Athens Journal of Sciences 2025, 12: 1-15
https://doi.org/10.30958/ajs.X-Y-Z

Event-based Energy Modelling

By Manuel Lehner[*]

Information technology is a major contributor to global energy consumption. This study estimates event-level energy usage through linear regression applied to synthetic signals generated from process event logs. These signals incorporate sine-modulated baselines, randomized noise, and event-specific energy costs. A custom framework supports signal generation, cost attribution, and performance evaluation. The model demonstrates high estimation accuracy for frequent and independent events, though performance declines with tightly sequenced process chains. R-squared values suggest a strong overall model fit, while percentage deviations vary depending on signal characteristics. The method assumes linearity, independence, and homoscedasticity—assumptions are largely satisfied in the synthetic dataset. Real-world applications, however, are limited by potential multicollinearity and require validation using actual event logs and measured energy data. Modular processes with stable event costs offer a promising testbed. Potential applications include process-level energy attribution, optimization, and client-specific billing. Although not yet deployable, this research lays the foundation for event-based energy modelling in complex IT environments.

Keywords: *process mining, energy efficiency, IT sustainability, data visualization, supply chain*

Introduction

As digital transformation accelerates across all sectors, the energy consumption of information technology (IT) systems has become a pressing environmental and economic concern. (Gelenbe 2023) Data centers, blockchain networks, and distributed computing infrastructures now account for a substantial share of global electricity usage. While aggregate energy consumption statistics are well documented, fine-grained insights into how individual computational events contribute to energy use remain limited. This knowledge gap hinders efforts to optimize energy efficiency at the process or service level and complicates attempts to assign energy costs to specific users or operations.

Recent advances in process mining and telemetry data collection have made it feasible to track detailed event logs of digital systems. However, directly linking these logs to energy consumption remains a challenge due to the noisy, high-dimensional, and often opaque nature of IT infrastructure. Traditional energy profiling methods either rely on coarse system-level measurements or require specialized hardware instrumentation, making them difficult to scale or integrate into live systems.

To address these limitations, this study explores an event-based modeling approach to estimate energy consumption using statistical inference from synthetic

[*]Scientific Staff, Chair on distributed Information Systems, University of Passau, Germany.

data. By combining process event logs with simulated energy signals that include deterministic (event-related) and stochastic (noise) components, the methodology aims to infer per-event energy costs using linear regression. This offers a scalable and potentially real-time mechanism to estimate energy footprints at a granularity compatible with process analysis and optimization.

Such a framework has promising applications: from attributing energy usage to specific business processes or users, to identifying inefficiencies in software workflows, and even enabling billing models based on actual resource consumption. However, the validity of this approach hinges on several key assumptions—including linearity, independence, and low multicollinearity—which may not hold in complex real-world systems. Therefore, this research provides a proof-of-concept using synthetic data and outlines the challenges and opportunities for applying the method to actual operational environments.

Literature Review

A look into the field of process mining reveals some trends with the view of adding sustainability to the field. Ma et al. (2020) proposes data driven optimizations on specific energy demanding industries with an elaborate collection of cloud-based monitoring, enabled by extensive hardware monitoring using RFID tags and smart sensors. While being a reasonable approach for the proposed industries, an extension towards regular businesses is unlikely due to lack of knowledge, personnel, funding or all the above. However, Delgado et al. (2023) shows a methodology to encompass sustainability data into process mining discovery, which are also securely shared using blockchain technology. While addressing the security needs for an enforceable mechanism sharing and trusting sustainability data, the way to retrieve the data to feed into such a reporting system is still not addressed. The new research of Fritsch (2024) gives a link of challenges between life cycle analysis (LCA) and Business Process Modelling (BPM) research projects. While creating a mapping between the fields, the methodologies of LCAs are notoriously difficult to adapt into real life applications, which can be seen by industries usually adapting statistical averages for their consumption data on non-financial reporting, instead of accurately measuring their values. This is done to a decrement as indications show an improvement in reporting when using actual measurements instead of statistical estimations (Tsalis et al. 2017).

