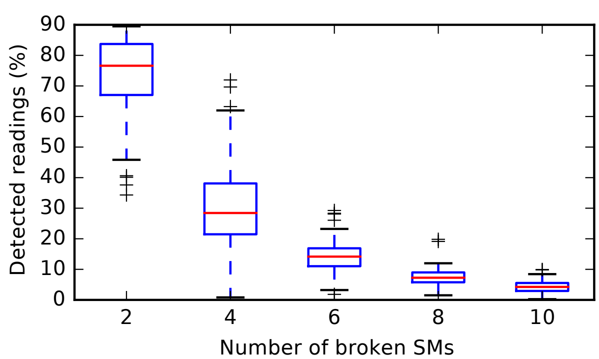

Figure 20: The detected percentage of manipulated readings for different numbers of simultaneous manipulated correlated SMs.

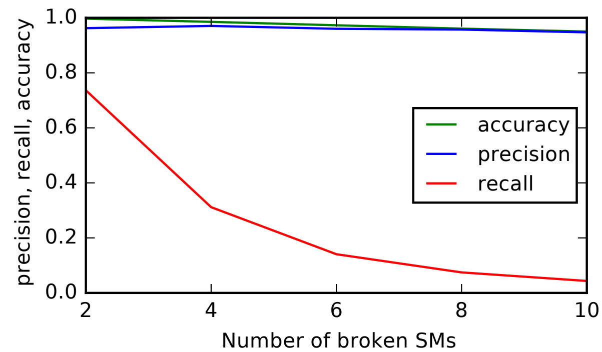

Figure 21: The precision, recall and average for different numbers of simultaneous manipulated correlated SMs.

larger than the number triggered by non manipulated SMs, as shown in Figures 5 and 9. This is also the case when multiple SMs are manipulated simultaneously, as shown in Figure 12. As currently done by system experts, all reported SMs need to be inspected. System experts usually start inspecting the SMs with the higher amount of alarms and stops when a false positive is encountered. If we define the detection rate as the percentage of broken meters that are indeed inspected, we can see in Figure 14 that the range of broken meters that are correctly detected goes from 100% (for 2 simultaneously broken meters) to approximately 65% for 10 simultaneously broken meters.. LoCoVolt's detection capability declines with increasing numbers of broken SMs with a rate for two bad SMs above 99% but then starts to decrease to reach 70% for ten bad SMs. Figure 12 shows that the number of TP alarms per SM decreases rapidly from a mean of 250 for two broken SMs to just 13 when the number of broken SMs is increased to 10. The average number of FP alarms however is independent of the number of bad SMs and smaller than two in all cases. The number of detected manipulated readings shows a similar trend as observed in Figure 15. The mean detection times for the broken SMs increases from 20 to 40 hours as can be seen in Figure 13. These trends are reflected in Figure 16 showing the precision, recall and accuracy. Both the precision and accuracy are close to one independent of the number of manipulated SMs, while the recall drops from 0.75 for two manipulated SMs to 0.05 when the number of manipulated SMs is increased 10. Even though the recall is only 0.05, the average number of alarms for a manipulated SM is still an order of magnitude larger than for a non-manipulated SM.

The results for the multi-correlated-SMs show similar trends. Figure 17 shows that the number alarms reported per TP or FP SM hardly changes, however the number of manipulated SMs that is detected declines faster as seen in Figure 19. This is also reflected in the number of manipulated readings that are identified which can be seen in Figure 20 as well as in Figure 21 which shows that the recall decreases faster in this case. The detection time increases faster when the broken SMs are highly correlated, as shown in Figure 18, with the mean detection time when for 10 broken SMs growing slightly over 60 hours.

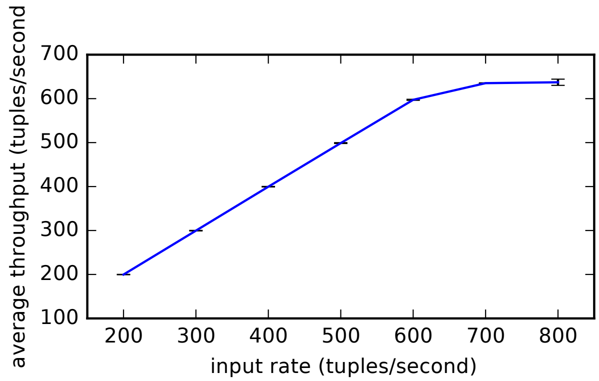

Figure 22: The mean and standard deviation of the throughput on the Odroid for different input rates.

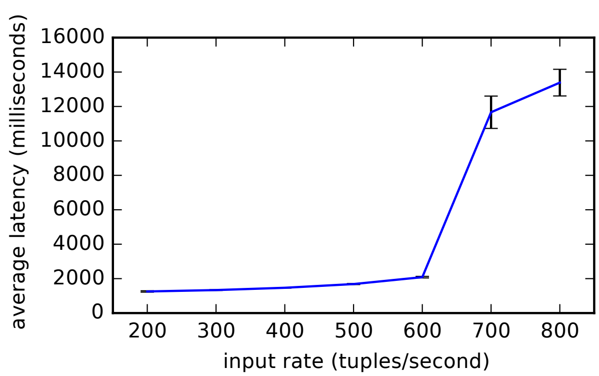

Figure 23: The mean and standard deviation of the latency on the Odroid for different input rates.

