Exploring regression models to enable monitoring capability of local energy communities for self-management in low-voltage distribution networks

Citation for published version (APA):
Mai, T. T., Nguyen, P. H., Haque, N., & Pemen, G. (2022). Exploring regression models to enable monitoring capability of local energy communities for self-management in low-voltage distribution networks. IET Smart Grid, 5(1), 25-41. https://doi.org/10.1049/stg2.12049

DOI:
10.1049/stg2.12049

Document status and date:
Published: 01/02/2022

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 28. Sep. 2022
Exploring regression models to enable monitoring capability of local energy communities for self-management in low-voltage distribution networks

Tam T. Mai\textsuperscript{1} | Phuong H. Nguyen\textsuperscript{1} | Niyam A. N. M. M. Haque\textsuperscript{1,2} | Guus A. J. M. Pemen\textsuperscript{1}

\textsuperscript{1}Electrical Energy Systems (EES) Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
\textsuperscript{2}ASML B. V., Veldhoven, The Netherlands

\textbf{Abstract}
This study proposes a data-driven approach to enable the self-management capability of local energy communities (LECs) via transformer congestion monitoring in low-voltage distribution networks. A set of regression models is adopted in this approach, while the data from residential smart meters (SMs) is leveraged. Four machine learning algorithms, namely ridge regression, support vector regression, random forest regression (RFR) and eXtreme gradient boosting regression (XGBR), are compared to select the best-performing regression model using a cross-validation method. A comprehensive framework is provided to facilitate comparison of the algorithm, consisting of data preprocessing, model fitting and validation, and model deployment. A thorough analysis is also SMs' measurements. The obtained results highlight that the regression-based method can effectively estimate the transformer loading, that is with the Pearson correlation coefficient $R$ and root mean square error calculated for the real values and the estimated values of around 0.98 and 0.87, respectively, by using only a limited set of SM measurements (5 out of 21 SMs used) provided by the LECs while preserving customers’ privacy rights. Among the examined algorithms, the XGBR algorithm appears the best method as it achieves adequate accuracy at significantly less simulation time (i.e. one-third of the simulation time of the RFR). By applying the proposed approach, the monitoring and self-management capability of the LECs can be realised.

\textbf{KEYWORDS}
learning (artificial intelligence), power system management, smart power grids

\section{INTRODUCTION}
With the inherent intermittency and randomness, the rapid proliferation of renewable energy sources (RES), especially solar photovoltaic (PV), in low-voltage (LV) distribution networks creates operational difficulties for the distribution system operators (DSOs) \cite{1}. One of these difficulties is the issue of congestion or thermal overloading of the network assets, for example lines, cables, and transformers \cite{2}, which likely occurs in urban power networks with high population density \cite{3}. As reported in \cite{4}, nearly 90\% of medium-voltage (MV)/LV distribution transformers in the Netherlands will become prone to overloading in 2040 caused by the new load and generation technologies, for example electrified heat-pumps, electric vehicles, and solar PV. The congestion issues can become a significant barrier to further deployments of RES in LV networks. Hence, proper monitoring of such congestions is critical for the DSOs to manage new load connections and network asset upgrading \cite{5}. The DSOs and the end users should be aware of the congestions to make informed decisions on controlling their...
equipment to resolve the problems, ensuring safe operation of the network. This study focuses on the congestions of the MV/LV distribution transformers.

However, several burdens prevent the DSOs from adequately monitoring and managing the congestions in their LV distribution networks. For congestion monitoring and management, the DSOs have to install advanced sensors at the MV/LV power substations [6], develop the communication infrastructure between their systems and customers’ premises, and build data management systems to handle the gathered data [7]. These developments are deemed costly and cannot be easily implemented, given the sheer scale of LV networks. A recent implementation of the local energy communities (LECs), along with the increasing availability of residential smart meters (SMs), has offered a solution to the congestion monitoring and management in LV networks. The LECs intend to actively participate in RES projects in their regions [8], which can improve the controllability and energy efficiency of the local networks [9]. While the LECs can access customers’ SMs, they generally do not have information on the network topology. Therefore, if cooperating actively with the DSOs, the LECs can provide essential operational supports to the utility grids [9]. In contrast, if the collection and management of all customers’ SM data, as well as the control of customers’ RES and consumption, are performed by the DSOs, this can impose additional burden on their work, especially when considering the high volume of SMs and the wide geographical dispersal of RES [10]. All these in turn can negatively affect congestion management as well as network operation. In this regard, using data-driven approaches (i.e., leveraging the residential SM data implemented by the LECs) is considered promising while supporting the network operation and preserving customers’ rights.