Methodology

Energy Event Evaluation

The mathematical evaluation of the dataset is done using a multiple linear regression combined with an ordinary least squares estimation to distinguish event costs from background noise such as uncorrelated processes on a particular machine or manufacturing site. The basic evaluation equation and estimation formular can be seen in equation 1. The following equation 2 describes the estimation method used to arrive at the values. The methodology relies on the assumption that every event relevant to the specific enterprises is logged in the event log.

Equation 1: Basic formulation of the linear equation where: E_{total} is the total energy consumption observed during the measurement process (dependent variable), $E_1 \ldots E_n$ the energy consumption associated with individual events recorded (independent variables or predictors). β_0 the intercept term, representing baseline energy consumption that is independent of the specific events. $\beta_1 \ldots \beta_n$ the regression coefficients, indicating the magnitude of contribution each event makes to the total energy consumption.

$$E_{total} = \beta_0 + \sum_{i=1}^{n} \beta_i \, iE_i$$

Equation 2: the Ordinary Least Squares (OLS) method is employed. This approach minimizes the sum of squared residuals for each interval I, when looking at all the intervals in a matrix X, the second equation has to be solved:

$$S = \sum_{i=1}^{n} (E_{total,i} - (\beta_0 + \beta_1 E_{1,i} + \beta_2 E_{2,i} + \beta_3 E_{3,i}))2$$

$$(X^T X)\beta = X^T Y$$

To be able to distinguish between noise signal and actual consumption values from each event the estimation needs two inputs, an **event matrix** and an **energy signal**. The event matrix is provided that was derived from the dataset using the timestamp. In code this was accomplished by parsing the dataset file and counting each occurrence of an event type during one interval. Listing 1. gives a small example of such an event matrix to a fictitious event log.

Listing 1. *Example of an Event Matrix to a Fictitious Event Log*

Given an event log with mm:ss timestamps and the event types:
***read_data**, **process_data** and **write_data**:*
00:01 read_data, 10:15 process_data,
12:13 read_data, 17:12 process_data,
18:00 write_data
Given the interval length of 15 minutes the resulting matrix is: $\begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$

The second input is ideally a combined real time energy measurement of all entities that consume energy to achieve the process goal. As such measurements contain a variety of unrelated consumption values, such as overhead processing power from computers, consumption data from other processes and many more, the methodology must be robust against noisy consumption signals. As the dataset provided does not contain such an energy measurement, a methodology to generate such an energy measurement was derived using simple assumptions on the way energy is consumed by each of those processes. These consumption values rely on a huge number of factors, not limited to hardware used, virtualizations or even the programming language of an application (van Kempen et al. 2025). Due to this limitation, it is hard to justify specific values for a given process, if the implementation of the processes is unknown. However, the goal of this study is to evaluate the general applicability of this method to a variety of different costs and implementations. To accomplish this an energy data generation framework was developed that can create energy consumption signals with optional variations to process costs, amount of noise or fluctuations during regular daytime hours. When looking at the results in chapter 4 the usage of concrete values was minimized, as they in themselves do not have any significance. Rather it is to show that for any known amount of energy that a given process uses, can this estimation methodology be used to reconstruct the costs of different events when only given an event matrix and the sum of all the measurements per time interval. The following subsection gives some insights into the dataset, as well as the data processing done for this research.

Dataset and Datastructures

BPI_Challenge_2019 is a well-known published dataset in process mining (van Dongen 2019). The event data is a collection of economic processes across the whole year 2018, containing 41 different event types with their corresponding timestamps and standardized process meta data. As the dataset contains outlier events out of the scope of the year 2018, this experiment used a filtered set, that encompasses all events attributable to the year 2018. No further alterations were made, and the filtering was done using pm4py functionalities found in the xes_filter.py package. For the energy generation that corresponds to the event log, an EnergyEvent data structure was designed to add details to the process events that are not part of the original data set. To get a better understanding of the results, the frequency of each event type can be found in table 1. To ease the referral to specific groups of events, the different event types are grouped into classes.

