Applied Energy 325 (2022) 119798

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

ShinyRBase: Near real-time energy saving models using reactive programming

A. Severinsen [a,*], Ø. Myrland [b]

[a] Norwegian University of Life Sciences, School of Economics and Business, Campus Ås, Universitetstunet 3, 1433 Ås, Norway
[b] UiT The Arctic University of Norway, School of Business and Economics, Breivangveien 23, 9010 Tromsø, Norway

HIGHLIGHTS

- Demonstration of an open-source tool using a reactive programming framework for measurement and verification and energy saving models.
- Fast development cycle without any need-to-know web programming languages like HTML, CSS or JavaScript.
- A use case documents energy savings in 40 different Norwegian food retail stores.

ARTICLE INFO

Keywords:
Real-time energy savings evaluation
Building energy retrofitting
Measurement and verification 2.0
Data driven models
Tao Vanilla Benchmarking model

ABSTRACT

To document energy savings from retrofitting a building, a reliable baseline model is needed. The development and implementation of the baseline model is an important step in the measurement and verification (M&V) process. Usually, an energy analyst enters the stage, collects data, do the estimation and delivers the baseline model. The modeling work of the energy analyst is done on either a proprietary or open-source statistical software, often using a coding script. If stakeholders want an updated report on energy savings, the analyst must re-do the whole process, for example on a monthly basis. This workflow is based on an *imperative* programming paradigm. The analyst holds on to the code that performs the analysis and re-run the code when agreed upon. The consequence of this workflow is that stakeholders are dependent on the energy analyst and that updated energy savings results must be planned and scheduled. However, emerging M&V 2.0 technologies enables automation of the energy saving reports. This paper demonstrates how energy savings from retrofitting's in the Norwegian food retail sector is continuously monitored and documented in a web application. The application is built using open-source tools where the baseline model is delivered through a *reactive* programming framework. As an energy savings baseline model, the Tao Vanilla benchmarking model (TVB) was set into production in the web application. The TVB is a linear regression model with well specified features, easy to interpret and has a history of excellent prediction performance. The proposed web application framework allows for a fast development cycle without any need-to-know web programming languages like HTML, CSS or JavaScript. The reactive framework delivers several advantages. First, the stakeholders will always have a current and real-time report on the savings. Second, complex methodologies are dynamically used by the end-user. Third, increased involvement by stakeholders and interaction with the analyst related to the methods used in the energy savings analysis leads to collaborative benefits such as faster disseminating of knowledge. These synergy effect leads to a better technical understanding from the end user perspective and enhanced practical understanding for the analyst. Finally, the paper presents an integrated look at the energy kWh savings versus the cost of the retrofitting's.

1. Introduction

The Intergovernmental Panel on Climate Change (IPCC) objective is to provide governments with scientific information to develop climate policies. The IPCC has a 195 member countries and thousands of contributors. The IPCC scientists assess thousands of published scientific papers each year, and the most recent report is summarized with the following quote,

* Corresponding author.
E-mail addresses: Alexander.severinsen@nmbu.no, alexander.severinsen@nmbu.no (A. Severinsen), oystein.myrland@uit.no (Ø. Myrland).

https://doi.org/10.1016/j.apenergy.2022.119798
Received 11 April 2022; Received in revised form 26 July 2022; Accepted 3 August 2022
Available online 16 August 2022

It is unequivocal that human influence has warmed the atmosphere, ocean and land. Widespread and rapid changes in the atmosphere, ocean, cryosphere and biosphere have occurred [1].

The building sector worldwide uses about 32 percent of all generated energy and 19 percent of energy-related greenhouse emissions [2]. Research show that the potential for increased efficiency in terms of energy cuts are some 30 to 40 percent [3]. Green buildings receive a lot of attention, especially in new design construction. And many new building standards have been developed internationally. For example, the German Passiv House Institute (PHI), Net Zero from the International Living Futures Institutes, Leadership in Energy and Environmental Design (LEED), and R-2000 by Natural Resources Canada. In Norway the standards NS 3700 for residential buildings and the NS 3701 for non-residential buildings apply. These standards are important to make sure that the environmental impact of new buildings stay as low as possible. Still, only about 1 to 3 percent of old buildings are replaced per year [3], thus the existing buildings will still be around in the foreseeable future. According to Statistics Norway the existing building stock in Norway is 4.23 million, and more than 2.6 million of these buildings are non-residential [4]. The number of buildings globally and the potential for energy reductions makes retrofitting an important addition to cut greenhouse gas emission and support the Paris climate agreement. Also, according to analysis by the Rockefeller Foundation and Deutsche Bank's climate change shop the business opportunity in retrofitting is substantial. An investment in the United States of $279 billion in retrofitting buildings could yield more than $1 trillion in energy savings over ten years. This equals 30 percent of the countries annual spending on electricity, and represents a created potential of more than 3.3 million cumulative job years of employment. Nonetheless, the existing upgrade rate is only 2.2 percent each year [3,5].

In this paper we collaborate with a medium sized Norwegian energy service company (ESCO) that has specialized in retrofitting within the retail food sector. The ESCO has recently agreed upon an energy performance contract (EPC) for 40 Norwegian food retail stores. The contract includes a yearly guaranteed energy savings target, and the ESCO also handles all the energy related issues within the contract period, including service and maintenance. The energy conservation measures (ECM) were implemented in the autumn of 2020 and consisted of a mix of the following: change to more efficient LED lightning, new coolers and/or freezers, new heating, ventilation and air conditioning (HVAC) and/or optimization of the stores control systems. The contract has a 10-year duration, and the first year of measuring the energy savings is 2021.

The main contribution of this paper is to demonstrate how to document and monitor energy savings using a dynamic web application based on a open-source reactive programming framework. The development of the application has been done in collaboration with the ESCO and other stakeholders in the EPC project over a two year period. The implementation of the application as a means of setting up baseline models and monitoring the energy savings for the 40 food retail stores has led to several synergy benefits; both collaborative and practical. The energy savings results from the modelling process was seen in real time in the application, on an hourly basis, and automatically updated each day. As a results of this there was no dependency on a energy analyst. The reports was always up-to-date, and the knowledge sharing between participating parties with different skill sets resulted in better baseline models. There was quick detection of unwanted energy increases and follow up of potential errors in the buildings technical system. In short; the web application gave the users a more coherent and reliable process of documenting the energy savings. This is in line with both [6] and [7] who find that web applications facilitate efficient collaboration between scientists and stakeholders and that the cross collaboration between researchers and easy dissemination of results is important for external validity [8].

Today, there are several web-based systems available that can monitor the energy consumption in buildings, but they do not offer any advanced modeling solutions, and are often based on proprietary software [9]. Open-source solutions are not common, and previous solutions focus on visualisations and reporting of electricity consumption, and does not offer baseline energy saving models [10,11]. Up until recently there has been little available research or case studies that document the use of M&V 2.0 energy saving estimation [12]. However, in 2018 [13] presented a residential energy management system, *reEMpy*, that is based on Python. This system is also aimed to energy service companies (ESCOs) to provide a solution that may be used to assess the energy needs of real life use cases by evaluating different algorithmic models, including load forecasting. Nonetheless, our solution differs in terms of the approach. Our proposed framework does not offer code for a final ready-made applications, but a reactive coding framework to allow quickly prototyping new functionality and baseline energy saving models in close collaboration with end-users. Furthermore, [14] proposes a platform for real-time M&V of energy performance. The platform computing tier is developed using the Java development toolkit within the Eclipse software. Conversely, the reactive framework presented in this paper requires no prior development skills. Also, in [14] the platform is demonstrated on one utility, while the usecase for the presented framework, *ShinyRBase*, is given for 40 food retail stores.

The web application *ShinyRBase*, was developed and implemented using R, a free software environment for statistical computing and graphics [15]. R is one of the most popular programming languages for statistics. Furthermore, the R library Shiny [16] was used to develop the reactive programming framework. Shiny makes it very easy to build interactive web applications straight from R without any need to know HTML, CSS or JavaScript. Additionally, Shiny makes it straightforward to use more than 18.000 available packages for a wide range of applications[1] [17]. For example, the popular R packages ggplot [18] and dygraphs [19] is used as tools for interactive visualization and to enhance user interaction with the baseline energy savings model. Also, tidypredict [20] is implemented to save and run predictions for the relevant models inside a database. As will be shown the reactive application gives offers a number of advantages to promote M&V 2.0. The Shiny library, which is the main component in the application, has been in active development since 2016 and has more than 5000 unique peer reviewed works to promote user interaction with scientific research [21].