While recognising added values of SM data from residential houses for LV network monitoring, most previous works focussed on physics-based approaches that limit their applicability in the diversified, complex LV networks [11–17]. In [11], a field demonstrator with residential SMs for LV network monitoring is presented. Dedicated monitoring systems, however, are still required at the substation’s secondary side. An approach based on the synchronous SM dataset is investigated in [12] to monitor LV distribution networks. In [13], the aggregated SM readings from LV end users are utilised to calculate transformer load profiles. Similarly, based on the aggregated SM data, a state estimation method for LV systems is described in [14], while a state estimation method for MV systems is discussed in [15]. The distribution system state estimation enables the DSOs to monitor their grid variables, that is voltages, currents, and powers, allowing network congestion monitoring. The authors in [16] propose an approach to process SM data in LV grids, including phase identification, data aggregation and uncertainty evaluation, to determine nodal powers in MV grids, that is LV transformer loading. However, the practical application of these methods discussed in [11–16] to monitor LV transformer loading has some significant impediments. For instance, not all customers are equipped with SMs [16], causing the summation of SM readings from the relevant customers to determine the distribution transformer loading inaccurately. Besides, the volume of SM data aggregation grows exponentially given the increasing number of customers, which likely creates difficulties in collecting and processing the data [18]. In this regard, it is desired to have a method that can use a reduced amount of SM data to monitor the congestions. The works in [17] propose methods to select representative meters for voltage monitoring in LV networks, which can be extended to realise congestion monitoring. However, a common issue with the application discussed in [17] is that active and reactive power data from residential SMs are used. Since the consumption data contains highly private information about the households, their privacy and security are subjected to violation [19]. Privacy concerns cause the customers in certain countries to raise objections against the use of SM readings. In contrast, the voltage measurements by residential SMs within the LEC context can be considered as resulting in no privacy concerns as they represent the network state. Our recent study in [20] shows that there exists a strong correlation between the nodal voltage magnitudes, that is measured by SMs, and the active power flows through distribution transformers. A limited (critical) set of these measurements leveraged by advanced machine learning (ML) techniques can be utilised to estimate transformer loading and then facilitate congestion management, for example, by controlling RES.

Among the ML techniques for power systems, regression models are commonly used because of their easy-to-implement and easy-to-interpret features [19]. Regression models mathematically formulate the correlation between one output variable and one or multiple input variables, afterwards allowing to estimate the output variable from the input variables. Regression models are widely employed for load forecasting [19, 21, 22], RES generation forecasting [23, 24], power system state estimation [25, 26], and supporting network operation [27, 28]. However, to the authors’ best knowledge, there have been no studies on regression models in transformer congestion monitoring that take into account the measurements from end-users’ SMs.

Based on the discussion presented above, this article proposes a data-driven approach implemented by the LECs to monitor transformer congestions in LV distribution networks. The proposed approach involves adopting different regression models and leveraging data measured from residential SMs. Four ML algorithms for realising regression models namely (a) Ridge Regression (RR), (b) Support Vector Regression (SVR), (c) Random Forest Regression (RFR), and (d) eXtreme Gradient Boosting Regression (XGBR) are compared to select the best one using a cross-validation (CV) approach. The motivation for adopting these ML algorithms is elaborated in the next sections. Among all the features discussed, the main contributions of this article can be summarised as follows:

- A thorough analysis based on summary statistic and graphical visualisation is implemented to discover the relationships between the transformer loading and voltage magnitudes measured at the point of connections (POC) of the houses.
- A regression-based method is introduced in combination with a limited series of residential SM data for transformer loading estimation. To this end, a data-driven monitoring
framework is provided to implement and validate the models. A comprehensive comparison with various ML algorithms is also conducted to select the best-performing regression model.