EnergyEvent is a data structure that provides additional information for the energy signal generation. As the methodology works on fixed measurement intervals, the duration of specific events may have an influence on the energy allocation of the event across different intervals. To simulate this, the duration of the event was randomized, and the energy costs were distributed across different sections of the time series. As this must be expected in a real-world environment, but no duration data is present in the dataset, a random quantity will account for some of the distortions to be expected when using real measurements. An overview of the data structure can be found in figure 1.

The results of the conversion from the xes file can be saved using the pickle function of python. This way the binaries can be saved for further analysis without the need for repeated conversions. The framework is adapted to use standardized xes nomenclature and should be able to work with other process mining datasets with little adjustments. This enables the reader and future research to reuse the framework with updated event logs or more accurate basecost assumptions when applicable.

Table 1. *Frequencies and Group of each Event Type in the BPI challenge Dataset*

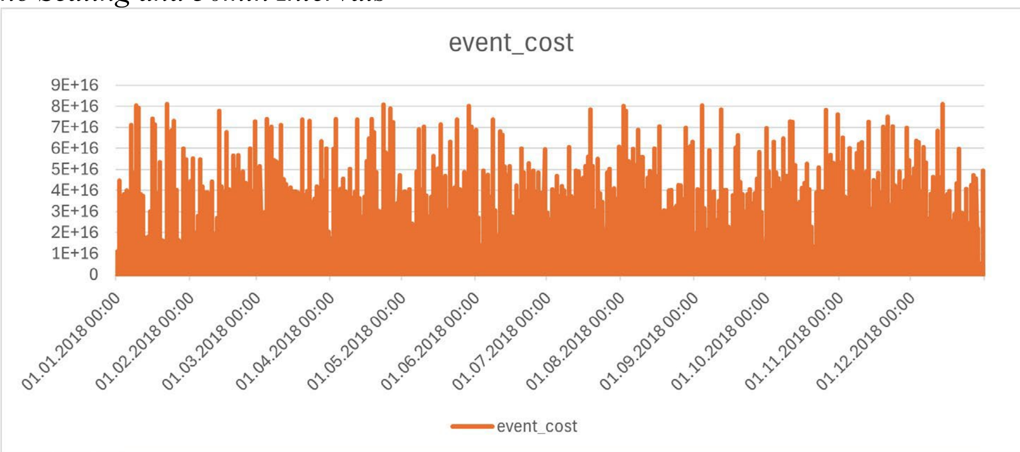
Event type	# Occurences	Grouping
'SRM: Created'	1625	Medium
'SRM: Complete'	1625	Medium
'SRM: Awaiting Approval'	1625	Medium
'SRM: Document Completed'	1625	Medium
'SRM: In Transfer to Execution Syst.'	1687	Medium
'SRM: Ordered'	1625	Medium
'SRM: Change was Transmitted'	1440	Medium
'Create Purchase Order Item'	251352	High
'Vendor creates invoice'	216700	High
'Record Goods Receipt'	305837	High
'Record Invoice Receipt'	218375	High
'Clear Invoice'	176755	High
'Record Service Entry Sheet'	164657	High
'Cancel Goods Receipt'	3047	Medium
'Vendor creates debit memo'	6213	Medium
'Cancel Invoice Receipt'	6768	Medium
'Remove Payment Block'	54216	High
'SRM: Deleted'	185	Low
'Change Price'	12189	High
'Delete Purchase Order Item'	8698	Medium
'SRM: Transaction Completed'	8	Low
'Change Quantity'	20924	High
'Change Delivery Indicator'	3172	Medium
'Change Final Invoice Indicator'	11	Low
'SRM: Incomplete'	6	Low
'SRM: Held'	6	Low
'Receive Order Confirmation'	32027	High
'Cancel Subsequent Invoice'	463	Low
'Reactivate Purchase Order Item'	535	Low
'Update Order Confirmation'	242	Low
'Block Purchase Order Item'	486	Low
'Change Approval for Purchase Order'	6961	Medium
'Release Purchase Order'	1610	Medium
'Record Subsequent Invoice'	151	Low
'Set Payment Block'	123	Low
'Create Purchase Requisition Item'	46585	High
'Change Storage Location'	40	Low
'Change Currency'	35	Low
'Change payment term'	7	Low
'Change Rejection Indicator'	2	Low
'Release Purchase Requisition'	467	Low