1.1. Novelty of the paper

The novel contributions of the present papers are threefold. First, the application is developed in collaboration with implementers and utilities. The reactive programming framework, as opposed to the imperative paradigm, offers a flexible and dynamic application that easily can be adapted to different building types. For instance, the vast number of R libraries can be integrated to deliver on changing M&V environments. To be able to convince investors in energy efficiency projects it is critical to provide current and trustworthy energy savings calculations. In that respect, the proposed reactive framework in this paper contributes to the current literature.

Second, the application is demonstrated on a use case continuously monitoring and documenting the energy savings of a large retrofitting project for 40 Norwegian food retail stores. This is the first paper that document the energy savings of a renovation project of this scale within the food retail sector in Norway. Previous research has established that uncertainty regarding the expected savings is a major obstacle that hinder renovations projects [46]. As such, the reactive framework and the documented energy savings in this paper is an example that may motivate new renovation projects, and ultimately produce cost savings

[1] https://cran.r-project.org/web/packages/#:~:text=Currently%2C%20the%20CRAN%20package%20repository%20features%2018316%20available%20packages.

and a reduction of GHG emissions.

Third, the baseline model used to estimate energy savings in the web application is a linear regression model, the Tao Vanilla Benchmarking model (TVB). The model was first published in [23], and was later used in the GEFCom2012 load forecasting competition as one of the top 25 performing contributions [24]. The TVB model is evaluated through the guidelines outlined in The International Performance Measurement and Verification Protocol (IPMVP), developed by the Efficiency Valuation Organization (EVO) [25]. In particular, the IPMVP measure “option C: whole building” is applied. Hence, data from utility meters are used to evaluate the energy performance of the whole building. Note that this option establishes the total savings of all implemented ECMs. In previous research the TVB was used in [26] to document energy savings from retrofitting’s with small expected effects. Further, [27] use the TVB model to document energy savings in 5 different Norwegian food retail stores that undertook major retrofitting’s. However, in the present paper the TVB is implemented to document energy savings for food retail stores within a reactive framework, near-real time for 40 different stores. Given the models previous prediction performance, easy implementation, and the lack of use to estimate savings in retrofitting projects the present paper promotes the novelty of the method as a benchmark model, and adds to the already established data-driven tools within the M&V industry.

The paper is split into 6 different sections. In the first section a presentation of the ESCO and details about the EPC contract and the food retail stores is given. Second, an overview of the data and the features that will be used for the TVB models. Third, the TVB model and the measures used for model performance is outlined. Furthermore, the fourth section gives a detailed presentation on how the Shiny library’s reactive programming framework is used to implement the TVB model. Fifth, the energy savings results from the EPC project is presented. The results are shown as they appear in the web application. Finally, the findings linked to both individual and synergy effects of this project based on user interaction with the application is given. Also, future development improvements that would have given the application more value is presented, both from a scientific and a practitioners point-of-view.

1.2. The ESCO and the energy performance contract

The energy service company (ESCO), Ohmia Retail AS, is a medium sized Norwegian company. The company has developed a product that is marketed towards the food retail sector in Norway as ‘*Energy as a service*.’ The customer pays a fixed monthly fee and need not to worry about necessary equipment investments, insurance and maintenance. Thus, Ohmia Retail takes full responsibility for all the stores technical infrastructure; freezers, coolers, lighting, and the HVAC system. For the food retailer this is a great asset as they can maintain their primary activity, food retail. The ESCO has signed an Energy performance contract (EPC) that includes a guaranteed energy savings for a building portfolio of 40 food retail stores in Norway. The contract has a 10 year duration, and started January 1st 2021. Hence, to document and monitor the guaranteed energy savings it is important for both the ESCO and the customer to utilize reliable methods. Historically the ESCO has used very basic methods to document savings, based on degree day normalization and often using quarterly reports. Thus, the ESCO had great interests in methods that could improve their monitoring workflow, which again motivated this research project.

1.3. Building portfolio, electric load and weather data

Norway has a central repository, Elhub (elhub.no), that daily collects energy use on an hourly level for almost all commercial and household buildings using the advanced metering (AMI) system. It is mandatory for all the Norwegian grid operators to update the central repository every day. This service was launched in February 2019. All the energy data from February 2019 and onward for the 40 buildings in this paper originates from Elhub. The energy data from 2018 up until January 2019 is collected from the building energy management system that was in operation before the launch of Elhub. Outside temperature data is collected from the Norwegian Meteorological Service (https://www.met.no). Each stores position (longitude and latitude) is mapped against a 2.5 km × 2.5 km grid of Norway. The temperature data used in the model stems from the closest weather stations. All energy and weather data is downloaded automatically on an hourly level on a daily basis into a PostgreSQL database. The PostgreSQL is an open-source object-relational database with more than 30 years of active development [28]. This database has a central role in this project. The results from the baseline model is stored, documented and used for daily predictions, all within the database. The next sections present the model and the framework used to set the model into production.

2. Methods

2.1. Estimating energy savings - the baseline model

To estimate the energy savings this paper follows the International Performance Measurement and Verification Protocol (IPMVP) option C: whole building”. Thus, data from utility meters are used to evaluate the energy performance of the whole building [25]. In particular hourly energy and temperature data is used to train a baseline energy saving model. This process involves choosing representative energy data for one whole year before any retrofitting is conducted, and then train a model to predict (after the retrofitting) what the energy consumption would have been without the ECMs. Furthermore, the Tao Vanilla benchmark (TVB) model [23] is used to estimate energy savings. Some energy conservation measures, for example changing to LED lights, may have an expected savings target below 10 %. In these cases the EVO recommends using sub meters as a means of documenting the savings. However, in this project there was no data available from sub-meters. Nonetheless, [26] finds that the TVB model is a good candidate to estimate savings that are below 10 %. Moreover, The TVB model is a well specified regression model and easy to estimate and understand. The following model specification is used:

$$Y_t = \beta_0 + \beta_1 M_t + \beta_2 W_t + \beta_3 H_t + \beta_4 W_t H_t + \beta_5 T_t + \beta_6 T_t^2 + \beta_7 T_t^3 + \beta_8 T_t M_t + \beta_9 T_t^2 M_t + \beta_{10} T_t^3 M_t + \beta_{11} T_t H_t + \beta_{12} T_t^2 H_t + \beta_{13} T_t^3 H_t$$

where Y_t is the actual load for hour t, β_i are the estimated coefficients from the least squares regression method; M_t, W_t and H_t are month of the year, day of the week and hour of the day. Furthermore, T_t is the outside temperature corresponding to time t. Note that the original TVB model includes trend and past loads. In this study the TVB model will reflect how a specific building perform based on a reference period, thus trend and lagged predictors are not included. The simplicity of model choice has several advantages for implementation in the web application. Also, as seen in the next section all the models were estimated within the web application.

Because the load often increases when the temperature drops and when the temperature increases, it is necessary to take this into account. This could be incorporated with linear piecewise functions. However, that would require cut-off temperatures which may be different across different buildings. Thus, this is included in the model using 3rd ordered polynomials of the temperature. Also, the model includes interaction effects between the polynomials of the temperature and the calendar variables *Hour* and *Month*, respectively. The rationale behind is that energy loads may be different when the temperature varies, and there might be differences across different months and hours.

Furthermore, to come up with the model specification, in his thesis [23], Hong tested seven different linear regression models. The testing was demonstrated using the case study of one week ahead hourly forecast for a medium US utility, and the TVB model was found to have best performance. Furthermore, the TVB load forecasting performance was also tested against possibilistic linear models (PLM) [23], p. 117] and artificial neural networks (ANN) [23], p. 136]. The TVB outperformed both modeling alternatives.