LoCoVolt performance

In order to study LoCoVolt performance, we also evaluate its throughput and latency when running on the Odroid. The throughput, measured in tuples/second, represents the rate with which an CU can process input tuples. The latency, measured in ms, represents the average time elapsed between the production of an output tuple and the receiving of the latest input tuple contributing to it. Note however that since LoCoVolt produces a reduced number of output tuples (only for alerts about broken SMs), the latency is measured at the input of operator S_5 in figure 3.

Figure 22 and 23 show the mean and standard deviation of the measured throughput and latency over three runs for every tested input rate. As shown, the Odroid can maintain a stable input rate of 600 tuples per second which is enough to sustain a reading interval of 1 second with a big margin. This throughput is more than sufficient for future applications, considering that the current reading interval is 1 hour. The latency is approximately 2 seconds at an input rate of 600 tuples/second, which is very small compared to LoCoVolt's detection time which is in the range of hours.

Evaluation summary

Broken SMs are currently usually detected when their readings exceed the allowed voltage range which in the case of this data set is $230V \pm 10\%$. A broken SM with a voltage manipulation rate of 0.005 would be detected after 20 days at the earliest. We show that the detection time using LoCoVolt is significantly smaller with a mean of 48 hours. The number of FP alarms generated by LoCoVolt is very small and greatly outnumbered by the number of TP alarms, enabling utilities to act on alarms swiftly since the possibility that the work required for an investigation is wasted due to a false alarm is very small. LoCoVolt can also detect situations in which there multiple broken SMs simultaneously, regardless of the correlation between these SMs. The detection rate decreases with increasing numbers of broken SMs, which is expected since a larger number of broken SMs will decrease the value of the weighted accusations. The detection rate is largest and the detection time smallest for the case where all broken SMs are picked randomly, but LoCoVolt outperforms the current detection method with a large margin even when the broken SMs are highly correlated.

We also show that it is possible to run the analysis on the next generation CUs. Carrying out processing in the deployed infrastructure would minimize the amount of data that needs to be uploaded to the utilities' central servers.

6 RELATED WORK

There is rising interest in the benefits from processing data in digitalized systems and especially so in the context of improving sustainable development in cities, where electricity networks is a key component in the infrastructure (c.f. e.g., [1, 12] and references therein). Especially in the latter it is highlighted that as data flows continuously in the systems, it is useful to process in a streaming fashion, before the data (or summaries of it [13]) is stored in big data-bases where it becomes infeasible to extract useful information in timely fashion. The latter is also one of the focal points of this paper.

The reliability of data and the robustness of the digitalized systems themselves are key issues. As examples, it is possible to mention that in [11] the authors have shown how to deal with data validation through continuous data-stream processing of the electricity consumption measurements so as to have trustworthy data for billing and for further processing to e.g. use in planning operations. In [6, 10] the authors study the problem of detecting potential intrusions in AMI, since these are highly motivated.

Besides protecting the robustness and the reliability of the infrastructures, it has been shown that general data processing and stream processing can generate valuable information, for e.g. detecting fraud, non-technical losses, power outages [3, 5, 8], thus protecting both safety/societal and economic aspects.

Our work combines stream processing of AMI voltage data with correlations of time series in order to detect broken SMs.

Correlation of consumption value time series has been used in a streaming fashion in order to clusters of similar customers [21].

Voltage data has seen increasing use in Smart Grids recently. E.g. voltage time series correlations have been used in order to verify the documented grid topology, either by correlating smart meter data with transformers [17, 18], or by correlating only smart meter data [16]. To the best of our knowledge, voltage data has never been used to identify broken smart meters nor have voltage correlations been done in a streaming and distributed fashion. Other uses of voltage data include power quality estimation in the grid [2].

Identification of bad individuals by peer to peer accusations has also been explored in wireless [15] and vehicular networks [19], where the notion of a group has a physical interpretation. Here we need to induce this information through the temporal dimension of the measurements of SMs, among the dynamic set reporting to the CU that locally processes the data.

7 CONCLUSIONS AND FUTURE WORK

The digitalization of electrical grids and in particular AMIs can provide the means to not only take and report measurements, but also to process the data in the deployed IT infrastructure and generate valuable information at the edge of the network, without relying on big cloud infrastructures and data centers. We strengthen this statement by addressing the problem of continuous distributed monitoring of voltage measurement streams, for detecting broken smart meters. Having this information is important for reliable billing, for prompt reaction for safety reasons, and, consequently, for the business value of the utility. We show that it is possible to have high accuracy and timely detection, even when the processing is done through resource-constrained devices such as the ones that are common in AMIs. The latter implies that this is achievable at a negligible cost for the utility.