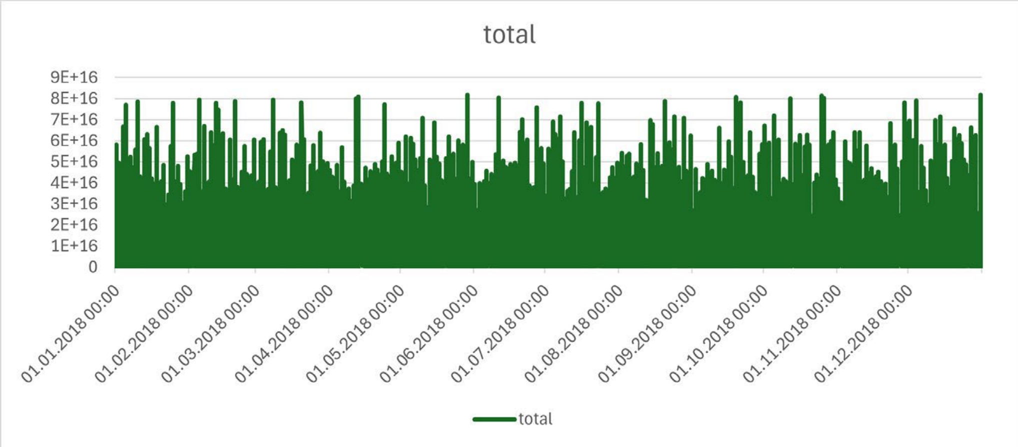
Figure 1. *EnergyEvent Datastructure*

Signal Generation

The energy signals that should simulate a correlating energy signature to the given dataset is created using the EventObject data structure. The code can be found in the jupyter notebook in the repository (Lehner 2025). The idea is to inject already known event costs to a signal comprised of a noise signal that is modulated by a sinus wave in order to simulate empirically shown day and night energy consumption differences (Gupta 2022). As the energy consumption of a commercial endeavor can vary drastically between industries, implementations and products, a representative value for a generic application cannot be chosen confidently. To account for this, the research introduces variations in the signal generation to encompass the influence of the order of magnitude in different factors and are described in detail in the next section. A sample energy signal of the total and event costs can be seen in figure 2.

Figure 2. *Comparison of the Event Costs and Total Energy Costs, Experiment with no Scaling and 30min Intervals*





Signal Variations

To give a broad insight into the applicability of the method, the signals were generated with a wide range of parameter variations. Specifically, the noise added to the signal, the altitude of the sine wave that modulates the signal, as well as the event base costs that are given by design are scaled by two orders of magnitude. As the performance of the estimations should be reduced by real world distortions as well as the varying amounts of noise within a system, this spectrum should be a sufficient insight into the resilience of the method.

Implementation

event_object.py

This module defines the core data structure and processing logic for event instances derived from XES logs. The EventObject class encapsulates key attributes such as event identifiers, case associations, timestamps, and event-specific durations. It supports cost evaluation based on static base costs (__BASECOSTS__) and dynamic temporal penalties incorporating duration and delay penalties (__DURATION_SCALE__, __END_SCALE__). Additionally, it provides methods to determine subsequent events within the same case and compute total event-related costs.

The module includes:

1. A pipeline (process_xes_events) that parses XES event logs using PM4Py, transforms them into EventObject instances, and applies user-defined processing functions.
2. Utility for loading serialized event data via read_event_objects_from_pickle.

event_cost_eval.py

This module provides a statistical modeling framework for estimating and validating the energy cost associated with event types based on empirical usage data. Key functionalities include:

1. **estimate_energy_per_event**: Applies multiple linear regression to estimate energy consumption per event type using event frequency data across time intervals, with optional intercept control.
2. **analyze_energy_regression**: Extends the basic estimation by leveraging statsmodels to extract confidence intervals and p-values, enabling significance analysis of each event type's contribution.
3. **compare_estimated_vs_expected_costs**: Computes and reports absolute and relative deviations between regression-estimated costs and known base costs, supporting model validation.
4. **visualize_cost_comparison**: Generates bar plots comparing estimated and expected costs per event type, with annotated percent deviations for interpretability.
5. **estimate_and_compare_costs**: An integrated workflow to estimate, compare, and visualize energy costs from interval-based event and consumption data.
6. **create_event_matrix**: Constructs a matrix representation of event occurrences per time interval from an input XES log, aligning with energy usage timestamps to support regression analysis.