The performance of the TVB models is measured with the *coefficient of variation root mean square error* (CV-RMSE). The CV-RMSE is calculated as follows:

$$CV - RMSE = \frac{\frac{\sum(\widehat{Y_i} - Y_i)^2}{n-k-1}}{\overline{Y}}$$

where $\overline{Y}$ is the mean of the measured energy consumption in the training data (the reference year). Y_i is the actual energy use in hour i, $\widehat{Y}_i$ is the predicted value of energy use in hour i from the TVB model, estimated on the baseline period. Furthermore, n is the sample size, and k is is the number of features in the model. This performance measure is recommended by the IPMVP [25]. However, the guideline does not propose a definite threshold that defines a good baseline model. The CV-RMSE has received some critique because the RMSE increase with both the average absolute error and the variance error, which is not desirable [29]. Still, since this paper does not compare between models the CV-RMSE is a good indicator to measure the performance of the models between different training data and different buildings. Furthermore, a "rule-of-thumb" where the CV-RMSE should be below 25 % is applied. In the next section when the web application and the reactive framework is presented it is also shown that within the application it is easy to switch reference year (training data) between the last three years, and the corresponding change in CV-RMSE can be seen instantly. Setting up the model and choosing reference year was done in collaboration with both the ESCO and the building owners. Large CV-RMSE (greater than25 %) was immediately inspected through visualizations of the data in the reference period and could in many instances be led back to the food store being out of operation and/or some extraordinary activity, for example in-store promotion activities. Consequently, the CV-RMSE was actively used to understand the modeling results together with the stakeholders, and for many buildings this collaboration led to a change of reference year and/or a better understanding of the actual data used as training data, eg. missing and/or unusual data.

2.2. The reactive framework energy monitoring solution

This section presents the reactive framework that was used to develop the web application that documents and monitors energy savings. The ESCO and their customers typically depends on an energy analyst to deliver the analysis that documents the energy savings in a project. The usual way of delivering results from statistical models is based on imperative programming, e.g. when $c = a + b$ then c is assigned the sum the terms a, b. If a, b changes then c needs to be re-evaluated to change. However, reactive programming will allow c to be updated instantly when a, b change [30]. This reactivity is the main idea behind the R library Shiny [16]. Shiny helps the energy analyst to promote real-time user interaction with the analysis through a user interface (UI). The Shiny reactive framework allows user input to be evaluated dynamically via the user interface, and the library comes with pre-defined templates for web based user interfaces. This avoids the need to learn web based programming languages such as HTML and/or JavaScript. However, in contrast to the reactive framework, an analyst would write a script that runs all the necessary analysis to deliver a report of the energy savings. Then the analyst has to re-do all the analysis on a regular basis, often monthly. The script is re-run and a new updated report is produced. This way of working is referred to as *imperative* programming. One example of this is seen in the below R and SQL code sequence (comments indicated with a #). In short, the analyst;

1. Selects data needed for the analysis. In this case this means writing an SQL code that extracts energy and temperature data (store_id, date_hour, temperature, kWh) from a database table. The buildings id's and the date intervals has to be specified by the analyst. The query is then run to pull data into a dataset train, using an R function from a database connection library DBI::dbGetQuery(pool, sql).
2. A formula for the baseline model is defined (in this case the TVB model).
3. The regression model is run (through the R function lm) and the results is stored in the object TVB_estimates for further inspection.

2.3. Imperative code sequence

```
# Step 1 - pull data from database to R as specified by script
sql <- (
"SELECT
store_id, hour, temperature, kwh
sum(timeseries_interval_observations.value) AS kwh
FROM energy_data
WHERE id IN ('6754','6789')
AND date BETWEEN '2019-01-01' AND '2019-12-31';"
)
train <- DBI::dbGetQuery(pool, sql)

# Step 2 - define the formula for linear regression model
TVB_formula <- as.formula("kwh ~ hour*weekday + month +
hour*temperature + hour*temperature^2 +
hour*temperature^3 + month*temperature +
month*temperature^2 + month*temperature^3")
# Step 3 - run the linear regression
TVB_estimates <- lm(TVB_formula, data = train)
```

Now, let us look at how the imperative programming approach is handled by R using the Shiny library to create a reactive framework with some simple user inputs in a web application. There are 6 steps involved;

1. The analyst set up the input fields that the user can access and interact with in the user interface (UI). Shiny has pre-canned UI elements that is used to define the UI (radio buttons, date range input fields, checkbox, etc). In addition, the data analyst sets up a UI element to display the analysis (tables, graphics, text). In this simplified example this is only a output field for text where the analyst plans to show the model coefficients from the TVB model.
2. The server environment is defined.
3. Data is selected based on the UI inputs from step 1.
4. The formula is set up (same as in the imperative script).
5. The regression model is run. Every time the user chose new input in the UI, this step is instantly re-run.
6. Set up a render statement to be passed back and displayed in the UI. As for this example; the linear regression models coefficients.

This was a basic workflow example. Yet this process only adds a few lines of extra R code, but turns the analysis into an web application that can be used to interactively chose stores and train data for the TVB models. The part of the web application that will be presented in the results section is about 1000 lines of code. This includes functions to handle pulling and pushing data between R and the database and code for error handling. Strictly speaking, it would be possible to write the application with some 500 lines of code. Hence, in contrast to other web application this is quite efficient.

2.4. Reactive code sequence

```
# Step 1 - Define UI
ui <- fluidPage(
titlePanel("Reactive code sequence"),
sidebarLayout(
checkboxGroupInput(ns("selected_meters"), 'Main meter(s)', choices = NULL),
dateRangeInput(
"Referenceperiod:",
start = ref_start_date,
```

(*continued on next page*)

(*continued*)

```
end = ref_end_date,
),
mainPanel(
textOutput("SQL_model_scoring")
)
)

# Step 2 - Define server environment
server <- function(input, output) {

# Step 3 - pull data inta a R dataframe with a reactive expression
train_data <- reactive({
sql <- glue::glue_sql(
"SELECT
hour, month, temperature, kwh
sum(timeseries_interval_observations.value) AS kwh
FROM energy_data
WHERE id IN ({metering_points*})
AND date BETWEEN {date_from} AND {date_to};",
.con = pool,
.envir = list(
metering_points = selected_meters(), # user selected store
date_from = ref_start_date, # user chosen from-date
date_to = ref_end_date # user chosen to-date
)) %>%
DBI::dbGetQuery(pool, sql)

# Step 4 - define formula for linear regression
TVB_formula <- as.formula("kwh ~ hour*weekday + month +
hour*temperature + hour*temperature^2 +
hour*temperature^3 + month*temperature +
month*temperature^2 + month*temperature^3")

# Step 5 - run linear regression and translate the coefficients to SQL.
# Reactive expression. Will re-run every time a user change the input fields.
model_coefficients_as_sql <- reactive({
train <- train()
TVB_estimates <- lm(TVB_formula, data = train_data)
sql_statement <- tidypredict::tidypredict_sql(TVB_estimates, dbplyr::
  simulate_postgres())
return(model_coefficients_as_sql)
})
output$SQL_model_scoring <- renderText({
print(model_coefficients_as_sql)
})
}
```

In Fig. 1 the reactive framework used to design the web application is presented.

The web application framework consists of 7 different steps.

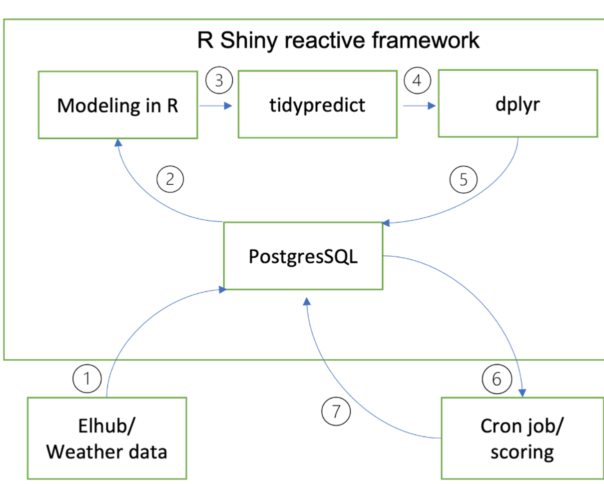


Fig. 1. Reactive framework.