The rest of the article is organised as follows: Section 2 presents the problem definition. Section 3 elaborates on the data pre-processing procedure. Next, Section 4 describes the procedure of training and evaluating various learning algorithms for calculating transformer loading. Afterwards, the numerical results and discussion presenting the proposed method’s efficacy are given in Section 5. Finally, concluding remarks are drawn in Section 6.

2 | PROBLEM DEFINITION

To provide the background for the study, this section discusses the monitoring of transformer congestion followed by the application of residential SM data. The data-driven monitoring framework for this study is then presented.

2.1 | Transformer congestion monitoring

This study aims to estimate the distribution transformer's loading, then enabling the monitoring of the transformer congestions. The loading of the transformer is defined as the instantaneous apparent power of the transformer, $S_{\text{trans}}$, which is due to either household power consumption or reverse power flows from residential RES units [3]. This work introduces a procedure to monitor the congestions stemming from both cases. Despite the ability of the transformer to withstand loads beyond a specified limit, prolonged overloading impairs the insulation of transformer windings [29], subsequently degrading the transformer's lifetime. Congestion monitoring, therefore, is essential for a coordinated control to be activated to restrict residential consumption or curtail RES generation.

2.2 | Residential smart meter data

Beyond the consumption measurement for billing, SMs can record voltage, current, and power quality information [30]. The information can be recorded at preset time intervals (typically 15, 30, or 60 min) and with high accuracy. These advanced features allow residential SMs to be already employed for management and planning in distribution networks, such as detecting energy theft, forecasting load, and verifying network topology. A detailed analysis of SM application in distribution networks can be found in [31, 32]. For determining the transformer loading discussed in this work, the voltage magnitude measured by residential SMs is used as an input variable. For the sake of simplicity, residential SMs will be termed as SMs in the rest of this article.

2.3 | Data-driven monitoring framework

A framework is introduced in this study to realise the data-driven approach for the transformer congestion monitoring, which has been schematically presented in Figure 1. This framework includes two main building blocks as follows:

- Model preparation: a comprehensive procedure for transformer loading estimation is executed involving data pre-processing, and ML model fitting and evaluation. The main outputs of the procedure are a reduced set of SMs identified as input data sources and the best-performing ML algorithm for the regression model. Individual steps in the procedure are elaborated in Sections 3 and 4.
- Model deployment: data from the selected SMs is collected and fit into the tuned, trained model given by the model preparation stage to estimate the transformer loading. The transformer congestion, subsequently, can be identified. The model performance is tested with the results discussed in Section 5.

![Figure 1: Methodological framework. ML, machine learning](image-url)
3 | DATA PRE-PROCESSING PROCEDURE

This section explains the procedure adopted to pre-process data in the framework for calculating transformer loading depicted in Figure 1. Implementation of the procedure for a specific test case is also described.

3.1 | Training data generation

An equitable training dataset plays a decisive role in obtaining promising results when utilising any data-driven technique. In this article, to prepare a reasonable training dataset, stochastic samples are generated by solving a substantial number of power flows and logging the corresponding results. For a given LV network and PV penetration levels, the variations in household load consumption and PV production need to be considered.

3.1.1 | Simulation approach

A simulation approach to generate the stochastic training dataset is developed, which is based on the method in [33]. In the following, the main stages are briefly described.