The module emphasizes both analytical rigor and practical usability, facilitating transparent and quantitative cost validation of process events.

The core of the signal generation was implemented in a jupyter notebook.The codebase implements a synthetic energy consumption data generator for business process management scenarios. The system creates realistic energy consumption patterns by combining periodic signals, noise, and process-specific energy costs derived from event logs. The core components are listed as follows:

time series foundation

The system begins by establishing a time series framework using pandas DateTimeIndex, creating regular intervals (15-minute default) for energy measurements throughout the specified period (2018).

periodic signal generation

The generate_sine_wave function creates a fundamental periodic pattern in energy consumption based on:

1. Daily cycles with configurable peak hours (default: 2 PM)
2. Amplitude variations (configurable)
3. Vertical shifts to establish baseline consumption

This models daily operational patterns in business environments where energy consumption follows predictable cycles.

noise integration

Random noise is applied via the noise_gen and apply_noise functions, simulating energy consumption unrelated to business processes (OS operations, background services, etc.). This adds realism by accounting for the variable, non-deterministic component of energy usage.

process-specific energy costs

The distribute_event_costs_series function:

1. Maps specific business process events to energy costs
2. Distributes these costs across time intervals based on event duration
3. Handles overlapping events and calculates proportional energy attribution

The system uses a detailed cost mapping dictionary (__BASECOSTS__) that assigns different energy weights to various business processes (e.g., "Record Goods Receipt": 99, "Create Purchase Order Item": 98).

experimental framework

The implementation includes a systematic approach to parameter exploration:

1. Noise scaling (0.1-10)
2. Base cost scaling (0.1-10)
3. Sine wave amplitude (1-1.5)
4. Sampling rate (30min)

The gen_experiment_parameters function generates combinations of these parameters, with each experiment fixing parameters at median values while exploring variations of a single parameter.

data transformation utilities

Post-generation utilities include:

1. convert_30min_to_1h_reset_index: Aggregates higher frequency data to lower resolution while preserving energy relationships
2. resize_sine_wave: Adjusts the amplitude of the periodic component without regenerating the entire dataset
3. for the analysis of this paper, not all utilities were used in order to not exceed content limits

methodology

The energy generation process combines these components sequentially:

1. Generate timestamp range at specified intervals
2. Apply periodic sine wave pattern with configurable parameters
3. Introducing random noise to simulate non-process energy consumption
4. Distribute process-specific energy costs from event logs
5. Calculate total energy consumption as the sum of these components

Each generated dataset is saved as a CSV file with naming conventions that encode the experimental parameters, facilitating subsequent analysis of how different factors influence energy patterns. By cloning the repository, all results can be calculated using the code provided and a glimpse into the main results are given in the next section.

Results

While giving some insights into the behavior of the methodology, it must be restated that the absolute values are of little interest and only give indication on whether the data collection and evaluation is meaningful given a concrete scenario. The synthetic energy signals are available in the repository as .csv files and contain separate columns for the noise part of the signal and the energy values infused by the event costs. Consider that when simply adding the base costs of the events within a timestamp interval will most likely not match the actual value of the signal, as the event costs have been allocated appropriately based on the start time of the event and the randomized duration in a linear fashion. (i.e. if the duration bleeds by half of the duration into the next interval also half of the costs are added to the next interval).

To evaluate if the methodology can be used in a specific application, this work provides a framework to get first insights, before any more elaborate implementations must be made. This chapter will provide a look into the general model performance, the well performing aspects and the problem areas.

General Model Performance

The performance of the linear regression model was evaluated using the analyze_energy_regression function of the code. Table 2 gives an overview of the r-squared, and the mean-percentage difference for all events of the dataset. As a main indicator of the overall performance the r-squared value gives good insight into the explanation-power of the model to predict overall energy. The values for the percentage difference can give insights into the performance when attributing costs to individual events, which is heavily influenced by the process structure.