1. Data from the Elhub repository (energy data on an hourly level from the main meter) is stored in a PostgreSQL database. The database is updated daily. This step takes place outside the R/Shiny reactive framework. However, the R library RPostgreSQL [31] handles data transfer between the database and R.
2. Modeling in R. This step is based on user inputs in the Shiny web application. The user chose the relevant building and the period to be used as training data. This is the reference period before the implemented ECMs. Typically, this would be one whole year of data (energy consumption and temperature data) before the installed ECMs. The user then chose the period to be predicted based on the trained model. This involves choosing the date that the ECMs were installed, and the period to be analyzed. Once these choices are made the app automatically runs the TVB model, the CV-RMSE is calculated and presented and the ECM period is scored and visualized (both the actual energy use and the predicted, the difference between these being the savings). Fig. 2 shows the UI that is available to the users. The blue line is the predicted kWh, while the red line is the actual energy consumption.
3. The R library tidypredict [20] reads the current model from step 2, creates a list objects with the necessary components to run predictions, and builds a formula based on the list object.
4. The R library dplyr [32] evaluates the formula through its database backend dbplyr and translates the code into SQL.
5. The translated SQL is stored in a text field in the database with additional details about the building, main meters, and the data used for the models. Not only will this SQL statement be available for continuously scoring new data, but it also works as a documentation for future reference of the model.
6. A cron job (automated Linux job scheduler) wakes up every morning when new data is stored in the database, pulls out the stored SQL from step 5 and use the new data and the SQL statement for new predictions. Hence, the energy savings from the last 24 h in added continuously.
7. The predictions are stored in the database (on an hourly level). This way the database is always updated with the most current data to calculate the energy savings. All the tables and visualizations in the application is based on this; hourly visualizations and aggregates of the results.

See Fig. 2 for a closer look at the UI from step 2 where the user interacts with the web application. In this particular example the reference year was 2018, and the ECM period was 2021. It is important to choose a year for training the TVB model that is representative of each building's energy usage before the installed ECMs. CV-RMSE is the "technical" indicator that guides the model making. However, the interactive user interface that pictures a year of energy and temperature data on an hourly level together with the prognosis from the TVB model made the modeling process more intuitive for the non-analytical building owners and the ESCO. They could observe how changing the reference base impacted the prognosis after the ECMs made the process of setting the reference data for the 40 buildings easier. This interactivity in the process was found to break down the complexity barrier and connect the stakeholders to the modeling process for increased collaborative benefits.

The entire model building process was done together with the ESCO and the building owners. First, a building under scrutiny was chosen, then year of reference was set. The reference year within the portfolio was either 2018, 2019 or 2020. The effect on choosing any of these could be seen immediately by comparing the predicted and the actual values in the time series plot of the UI. For example, one could train a model on energy data from 2018 and then predict the energy consumption in 2019. Given no ECMs in 2019 the actual value and the predictions should follow each other closely. This exercise generated a lot of very useful discussions - both for the building owner and the analyst. For example, the CV-RMSE was above 25 % for many of the models. Often,

Fig. 2. User interface example of TVB in the web application.

this could be related to issues such as; missing data, in-store promotion, fault with the AMI energy meters and/or some other unusual building activity. The visualizations was either performed using the R library ggplot2 [18] or the interface to dygraphs described in [19].

3. Results

3.1. Energy savings

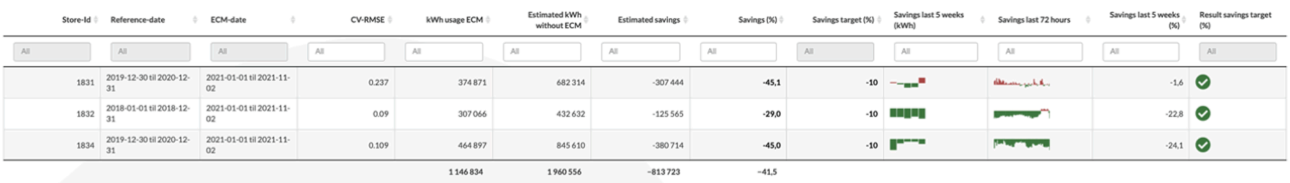
After a user has set up a baseline model with reference data and the ECM period, accepted the CV-RMSE and stored the baseline model, the stores end up in the web application as seen in Fig. 3. This table presents an example from the web application of three of the stores (Store-id 1831, 1832, 1834) with the largest energy savings. The savings are aggregated between 2021 and 01-01 and the current date. The table is updated every day based on the last 24 h. As can be seen the savings range from 45.1 % – 29.0 %. Also, the reference dates, ECM dates and the CV-RMSE for each underlying model are displayed. Additionally, the savings are presented in three different ways. First, the total savings (field: 'Estimated savings'), which is the actual kWh usage minus the predicted in the ECM period. Second, the results the last 5 weeks is presented as a bar plot in the field 'Savings last 5 weeks (kWh).' Each bar represents a week. The bars are colored green if saving energy, and red if actual energy use is above the predicted level. This gives the end user a way of quickly comparing the aggregated result with the results over the last 5 weeks. For example, store '1831′ in the first row has a total energy saving of 45.1 %, but only a 1.6 % savings the last 5 weeks. For this particular store this was somewhat expected as the implemented ECMs was related to winter ECM, since the store had changed the heating system. However, Fig. 3 also shows the savings the last 72 h as bars (each bars equals an hour; green color if saving, and red if higher use than predicted). This indicates that the store needs further investigation into possible causes of higher than expected energy usage.

Table 1 presents the aggregated results for each of the stores ECMs for the first 11 months of 2021, from January 1st 2021 up until November 30th 2021. The three stores with the largest percent energy savings where those with store-id 1522, 1249, and 1520 with energy savings reductions of 46 %, 48 % and 56 %, respectively. However, compared to many of the other stores within this building portfolio the stores undertook quite extensive ECMs, such as both change of lighting, HVAC and the refrigeration system. The three stores with the lowest energy savings where store-id 1572, 1538 and 1653. These three stores actually had an increase in the energy use when comparing with the reference year. However, these results are due to the ESCO replacing oil boilers with heat pumps to heat the buildings (in 2021), hence the increase in electricity consumption. These issues are regulated in the EPC contract. The aggregated results are available to the ESCO at any time in the web application, and updates daily. Every month the ESCO generates a report from the dashboard and has a review meeting together with the customer.

Furthermore, the ESCO wanted a web dashboard that could be used to follow the energy savings in more details from day to day. To deliver on that a visualization that displays the average energy savings for all 24 h of the day, but also split into all week days, ia developed. As can be seen from the top bar plot in Fig. 4 the savings are at its peak at 10:00 with a reduction in energy usage of 25 %. In general it is best at opening hours between 07:00 and 21:00. Further, one may study the savings for each individual day, were Sunday is best (-21,3%) and Tuesday worst (-14.5 %). Also, note that the savings are very low during Tuesday evening and night, and at its worst negative (red bar) on Tuesdays at 23:00. Visual inspection gives a quick overview of the performance of the ECMs and potential for detecting errors and faults.

3.2. Cost-Benefit analysis

So far in this paper one have only considered the actual electricity consumption (kWh) savings. However, from the ESCO point of view, translating these savings into profitability is important. In the EPC project the profitability analysis have been ad hoc and based on manually collecting the relevant data, costs, electricity prices and grid



Store-Id	Reference-date	ECM-date	CV-RMSE	kWh usage ECM	Estimated kWh without ECM	Estimated savings	Savings (%)	Savings target (%)	Savings last 5 weeks (kWh)	Savings last 72 hours	Savings last 5 weeks (%)	Result savings target (%)
All	All	All	All	All	All	All	All	All	All	All	All	All
1831	2019-12-30 til 2020-12-31	2021-01-01 til 2021-11-02	0.237	374 871	682 314	-307 444	**-45,1**	**-10**			-1,6	
1832	2018-01-01 til 2018-12-31	2021-01-01 til 2021-11-02	0.09	307 066	432 632	-125 565	**-29,0**	**-10**			-22,8	
1834	2019-12-30 til 2020-12-31	2021-01-01 til 2021-11-02	0.109	464 897	845 610	-380 714	**-45,0**	**-10**			-24,1	
				1 146 834	**1 960 556**	**−813 723**	**−41,5**					

Fig. 3. Energy savings as presented in the web application.

rent, and then performing the analysis in a spreadsheet. However, since the underlying data mostly is available through web API's and is continuously updated it is a natural next step to develop a profitability dashboard as an extension of the web application. Thus, in this section the available data is presented, prices and rent, and how these can be integrated with the kWh savings (as reported in the previous section). And at last a benefit cost analysis that should be relatively straightforward to implement in the *ShinyRBase* web application is offered.