• Firstly, for \( N \) households, a set of profiles \( P \), with size \( |P| \) of one year and a time resolution of 15 min (i.e. \( |P| = 35,040 \)), of load consumption and PV generation are built following the modelling approach presented in the next sections. Consequently, the obtained profiles are randomly distributed to the houses in the test network. For simplicity, it is assumed that PV units have the same phase connection as the houses.
• Secondly, one-year time-series power flows with size \( |P| = 35,040 \) are solved iteratively. A data sample \( (X, Y) \) is recorded at each execution. For this, \( X \) is a set of input variables, which is regarded as the SM data, including voltage magnitudes and net active and reactive power measured at the POC of all houses, that is \( V_i, P_{i\text{net}}, \) and \( Q_{i\text{net}} \) with \( Y \in N \), respectively. \( X \) is the output variable, that is the transformer loading, \( S_{\text{trans}} \).
• Thirdly, the obtained dataset is split into two parts: 80% of the dataset is used to simulate the historical data for model preparation, and the remaining dataset simulates future data for model deployment. These sets are then stored for later processing steps.

3.1.2 | Test network description

The test network is based on the IEEE European LV distribution network [34]. This three-phase, radial LV network is energised from a 250 kVA, 11/0.4 kV distribution transformer with secondary-side voltage level settings of 1.03 p.u. The remaining features are retained as in the original IEEE European LV distribution network. A geographical single-line diagram of the IEEE European LV test network is displayed in Figure 2. The test network serves as an unbalanced grid supplying 55 residential users with single-phase connections, of which there are 21 users on phase A (i.e. \( N^A = 21 \)), 19 users on phase B, and 15 users on phase C. Every household is assumed to own a PV system. The rated capacities of PV systems in the test network are randomly chosen between 4.06

**FIGURE 2** Single-line diagram of the IEEE European low-voltage test network. The numerical value denotes the number of the node where the load is connected [34]
and 6.27 kWp [33], which is derived from the information on the real residential PV installations in the Netherlands. The test network is assumed to have a LEC that can access the residential SM data and control RES at the customer premises. For the demonstration purpose, an overview of household load characteristics and installed PV capacities for phase A is shown in Table 1. Furthermore, in this study it is assumed that the topology of the test network remains constant.

### 3.1.3 Load consumption and photovoltaic generation profiles

The household load consumption used in the training data generation is modelled based on real one-year 15-min historical SM measurements in terms of active power from residential consumers in the Netherlands. To consider reactive power consumption, all loads have been assumed to have a power factor of 0.97 (inductive).

The synthetic PV generation profiles are built on a PV model, which utilises meteorological data and specified PV factor of 0.97 (inductive).

#### Table 1 Summary of load characteristics and PV installations in phase A

| House no. | Node no. | Peak load (kW) | Installed PV capacity (kWp) |
|-----------|----------|----------------|-----------------------------|
| H1        | 34       | 3.9            | 6.1                         |
| H3        | 70       | 3.1            | 4.7                         |
| H4        | 73       | 2.9            | 4.3                         |
| H5        | 74       | 2.8            | 4.7                         |
| H9        | 225      | 4.1            | 4.9                         |
| H14       | 289      | 3.6            | 4.1                         |
| H20       | 349      | 3.5            | 4.7                         |
| H21       | 387      | 3.0            | 5.0                         |
| H22       | 388      | 4.2            | 5.2                         |
| H25       | 502      | 4.8            | 5.0                         |
| H29       | 562      | 3.4            | 6.2                         |
| H30       | 563      | 3.6            | 4.1                         |
| H31       | 611      | 2.5            | 4.5                         |
| H34       | 629      | 3.4            | 4.5                         |
| H46       | 817      | 4.0            | 4.6                         |
| H48       | 860      | 2.7            | 5.5                         |
| H49       | 861      | 3.5            | 5.5                         |
| H51       | 896      | 2.8            | 4.3                         |
| H52       | 898      | 3.6            | 4.9                         |
| H54       | 900      | 4.1            | 5.0                         |
| H55       | 906      | 2.7            | 5.1                         |