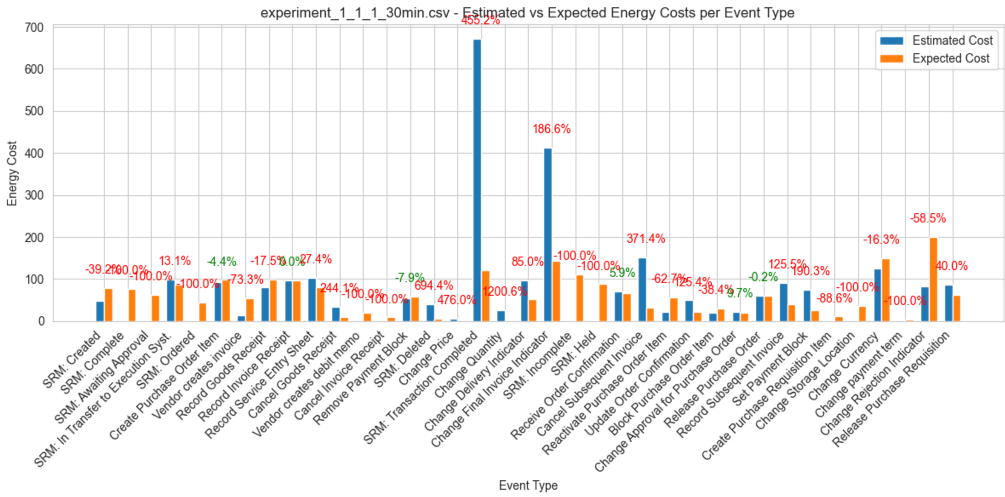
Table 2. *Statistical Evaluation of Model Performance in each Experiment*

Scaling Parameters			R-squared	Mean Absolute % Difference
Cost	Noise	Amplitude		
0.1	0.1	1	0.7919	150.93
0.1	1	1	0.7914	146.74
0.1	10	1	0.7913	145.2
1	0,1	1	0.7948	298.5
1	1	1	0.7919	150.20
1	10	1	0.7913	146.67
10	0.1	1	0.6716	2770.89
10	1	1	0.7948	287.99
10	10	1	0.7919	150.59

1	0,1	1.5	0.7937	84.99
1	1	1.5	0.7916	148.24
1	10	1.5	0.7913	1738.64
10	0.1	1.5	0.7639	99.65
10	1	1.5	0.7937	85.23
10	10	1.5	0.7916	147.30

As a blank modelling approach this does not work that well, since process chains and dependencies cannot be accounted for in a least squares estimation model. Figure 3 shows the resulting estimations in comparison to the real costs of the experiment with the cost-/ noise-/ and amplitude parameters each set to 1. As visible from the illustration, the estimations vary widely for the different event types. With the frequency of the types in mind it shows that higher frequency events are estimated more accurately than low frequency events. This is likely due to the higher chance of separating clear signal data from varying noise during each of the occurrences. However, the estimations show even better results for event types that better fulfill the assumptions, which are discussed in detail in the next section. This is mainly independence due to being first in a process chain, and not closely followed by events afterwards.

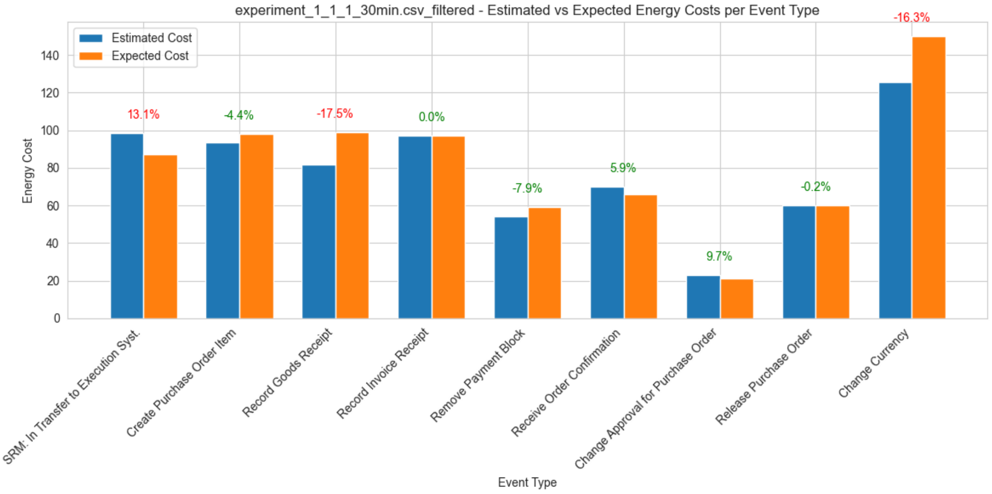
Figure 3. *Estimation Result of Experiment Setting (1,1,1) and a 30-minute Interval*