3.3. Electricity prices and grid rent

Electricity prices in the end-user market in Norway comprise of physical power and grid rent for transmission of the electricity by the local grid company. Nord Pool spot is the marketplace for physical electricity contracts, and is the place where the Norwegian electricity spot prices are set. The electricity prices used in this paper were downloaded from Nord Pools website as a spreadsheet.[2] However, they do have available an API that can be used to automate this step within an application.[3] Furthermore, the second price component of electricity consumption is grid rent. The food retail stores in this paper belongs to different grid owners, and to simplify-one of the largest grid owners in the area, *Vevig AS* is chosen, as a basis for calculating grid rent for all of the relevant stores. A summary of Vevig's tariffs is presented in Table 2 below. For further details please see: https://vevig.no/nettleie-og-vilkår/nettleie-næring.

The effect prices are weighted based on months. For instance, in January, February, November and December these weight equals 1. Further, between March and October the weights vary from 0.9 to 0.6. Hence, given a store with a maximum load of 210 kW in March the effect price is $210 \times 0.9 \times 55.9 = 10\,564$. Hence, there is a "penalty" for larger loads in the colder months. The end user's total electricity bills also consists of a fee earmarked for the energy fund Enova SF's (owned by the Ministry of Climate and Environment) work to reduce greenhouse gas emission and to strengthen security of supply. There is also a variable fee for electricity certificates. This fee depends on the developments in the electricity certificate market. At last there is a consumption tax on electricity.

It is not easy to automate the collection of grid rent data. There are more than 100 grid owners in Norway and many of them have different grid rent pricing strategies. Nonetheless, the pricing scheme often follow the same pattern as presented for Vevig. Hence, summer prices versus winter prices, and typically a penalty for larger loads, in particular in the winter. Probably the best solution would be to develop a user interface in the web application where the grid rent is based on user input.

3.4. Benefit cost analysis

In this section a closer look at the benefit cost analysis is presented. Unfortunately, due to confidentiality issues we are not allowed to share the details behind the actual cost elements. However, it is still found useful to propose a general method on how to approach this EPC project, in particular as the customer pays a monthly fee that includes the guaranteed energy savings, but also service and maintenance over a 10 year period. The results presented in this section is for the period January 1, 2021 - November 30, 2021, thus an 11 month period.

Usually when investigating the profitability of a project one can look at the Present Value (*PV*). This is defined as:

$$PV = X_0 + \sum_{t=1}^{T} \frac{X_t}{(1+r)^t}$$

where X_0 is the initial cost of the project, X_t is the net cash flow generated by the project for $t = 1, \dots, T$ periods, and r is the discount rate. However in this case, there is no initial cost like an investment in new equipment. This is done by the ESCO, and the *customer* pays a monthly lump sum for all 40 stores for the 10 years of the energy performance contract.

This "subscription" to energy savings and new improved equipment through the energy contract makes the standard way of looking at cost/benefit measures obsolete. The monthly lump sum cost paid by the *customer* to the ESCO can be considered an annuity over the 10 year period. It is like leasing a car. The monthly payment covers the ESCO's investments in the contract in addition to their profits. The benefits for the *customer*, since there is no initial investments, are "avoided" costs. These benefits for the *customer* are taken into account by looking at the investments costs by the ESCO. The *customer* gets benefits from reduced electricity costs and avoided administrative costs for equipment maintenance.

In terms of the actual implemented ECMs, all the stores changed to more efficient LED lightning, nine stores got new refrigeration systems, five stores got new heating, ventilation and air conditioning (HVAC). In addition the *customer* avoids insurance costs on the refrigeration system.

In order to calculate the benefits the present value (*P*) of an annuity based on the investments (*PV*) by the ESCO is calculated as:

$$P = PV \times \frac{\frac{r}{k}}{1 - \left(1 + \frac{r}{k}\right)^{-nk}}$$

where r is the annual interest rate, k is the number of compounds per year (12 months), and n is the number of years.

As an example, if you need to invest $space(100000)$ (*PV*) in new lights in a store today, the present value (*P*) of an annuity at monthly installments (k) for 10 years (n) at 2.5 % interest (r) is 942.7 per month.

The project has run for 11 months. In Fig. 5 (panel **A)** the monthly benefit cost ratio (*BCR*) is calculated. The benefits consists of two parts. The value of savings in energy costs. This is a variable part. And the fixed benefits from "avoided" costs as discussed above.

The average *BCR* is 1.48. Further, there is an increase in the ratio the last four months. The average percentage savings in kWh is some 13.52 % (panel **B**). This indicates that the variable component in the BCR is the price of electricity. Also, panel **C** shows that there is an increase in the average price of electricity. In the same panel (**C**) one also see that the variable portion of the BCR is above one in September.

The finding that electricity costs is contributing a relative high share in the benefit cost calculations is interesting, even when the energy savings in this period is between 11 and 16 %. One might expect higher electricity prices in the future, making the contributions of energy savings relatively large in the cost benefit perspective. Note also that before the building owner singed up for the EPC project they handled the

[2] https://www.nordpoolgroup.com/4ab28c/globalassets/marketdata-excel-files/elspot-prices_2021_hourly_nok.xls.

[3] https://www.nordpoolgroup.com/trading/api/.

Table 1
Energy savings as of November 30 2021.

Store-ID	Ref. year	CV-RMSE	Tot. kWh	Tot. prog. kWh	Savings (kWh)	Savings (%)
1522	2018	0,178	413 439	934 539	−521 100	−55,760
1249	2018	0,086	262 384	508 378	−245 994	−48,388
1520	2018	0,078	505 378	940 101	−434 723	−46,242
1546	2018	0,088	100 448	183 416	−82 968	−45,235
1560	2019	0,131	388 999	579 186	−190 187	−32,837
1555	2018	0,139	259 624	378 003	−118 379	−31,317
1551	2018	0,114	652 130	901 184	−249 054	−27,636
1524	2018	0,09	345 700	476 985	−131 284	−27,524
1573	2019	0,124	1 429 274	1 967 543	−538 269	−27,357
1562	2018	0,136	861 749	1 172 593	−310 844	−26,509
1563	2019	0,108	361 473	467 376	−105 903	−22,659
1526	2018	0,086	350 440	452 670	−102 230	−22,584
1557	2018	0,077	285 757	369 029	−83 272	−22,565
1531	2019	0,073	184 752	229 118	−44 366	−19,364
1556	2018	0,096	222 500	275 873	−53 373	−19,347
1566	2018	0,106	561 216	693 902	−132 686	−19,122
1681	2018	0,138	182 268	223 251	−40 983	−18,357
1569	2020	0,213	978 826	1 192 884	−214 057	−17,945
1548	2018	0,122	109 926	133 574	−23 648	−17,704
1536	2018	0,119	91 541	110 778	−19 237	−17,366
1540	2018	0,101	204 746	241 196	−36 450	−15,112
1528	2018	0,125	346 505	398 251	−51 746	−12,993
1731	2018	0,143	1 324 183	1 512 011	−187 828	−12,422
1564	2018	0,087	426 011	484 579	−58 569	−12,087
1552	2018	0,149	549 568	623 214	−73 646	−11,817
1554	2018	0,129	255 964	289 778	−33 814	−11,669
1533	2019	0,171	877 886	988 021	−110 135	−11,147
1529	2018	0,107	132 523	147 924	−15 400	−10,411
1542	2018	0,121	80 213	88 055	−7 842	−8,905
1549	2018	0,093	104 823	112 850	−8 027	−7,113
1545	2018	0,122	217 618	233 760	−16 143	−6,906
1547	2018	0,136	213 985	223 827	−9 842	−4,397
1553	2019	0,118	198 001	206 148	−8 147	−3,952
1640	2019	0,073	718 989	744 872	−25 883	−3,475
1558	2018	0,102	259 053	258 983	70	0,027
1532	2018	0,127	100 878	99 774	1 104	1,106
1527	2018	0,1	2 594 299	2 547 885	46 414	1,822

Table 1 (*continued*)

Store-ID	Ref. year	CV-RMSE	Tot. kWh	Tot. prog. kWh	Savings (kWh)	Savings (%)
1653	2018	0,11	129 735	127 324	2 411	1,893
1538	2018	0,095	157 216	152 955	4 261	2,786
1572	2019	0,189	580 417	545 210	35 208	6,458

insurance, service and maintenance internally. This was often a time involving process, and for many food retailers taking away the burden of handling this is also important to recognize - an element difficult to integrate into the benefit cost analysis.