Abbreviation: PV, photovoltaic.
the transformer loading on phase A ($S_{trans}^A$) and voltage magnitudes at the POC of all houses connected to the same phase ($V_i$ with $i \in \mathcal{N}^A$) in the dataset. A pair plot comprises histogram plots along the diagonal, Pearson correlation coefficients below the diagonal, and scatter plots above it [21], which constitute a matrix of relationship between each variable. The resulting Pearson correlation coefficients illustrate the extent of linear dependence either between $S_{trans}^A$ and $V_i$ with $i \in \mathcal{N}^A$ or among $V_i$ with $i \in \mathcal{N}^A$, which are calculated utilizing Equation (2),

$$R(\nu, \mu) = \frac{E[(\nu - \mu_\nu)(\mu - \mu_\mu)]}{\sigma_\nu \sigma_\mu}$$

where $R$ denotes the Pearson correlation coefficient between two variables $\nu$ and $\mu$, and $E$ represents the expectation with standard deviations $\sigma_\nu$ and $\sigma_\mu$ and means $\mu_\nu$ and $\mu_\mu$. As can be seen from the subplots below the diagonal in Figure 3 this section and Figures A1 and A2, the resulting coefficients $R$ between $S_{trans}^A$ and $V_i$ with $i \in \mathcal{N}^A$ have significant positive values above 0.9, indicating strong correlation between them. It means that increasing voltage magnitudes correlate with higher transformer loading. The voltage levels, therefore, can serve as explanatory variables for calculating transformer loading (i.e. response variable).

The relationship between $S_{trans}^A$ and $V_i$ with $i \in \mathcal{N}^A$ can also be validated by the computed scatter plots in Figure 3 in
this section and Figures A1 and A2. In these figures, pairs of numerical data are plotted to visualise the relationship of the variables. Interestingly, the scatter plots indicate clear V-shaped patterns, highlighting a potential for applying regression models to compute power flow through the transformer from the nodal voltage magnitudes. Thus, the statistical and graphical analysis prove that combining voltage measurements from SMs within a particular LEC and regression models provides a possible solution to the transformer loading estimation.

It is important to mention that, although net (active and reactive) power measurements are also available from household SMs, using such measurements to estimate transformer loading will violate the privacy rights of customers. Hence, active and reactive power data from SMs are deemed unsuitable input candidates for data-driven monitoring of transformer loading.

### 3.3 Feature Selection

Since the dataset contains a high number of features (variables), feature selection (variable elimination) is of importance to select a subset of relevant features directly from the original set [40]. The reason for this is that some features are irrelevant, redundant, or noisy for model construction, which likely degrades the performance of the model [41]. Additionally, high-dimensional data results in more computational burden and increased execution time for model training, thus reducing computational efficiency [42]. Feature selection aims to reduce the dimensionality of the data, subsequently decreasing computational requirements, accelerating the learning process, and enhancing the model performance [41, 42].

Feature selection methods can be categorised as supervised, semi-supervised and unsupervised according to the label information [40]. Solving regression problems generally involves applying supervised feature selection that measures the importance of an individual feature via its correlation with the regression output [43]. On the other hand, methods of feature selection can be broadly classified into three groups: filter, wrapper and embedded according to the searching strategy [40]. Because it has the merits of both filter and wrapper methods, the embedded method is preferable to realise feature selection [43]. The embedded method leverages the inherent structure of a learning algorithm to incorporate feature selection into the learning process [43]. Among available models for embedded method-based feature selection, regularisation models, for example least absolute shrinkage and selection operator (LASSO), are most widely utilised [43].