Zoom on better Performing Events

When looking at the specifically well performing event types the picture changes quickly and gives some details on how to improve the methodology. In figure 4 the results of the best performing events of the same parameters can be seen. With comparison to table 2 most of the consistently well-performing events are of high frequency. When comparing the different experiment runs, low frequency event types are estimated quite accurately, however this is likely due to random variance.

Figure 4. *Estimation Results of Experiment Setting (1,1,1) and a 30-minute Interval. Best Performing Estimations only*



Problem Areas of Performance

The main areas of performance drop occur when looking at process chains that happen in some rapid succession. In the case of most of the SRM events, this is especially obvious as the cost of the whole process chain was summed up into the last event of the process chain SRM: transaction completed. This explains most of the huge spike of the costs in figure 2. However, as the model also does not account for all the process-induced energy costs from the signal, additional errors in that spike must be expected. To alleviate this discrepancy between the signal and the estimations, the correlations between different events must be considered. Different ways to address this are discussed in the next section.

Discussion

The evaluation methodology makes basic assumptions on the nature of how energy consumption behaves with respect to events in the log. The model relies on several assumptions regarding the system and its energy consumption behavior:

1. **Linearity**: The relationship between the individual event-related energy costs and the total energy cost is assumed to be approximately linear.
2. **No Multicollinearity**: The predictor variables (i.e., the energy costs of individual blockchain events) should not be highly correlated with each other. For instance, if event types such as validation and storage are strongly interdependent, this may lead to multicollinearity issues.
3. **Homoscedasticity**: The variance of the error terms should remain constant across all levels of the predictors.

4. **Normal Distribution of Errors**: The residuals, defined as the differences between observed and predicted total energy costs, should be normally distributed.

Most of the assumptions are present in the synthetic energy data per design, which probably gives an overestimation of the effectiveness of the method. This is possibly counterbalanced by the multiple orders of magnitude represented in the dataset, as a difference of 10^4 in energy consumption will mean that a meaningful increase in energy efficiency of the event costs is not impacting the total energy consumption of the operation. The decision, whether the proposed methodology is applicable to a potential process, must take a lot of previous knowledge into account, so that the above-mentioned assumptions hold to a meaningful extent. Further process mining techniques can be used to single out process variants or process paths that more clearly fit the assumption of no multicollinearity is met. Circular or strongly correlated process structures can be suspected to have some correlation in their energy signature and therefore parts of the costs will be underestimated by the bias coefficient. A way to address this drawback would be a restrictive modeling approach towards only independent variables and switching to other methodologies when looking at highly dependent event variables.

To estimate the real amount of multicollinearity within a real process, future work must be completed providing both datapoints 1, an extensive event log and the corresponding energy consumption measurements. To provide a baseline of known event costs, the process must be modular in such a way that events can be process-independent of previous steps and the amount of energy needed should be independent of previous steps. These constant energy consumption measurements then can be used as a ground truth to evaluate the methodology in an application. Other than a simplifying approach, a statistical learning algorithm or feature extraction can be used to learn cost features from real time data. Depending on the feasibility of each of the next research projects, either method can lead to great improvements. Further economic computation trends, like off-site computation and cloud computing, will make data collection even more difficult. However, hardware to monitor and record energy consumption become affordable and often also included in regular commercial products. As a potential upside, providers of computing services that would like to separate energy costs to distinct clients could benefit from the application of an event-based energy model. This way a billable energy footprint could be attributed to a single process owned by the client. In return this consumption rate can then be incorporated into the energy balance calculation of the economic chain down the line. If this is a feasible way of accounting for energy costs must be shown with real data.