All the analysis in this section was conducted using the R library lifecontingencies [33], and as such the library is easy to fully integrate into the Shiny web application. Also, the library has many other options for financial analysis that may be incorporated.

4. Discussion

Developing the *ShinyRBase* framework and application outlined in this paper has been a two year research project in close cooperation with the ESCO. Weekly status meetings were done throughout the process. The application was in fully operation in January 2021. Hence energy savings for the 40 food retail stores has been monitored on a day to day basis since then. Note that the savings are estimated in the context of measuring energy use at the whole facility. The interaction when setting up the baseline model has resulted in knowledge sharing for both sides. First and foremost, the main objective from the ESCO points of view was to have a web application that could be used to closely follow the energy savings, and to make sure that they complied with the guaranteed savings as agreed upon in the EPC contract. The ESCO did not share information about how they approached the energy audit and the calculations of the potential energy savings within the buildings under study. Often, in the audit stage of the retrofitting project simulations of the energy savings could be conducted through software such as EnergyPlus.[4] It would be useful to have access to such simulations as a comparison between the simulations and the predicted savings could be used to adjust the simulations, and to enhance the understanding of the effect of the retrofitting. Also, the provided energy data was only for the whole building, and access to sub-meters (e.g., ventilation, refrigeration, lighting) could further improve the energy saving analysis.

Furthermore, during 2021 the application was used to discard training data due to high CV-RMSE (visual inspection). In this process the application detected: lights that were on during the night, ventilation that was running in day-mode during night, in-store promotions in the training data, AMI meters that stopped working or had an technical error that resulted in to much reported electricity, and errors in the Elhub data repository that gave zero electricity reported, but also some very high levels of energy consumption (large hourly peaks). Some of these errors would have been possible to detect with a standard energy monitoring system (EMS) and/or through more standard reports from a energy analyst. However, the continuously real-time aspect that the web application enabled has been a great advantage to facilitate such findings. At the same time, building owners may benefit from other automated approaches such as occupant-building interaction via smart zoning of thermostatic loads or demand management via distributed control. Still, extending the use of energy savings baseline models such as TVB (within a reactive programming paradigm) may prove additional information that can be used to detect potential errors in the technical infrastructure. For instance, given that a model was trained on data from, for example, the refrigeration system within a time span when the

[4] https://energyplus.net.

Fig. 4. Optimization - following the energy savings over days and hours.

Table 2
Vevig grid rent tariffs.

Description	Price
Fixed price (NOK, monthly)	2048
Effect price (NOK)	55.9
Winter (Øre)	0.063
Summer (Øre)	0.042
Consumption tax (Øre)	0.1669
Enova SF energy fund (NOK, yearly	800

system was running under optimal settings, the prediction can be set into production and the running difference between the actual energy consumption may be an indication of some non-optimal setting in the system.

Working together with the different stakeholders, the ESCO and the building owners, has surely increased the overall knowledge about the drivers of energy savings within the EPC project. These synergy effects ended up in a more usable web application, and as a tool for error detection. The app in itself has integrated the different stakeholders in ways that otherwise would have been difficult. Hence, the reactive framework streamlines the M&V process and deliver significant value. These findings echo the study by [34]. Furthermore, an important feature of M&V 2.0, as [12] also notes, in many energy efficiency projects there is a time lag between implementation of the ECMs and the evaluation of the savings. This lag hinders on-going changes in the ECMs that may further reduce the energy savings, e.g. optimize the control units in a HVAC. For ESCOs that has energy saving contracts the opportunity to identify and correct these failures may increase payments.

The *ShinyRBase* application was used with the following workflow. The user choose a baseline model based on a reference year. Then the model was evaluated using the CV-RMSE and by visual inspection of the actual versus predicted energy consumption on an hourly level. When the user was happy about the quality of the model they saved the model. The model was then automatically scored on a daily basis when new energy and temperature data was stored in the database. This was repeated for each of the 40 stores. The stakeholders used a web dashboard where they could monitor energy savings, both aggregated, the last 5 weeks and the last 72 h. This process was not flawless. There were several models whose predictions made little sense. However, this was either when the CV-RMSE was larger than 25 %, or when predictions was made using temperatures in the ECM period that was not present in the training data. Today, the application handle this with some basic rules that rolls back the predictions made when the feature space is not "fully covered." The roll back is very basic and just takes the last predictions (1 h back). But so far this has been a successful workaround.

4.1. Alternative baseline models

What about the TVB model and the potential need for more advanced models in operation? In this research this is an issue approached very carefully. Current research from field experience show that interpretability of models may keep the clients from accepting a black-box model (ex. artificial neural network) [35]. Furthermore, throughout this project it has been important to balance the scientific perspective with the stakeholders practical perspective. For example, it has not been easy to argue for more advanced models as the end-user already had concerns about the TVB model. Typically, the users wanted to use available development resources to enhance the visualization and layout in the application. Nonetheless, the modeling approach should be further developed taking into consideration more recent research findings. For instance, as [36] points out, a one-size-fits-all model is not realistic to be reliable across different building types. Hence, the web application should make it easier to compare models of different complexity because the user instantly gets a visualizations of the black-box model, for example a time-series plot with the actual values versus the predicted values. The visuals is a potential solution to break down this barrier. For instance, particle swarm optimization, similar to what was implemented in [14] could be integrated into the reactive framework through R libraries such as pso or psoptim [37,38].

The methods that are currently reviewed and tested is based on the main findings from the ASHRAE 2019 Kaggle competition "Great Energy Predictor III. How much energy will a building consume?"[5] This competition attracted 4,370 participants from 94 countries. The prize money for the winning team was $25,000. A detailed overview of the machine learning workflows and the winning teams are presented in [39]. The top 5 solutions were reproduced by [39] and the

[5] https://www.kaggle.com/c/ashrae-energy-prediction.

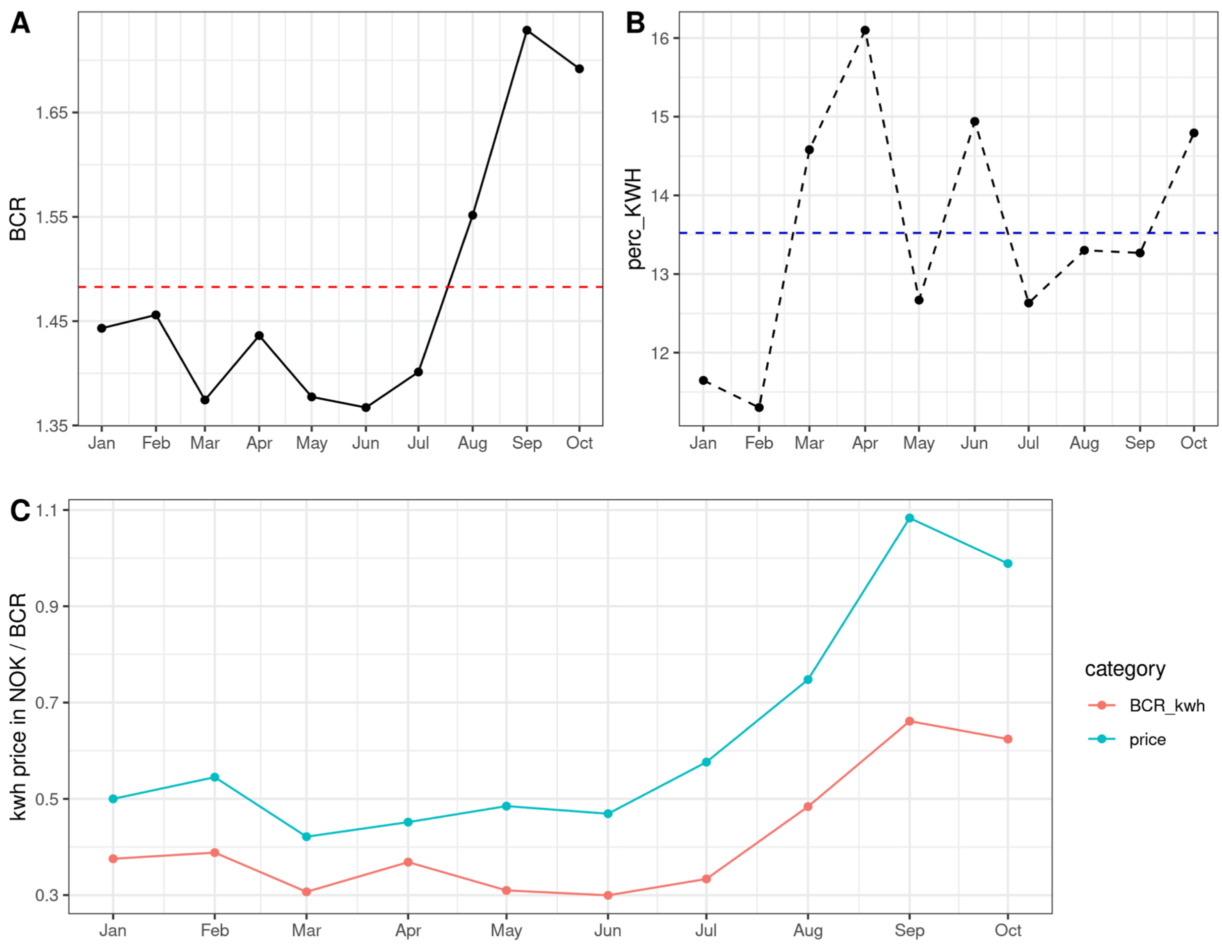


Fig. 5. BCR and energy savings in the first 11 months of the EPC project.