Continuous stream-based monitoring can be beneficial for a series of other purposes, including facilitating planning operations, use of renewables and identifying other types of anomalies and unwanted situations.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation, by the Swedish Research Council (Vetenskapsrådet) proj. "HARE: Self-deploying and Adaptive Data Streaming Analytics in Fog Architectures" grant nr. 2016-03800, the Swedish Foundation for Strategic Research, project FiC, grant nr. GMT14-0032, by the EU Horizon 2020 Framework Programme under grant agreement 773717 and by the collaboration framework of Göteborg Energi and Chalmers Energy Area of Advance project STAMINA and project INDEED.

REFERENCES

[1] Damminda Alahakoon and Xinghuo Yu. 2016. Smart electricity meter data intelligence for future energy systems: A survey. *IEEE Transactions on Industrial Informatics* 12, 1 (2016), 425–436.

[2] Carlos Almeida and Nelson Kagan. 2011. Using genetic algorithms and fuzzy programming to monitor voltage sags and swells. *IEEE Intelligent Systems* 26, 2 (2011), 46–53.

[3] Carlos Barreto and Alvaro A Cárdenas. 2015. Detecting fraud in demand response programs. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on.* IEEE, Osaka, Japan, 5209–5214.

[4] John Bird. 2007. *Higher engineering mathematics.* Routledge Publishers, London.

[5] Viktor Botev, Magnus Almgren, Vincenzo Gulisano, Olaf Landsiedel, Marina Papatriantafilou, and Joris van Rooij. 2016. Detecting non-technical energy losses through structural periodic patterns in AMI data. In *Big Data (Big Data), 2016 IEEE International Conference on.* IEEE, Washington DC, USA, 3121–3130.

[6] Alvaro A Cárdenas, Robin Berthier, Rakesh B Bobba, Jun Ho Huh, Jorjeta G Jetcheva, David Grochocki, and William H Sanders. 2014. A framework for evaluating intrusion detection architectures in advanced metering infrastructures. *IEEE Transactions on Smart Grid* 5, 2 (2014), 906–915.

[7] Apache Foundation. 2014. Apache Flink. https://flink.apache.org/. (2014). last accessed: August 3, 2020.

[8] Zhang Fu, Magnus Almgren, Olaf Landsiedel, and Marina Papatriantafilou. 2014. Online temporal-spatial analysis for detection of critical events in cyber-physical systems. In *Big Data (Big Data), 2014 IEEE International Conference on.* IEEE, Washington DC, USA, 129–134.

[9] Vincenzo Gulisano. 2012. *StreamCloud: an elastic parallel-distributed stream processing engine.* Ph.D. Dissertation. Universidad Politécnica de Madrid.

[10] Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. 2014. Metis: a two-tier intrusion detection system for advanced metering infrastructures. In *International Conference on Security and Privacy in Communication Systems.* Springer, Beijing, China, 51–68.

[11] Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. 2014. Online and scalable data validation in advanced metering infrastructures. In *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES.* IEEE, Istanbul, Turkey, 1–6.

[12] Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. 2014. When smart cities meet big data. *Smart Cities* July 2014, 98 (2014), 40.

[13] Vincenzo Gulisano, Valentin Tudor, Magnus Almgren, and Marina Papatriantafilou. 2016. Bes: Differentially private and distributed event aggregation in advanced metering infrastructures. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security.* ACM, Xi'an, China, 59–69.

[14] Ltd Hardkernel co. 2015. Hardkernel ODROID XU4. http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825. (2015). last accessed: August 3, 2020.

[15] Sung Yul Lim and Yoon-Hwa Choi. 2013. Malicious node detection using a dual threshold in wireless sensor networks. *Journal of Sensor and Actuator Networks* 2, 1 (2013), 70–84.

[16] Wenpeng Luan, Joshua Peng, Mirjana Maras, Joyce Lo, and Brian Harapnuk. 2015. Smart meter data analytics for distribution network connectivity verification. *IEEE Transactions on Smart Grid* 6, 4 (2015), 1964–1971.

[17] Rajendu Mitra, Ramachandra Kota, Sambaran Bandyopadhyay, Vijay Arya, Brian Sullivan, Richard Mueller, Heather Storey, and Gerard Labut. 2015. Voltage correlations in smart meter data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, Sydney, Australia, 1999–2008.

[18] Houman Pezeshki and Peter Wolfs. 2012. Correlation based method for phase identification in a three phase LV distribution network. In *Power Engineering Conference (AUPEC), 2012 22nd Australasian Universities.* IEEE, Bali, Indonesia, 1–7.

[19] Maxim Raya, Panagiotis Papadimitratos, Imad Aad, Daniel Jungels, and Jean-Pierre Hubaux. 2007. Eviction of misbehaving and faulty nodes in vehicular networks. *IEEE Journal on Selected Areas in Communications* 25, 8 (2007), 1557–1568.

[20] Navigant Research. 2012. Smart Grid Networking and Communications. (2012).

[21] Pedro Pereira Rodrigues and João Gama. 2009. A system for analysis and prediction of electricity-load streams. *Intelligent Data Analysis* 13, 3 (2009), 477–496.

[22] Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45, 4 (2009), 427–437.