The data set used in this experiment maps all economic events of the enterprise, however it does not map to a realistic event collection when looking into computation costs. To analyze and possibly improve energy efficiency, the design and decisions which datapoints are recorded should be made using the biggest contributors to energy consumption in IT. The procurement of new IT products and

[1]Combined energy measurement of all contributing process devices, as well as the process event data.

the usage. Since a general determination which side of the consumption fields dominate (Hilty et al. 2007) a closer look at improving energy efficiency within processes is.

Conclusions

In conclusion this research is a step to provide reasonable indicators that with some modifications an event-based energy modelling system can be implemented in order to give insight into:

1. Process specific energy costs
2. Possibilities to model modifications in the process
3. Partly generate energy footprints for different process end products
4. Further usage of already collected energy data

However, a blank usage of this methodology is unfit to provide reliable values for the purpose of determining event costs. This can possibly be achieved when providing improvements to the methodology that ensures either a stricter adherence to the assumptions of linear regression models for each and every event type under consideration, or correlation information using process mining techniques to augment the existing data sources.

While at this point it cannot be estimated if a calculation can compare to real measurements, but similar approaches on stock exchange data had R^2 scores of about 0.96 (Majumder et al. 2022) which gives some reason to be optimistic. With the inclusion of real measurements in the energy of some processes, also other methods open up. These include, but are not limited to, learning algorithms and feature extractions to create a reliable modelling framework.

Acknowledgments

Many thanks for continuous research guidance are owed to Prof. Dr. Dr. Harald Kosch of the University of Passau, Prof. Dr. Rainer Groß of the Technische Universität Nürnberg, Dr. Thomas Götz of Atos/Eviden and Ms. Alexandra Knupe of Atos.

References

Delgado A, et al. (2023) Adding the sustainability dimension in process mining discovery algorithms evaluation. *Lecture Notes in Business Information Processing* 490: Business Process Management Forum.

Fritsch A (2024) Sustainability analysis patterns for process mining and process modelling approaches. *Lecture Notes in Business Information Processing* 533: ICPM. Springer.

Gelenbe E (2023) *Electricity Consumption by ICT: Facts, Trends, and Measurements.* Ubiquity, Article 1, August.

Gupta R, Antony A, Walia A, Dhingra N, Tathagat T, Varma P (2022) Examining the variation in daily and peak electricity use. University of Bath.

Hilty LM, Coroama V, Ossés de Eicker M (2007) The role of ICT in energy consumption. *ICT-ENSURE: European ICT Environmental Report*, 30.

Lehner M (2025) Event-based energy modelling – Repository for the BPI Challenge research. GitHub. https://github.com/Free2ndThought/event_based_energy_modelling_bpi2018.git (accessed 30 June 2025).

Ma S, et al. (2020) Data-driven sustainable intelligent manufacturing based on demand response for energy-intensive industries. *Journal of Cleaner Production* 274.

Majumder A, Rahman MM, Biswas AA, Zulfiker MS, Basak S (2022) Stock market prediction: A time series analysis. In: *Smart Systems: Innovations in Computing* (Smart Innovation, Systems and Technologies).

Tsalis T, Avramidou A, Nikolaou IE (2017) A social LCA framework to assess the corporate social profile of companies: Insights from a case study. *Journal of Cleaner Production* 164: 1665–1676. Elsevier.

van Kempen N, Kwon HJ, Nguyen DT, Berger ED (2024) It's not easy being green: On the energy efficiency of programming languages. *arXiv* (October 2024). https://arxiv.org/html/2410.05460v1 (accessed 30 June 2025).

van Dongen BF (2019) Dataset BPI Challenge 2019. *4TU.Centre for Research Data.* http://icpmconference.org/2019/wp-content/uploads/sites/6/2019/02/BPIChallenge2019CSV.zip