accompanying code can be found on github.[6] The winning solutions are presented in Table 3. As can be seen 4 out of 5 used multiple methods and post-processing of data with ensembling and weighting. All the winning solutions used Light GBM, and three of the winning teams used Catboost, and two used XGBoost.

The methods used to compete in the Kaggle competitions are quite technical. However, the winning solutions where all coded in Python [40], which is also open-source, and the code for the winning solutions can found in the above mentioned github repository for easy reproduction. The teams used pre-canned solutions (modules) to train the models, and replication in other settings should therefore be relatively straightforward. For example, all models are also available in R; lightgbm [41], xgboost [42], catboost (not part of CRAN yet, but can be downloaded as a development version from the catboost.ai website[7]), MLP (multi-layer perceptron) [43]. Most of the ensambling and weighting can be handled in the stacks package [44]. In a bi-lingual R and Python team, or if a method is only available in Python it is possible to use the R package reticulate [45], which provides a comprehensive set of tools for interoperability between Python and R. For example you can call Python from within R, translate between R and Python objects and there are flexible bindings to different versions of Python, both virtual and Conda environments.

As previously mentioned it is beneficial to have the coefficients from the models saved into a database for in-database continuously scoring

Table 3
Kaggle top 5 performing teams - modeling solutions.

Rank	Team	Features	Modeling	Post-processing
1	Matthew Motoki and Isamu Yamashita (Isamu and Matt)	28 features	CatBoost, LightGBM, and multi-layer perceptron	Ensembled the model predictions using weighted generalized mean
2	Rohan Rao, Anton Isakin, Yangguang Zang, and Oleg Knaub (cHa0s)	Temporal features, building metadata, statistical features of weather data	Catboost, XGBoost, LightGBM, and Feed-forward Neural Network	Weighted average
3	Xavier Capdepon (eagle4)	21 features including raw weather and meta data	Catboost, Keras CNN, LightGBM	Weighted average
4	Jun Yang (不用leakage 上分太难了)	23 features weather lag features and aggregates	XGBoost and Light GBM	Ensembles. Weights were determined using the leaked data
5	Tatsuya Sano, Minoru Tomioka, and Yuta Kobayashi (mma)	Target encoding using percentile and proportion and the weather data temporal features	LightGBM	Weighted average

[6] https://github.com/buds-lab/ashrae-great-energy-predictor-3-solution-analysis.
[7] https://catboost.ai/en/docs/installation/r-installation-binary-installation.

and monitoring of the ECMs. This is in the application handled by the package tidypredict, where xgboost, random forest and tree models are already a part of the tidypredict library. However, catboost, light-GBM and MLP is presently not possible to translate into SQL. Still, running predictions inside databases based on this is relatively straightforward to implement. While future work will carefully review these methods potential to improve the baseline models, the question raised by the ESCO several times along the project, "will more advanced modeling approaches enable the application to deliver more reliable results?" must also be recognized. To answer this there is a need to balance predictive accuracy versus the value of a slightly better model. For instance, [29] points out that because of cost and constraints, the stakeholders are less prone to embrace innovations from a modeling point of view. Often they chose a very simplified model to estimate energy savings (eg. just one temperature variable). This is important to take into consideration. For example, the ESCO in this study was used to using energy - temperature curves from weekly aggregated data with a regression model based on only the average weekly temperature as an independent variable. In this study, the web application made it much easier for the ESCO's transition from this model to an hourly based TVB model. To illustrate, they were given the chance to play around with the TVB model and to compare the overall results with the ET-curves (which gave very similar results). This comparison gave the user confidence as well as they could also see the extra benefits of looking at the results from an hourly perspective.

4.2. Baseline models in web applications - advantages and disadvantages

The advantages and disadvantages to deliver analytics using a web application instead of as a static report from an energy analyst can be divided into two phases; the development phase and the phase when the application is in production. During development one find that:

1. The stakeholder is closely involved in deciding what information will be presented in the application, and is able to try early versions of the app interactively. This leads to a strong ownership for the end product. However, this phase requires more involvement and is time consuming.
2. Different parties have different skill-sets, and working together during the development process means that those skill-sets are reflected in the application. These are collaborative benefits that may be challenging to achieve otherwise. Because the application is real-time and dynamic there is an instant feedback from the users that may be difficult to capture in a workflow where the analysis is delivered as a static report. This interactivity was found to "trigger" curiosity and need for more information. It is easier for the user to "play" with the application than to order a new report with more information.

Several advantages were found during the phase where the application is finished and set into production:

1. The results are available at the users convenience.
2. Increased efficiency for the analyst and the stakeholder. Less repetitive work for the analyst and no user dependence on the analyst to deliver.
3. The models are documented in the database for easy reproduction and daily scoring.
4. Because it is possible to closely follow the ECM's actual versus predicted consumption on an hourly level in real-time, the application is not only used to monitor the energy, but also to optimize parts of the technical system.
5. Several errors in the technical system was quickly detected.

Furthermore, the reactive framework as offered by the Shiny application and adjoining R libraries enabled fast prototyping of different solutions, web dashboards and ways to visualize and report on the TVB model. Compared to the imperative programming scheme with static reports and a strict dependence on the energy analyst the proposed framework has proven valuable.

5. Conclusions

This paper demonstrates the development of a web application, *ShinyRBase*, using a reactive framework to document and continuously monitor and benchmark energy savings for 40 food retail stores in Norway. Using open source tools, R, Shiny and adjoining libraries, this process was relatively straightforward, compared to the more standard way of delivering energy savings report. There is no need to know HTML, CSS or JavaScript to do this. The reactive framework within the Shiny library and the automated way of developing a user interface handled those aspects. The end-user was trained to make them self-sufficient in terms of setting up baseline models for the different buildings and to continuously monitor the energy savings. The baseline models was based on a well specified linear regression, the Tao Vanilla Benchmarking model.

Complex methodologies was instantly used by the end user without the need of advanced computation skills. The development and the use of the application promoted collaboration between practitioners (the ESCO and the customers) and the researcher/analyst. This collaboration resulted in an app that was fit for purpose because of the advise and the on-going interactive use from the collaborators. The advantages was twofold. First, during development the stakeholders took part of the process, which resulted in increased ownership and engagement. The different participating parties had different skill-sets, and working together during development those skill-sets ended up in a final application that was more relevant. Second, after the application was in production there were several other advantages compare to using a standard report to follow the energy savings. The users could look into the results at their own convenience, and always had fresh and current updates that was easy to monitor. Thus, a more efficient workflow for both the energy analyst (less repetitive work) and the end users (self-sufficient). The parameter estimates from the linear regression models for the different stores was saved in a table in the database (as a SQL query). This worked both as a documentation of the models, and as a useful way of scoring and updating the results every day. Hence, having the models documented in a database gave both easy and reliable reproduction of the models. Finally, since the results was always current and it was easy to closely monitor the savings, it was also easy to detect when the savings trended negatively. This enabled the users to quickly detect several errors in the technical system, such as ventilation in day-mode during night and lights that was not turned off. Some caveats should be mentioned. Even though the Shiny app simplifies the process of setting up a reactive framework there are still some added complexity. This way of working is more involved and time-consuming during the development phase, and running the app in production mode requires knowledge about setting up a server environment.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was supported by the Norwegian Research Council (NRC) through the Industrial PhD project [grant number 283002]. Opinions expressed and conclusions arrived at are those of the authors, and are not necessarily to be attributed to the NRC. Also, thank you to the ESCO Ohmia Retail AS for access to energy data, and many helpful discussions throughout the project.

References

[1] Allan RP. Climate change 2021: The physical science basis : Working group I contribution to the sixth assessment report of the intergovernmental panel on climate change. IPCC Secretariat: WMO; 2021.
[2] Sánchez-Garcia D, Rubio-Bellido C, del Río JJM, Pérez-Fargallo A. Towards the quantification of energy demand and consumption through the adaptive comfort approach in mixed mode office buildings considering climate change. Energy Build 2019;187:173–85.
[3] Hawken P, editor. Buildings and cities: retrofitting. In: Drawdown: The most comprehensive plan ever proposed to reverse global warming. Penguin; 2017. p. 102–3.
[4] Statistics norway building stock. https://www.ssb.no/en/bygg-bolig-og-eiendom/bygg-og-anlegg/statistikk/bygningsmassen.
[5] United states building energy efficiency retrofits: Market sizing and financing models. Rockefeller Foundation; 2012.
[6] Wages NA, Petroni GR. A web tool for designing and conducting phase I trials using the continual reassessment method. BMC Cancer 2018 Feb;18(1):133.
[7] Klein T, Samourkasidis A, Athanasiadis IN, Bellocchi G, Calanca P. webXTREME: R-based web tool for calculating agroclimatic indices of extreme events. Comput Electron Agric 2017 Apr;136:111–6.
[8] Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, Percie du Sert N, et al. A manifesto for reproducible science. Nat Hum Behav 2017;1(1).
[9] Lobaccaro G, Carlucci S, Löfström E. A review of systems and technologies for smart homes and smart grids. Energies 2016 May;9(5):348.
[10] Google powermeter [Internet]. Google; 2016. Available from: https://developers.google.com/powermeter/.
[11] EmonCMS. Open-source energy visualisations [Internet]. EmonCMS; 2021. Available from: https://emoncms.org/.
[12] Franconi E, Gee M, Goldberg M, Granderson J, Guiterman T, Li M, et al. The status and promise of advanced M&V: An overview of “M&V 2.0” methods, tools. and applications 2017.
[13] Fagiani M, Severini M, Valenti M, Ferracuti F, Ciabattoni L, Squartini S. rEMpy: A comprehensive software framework for residential energy management. Energy Build 2018 Jul;171:131–43.
[14] Ke M-T, Yeh C-H, Su C-J. Cloud computing platform for real-time measurement and verification of energy performance. Appl Energy 2017 Feb;188:497–507.
[15] R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2020. Available from: https://www.R-project.org/.
[16] Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, et al. Shiny: Web application framework for r [Internet]. 2021. Available from: https://CRAN.R-project.org/package=shiny.
[17] Li J, Cui B, Dai Y, Bai L, Huang J. BioInstaller: A comprehensive R package to construct interactive and reproducible biological data analysis applications based on the R platform. PeerJ 2018 Oct;6:e5853.
[18] Wickham H. ggplot2: Elegant graphics for data analysis [Internet]. New York: Springer-Verlag; 2016. Available from: https://ggplot2.tidyverse.org.
[19] Vanderkam D, Allaire J, Owen J, Gromer D, Thieurmel B. Dygraphs: Interface to 'dygraphs' interactive time series charting library [Internet]. 2018. Available from: https://CRAN.R-project.org/package=dygraphs.
[20] Kuhn M. Tidypredict: Run predictions inside the database [Internet]. 2020. Available from: https://CRAN.R-project.org/package=tidypredict.
[21] Kasprzak P, Mitchell L, Kravchuk O, Timmins A. Six years of shiny in research - collaborative development of web tools in R. The R Journal 2020 Dec;12(2):20–42.
[23] Hong T. Short term electric load forecasting [Internet] [PhD thesis]. North Carolina State University; 2010. Available from: https://repository.lib.ncsu.edu/handle/1840.16/6457.
[24] Hong T, Pinson P, Fan S. Global energy forecasting competition 2012. Int J Forecast 2014 Apr;30(2):357–63.
[25] EVO Efficiency Valuation Organization. International performance measurement and verification protocol. 2016.
[26] Severinsen A, Hyndman RJ. Quantification of energy savings from energy conservation measures in buildings using machine learning. In: ECEEE summer study proceedings. 2019. p. 757–66. https://www.eceee.org/library/conference_proceedings/eceee_Summer_Studies/2019/4-monitoring-and-evaluation-for-greater-impact/quantification-of-energy-savings-from-energy-conservation-measures-in-buildings-using-machine-learning/.
[27] Severinsen A, Myrland Ø. Statistical learning to estimate energy savings from retrofitting in the Norwegian food retail market. Renew Sustain Energy Rev 2022;167:112691.
[28] The PostgreSQL Global Development Group. PostgreSQL [Internet]. 2021. Available from: https://www.postgresql.org.
[29] Agenis-Nevers M, Wang Y, Dugachard M, Salvazet R, Becker G, Chenu D. Measurement and verification for multiple buildings: An innovative baseline model selection framework applied to real energy performance contracts. Energy Build 2021;249:111183.
[30] Grolemund G. Shiny - how to understand reactivity in r [Internet]. 2016. Available from: https://shiny.rstudio.com/articles/understanding-reactivity.html.
[31] Conway J, Eddelbuettel D, Nishiyama T, Prayaga SK, Tiffin N. RPostgreSQL: R interface to the 'PostgreSQL' database system [Internet]. 2017. Available from: https://CRAN.R-project.org/package=RPostgreSQL.
[32] Wickham H, François R, Henry L, Müller K. Dplyr: A grammar of data manipulation [Internet]. 2020. Available from: https://CRAN.R-project.org/package=dplyr.
[33] Spedicato GA. The lifecontingencies package: Performing financial and actuarial mathematics calculations in R. Available from Journal of Statistical Software [Internet] 2013;55(10):1–36. https://www.jstatsoft.org/v55/i10/.
[34] Granderson J, Price PN, Jump D, Addy N, Sohn MD. Automated measurement and verification: Performance of public domain whole-building electric baseline models. Appl Energy 2015 Apr;144:106–13.
[35] Molnar C. Interpretable machine learning a guide for making black box models explainable. https://leanpub.com; 2021.
[36] Cui C, Wu T, Hu M, Weir JD, Li X. Short-term building energy model recommendation system: A meta-learning approach. Appl Energy 2016 Jun;172:251–63.
[37] Bendtsen CP. Available from: Particle swarm optimization [Internet] 2012. https://CRAN.R-project.org/package=pso.
[38] Psoptim CK. Available from: Particle swarm optimization [Internet] 2016. https://CRAN.R-project.org/package=psoptim.
[39] Miller C, Arjunan P, Kathirgamanathan A, Fu C, Roth J, Park JY, et al. The ASHRAE Great Energy Predictor III competition: Overview and results. Science and Technology for the Built Environment 2020;26(10):1427–47.
[40] Python Core Team. Python: A dynamic, open source programming language [Internet]. Python Software Foundation; 2019. Available from: https://www.python.org/.
[41] Ke G, Soukhavong D, Lamb J, Meng Q, Finley T, Wang T, et al. Lightgbm: Light gradient boosting machine [Internet]. 2020. Available from: https://CRAN.R-project.org/package=lightgbm.
[42] Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, et al. Xgboost. Available from: Extreme gradient boosting [Internet] 2021. https://CRAN.R-project.org/package=xgboost.
[43] Bergmeir C, Benítez JM. Neural networks in R using the stuttgart neural network simulator: RSNNS. Available from Journal of Statistical Software [Internet] 2012;46(7):1–26. https://www.jstatsoft.org/v46/i07/.
[44] Couch S, Kuhn M. Stacks: Tidy model stacking [Internet]. 2021. Available from: https://CRAN.R-project.org/package=stacks.
[45] Ushey K, Allaire J, Tang Y. Reticulate: Interface to 'python' [Internet]. 2021. Available from: https://CRAN.R-project.org/package=reticulate.
[46] Kontokosta CE. Modeling the Energy Retrofit Decision in Commercial Office Buildings. Energy Build 2016;131(November):1–20. https://doi.org/10.1016/j.enbuild.2016.08.062.