In this article, the LASSO approach is adopted to perform supervised feature selection on the training dataset. This adoption allows preventing bias in comparing ML algorithms used for the regression problem (as discussed in the next steps) as the LASSO model is not in the list of the ML algorithms to be compared. The importance values of a single feature (i.e. $V_i$ with $V_i \in \mathcal{N}^A$) to the regression output (i.e. $S_{\text{trans}}$) calculated on the training dataset by using LASSO is illustrated in Figure 4. It is observed that the houses are ranked in order of importance with their values varying largely. Several houses have small or even zero importance values, while others have high values. This is derived from the working principle of LASSO that minimises the fitting errors whilst shrinking the magnitude of feature coefficients [43]. An importance threshold $\omega$ is then required to define the relevant features with their importance values greater than the threshold. Therefore, with $\omega = 2.5$, the voltage levels at the POC of five houses, including H3, H1, H31, H29, and H48, are selected from a total of 21 houses on phase A to be the explanatory variables of the regression model. Choosing this value of $\omega$ for feature selection has no loss of generality because the main focus of this study is to compare the relative performance of various ML algorithms. The feature selection based on LASSO is conducted separately in a task before the training process; thus, the same subset of features will be selected for every algorithm fitting. Using the same feature subset (i.e. same inputs) for the algorithms, along with tuning its hyperparameters and comparing the generalisation and stability of its performance (as discussed in the next section), facilitates the fair comparison of the algorithms. In addition, selecting the feature based on this $\omega$ value is adequate to exemplify the model fitting and validation procedure. Hence, the dataset corresponding to the selected features can be formed from the training set for constructing the regression model in a later stage.

### 4 Machine Learning Model Fitting and Evaluation Procedure

Given the provided input data, this section focusses on the procedure of training and evaluating several ML algorithms for calculating transformer loading, as depicted in Figure 1. It is
important to note that properly evaluating and comparing different ML algorithms necessitate addressing two matters. Firstly, the hyperparameters of the algorithm should be tuned before the evaluation stages as their values will highly impact the algorithm performance. Secondly, the comparison of various algorithms should be based on generalisation [7] and the stability of their estimated performance [44]. In this section, these two essential concerns are thoroughly considered using nested CV, aiming to facilitate the fair comparison between the studied ML algorithms. The hyperparameter tuning task is presented in Section 4.2, and the algorithm-comparing task is elaborated in Section 4.3.

4.1 | Machine learning algorithm list preparation

As there are several different ML algorithms for the regression problem, this section aims to prepare a list of various ML algorithms selected to compare their performance for transformer loading estimation. There are four ML algorithms added in the list as follows:

- **Ridge Regression**: This regression model can be formulated in matrix form as [45],

\[
\hat{\beta}(\alpha) = (X^T X + \alpha I)^{-1} X^T Y
\]  

where \(\hat{\beta}(\alpha)\) is the RR estimator, \(\alpha\) is the penalty parameter, \(X\) and \(Y\) are explanatory and response variables, respectively, and \(I\) is the identity matrix.

- **Support Vector Regression**: Support Vector Regression attempts to find the coefficients that minimise the function given by [46].

\[
\varepsilon \sum_{i=1}^{N} (\lambda_i^+ + \lambda_i^-) - \sum_{i=1}^{N} (\lambda_i^+ - \lambda_i^-) y_i \\
+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-) \kappa(x_i, x_j)
\]  

subject to

\[
\sum_{i=1}^{N} (\lambda_i^+ - \lambda_i^-) = 0 \quad \forall \lambda_i^+, \lambda_i \in [0, C]
\]

while the function approximation for SVR is written as

\[
\hat{f}(x) = \sum_{i=1}^{N} (\lambda_i^+ - \lambda_i^-) \kappa(x_i, x)
\]

where \(\varepsilon\) denotes the epsilon parameter, that is the tolerance for error, \(N\) denotes the size of data, \(\lambda_i, \lambda_i^+, \lambda_i^-\) denote the Lagrange multipliers with non-negative values, \(\kappa(\ldots)\) denotes the kernel coefficient, \(C\) denotes the regularisation strength, and \(x\) and \(y\) denote explanatory and response variables, respectively.

- **Random Forest Regression**: As a tree-based ensemble algorithm, the function approximation of RFR (i.e. a collection of \(Tr\) randomised regression trees) for a training sample \(D_N = (X, Y)\) can be formulated as [47],

\[
\hat{f}_{Tr,N}(x, \Theta_1, \ldots, \Theta_{Tr}) = \frac{1}{Tr} \sum_{i=1}^{Tr} f_N(x, \Theta_i)
\]  

where \(Tr\) is the number of randomised regression trees in the forest, \(N\) is the size of data, \(\Theta_1, \ldots, \Theta_{Tr}\) are random variables (i.e. independent of the sample \(D_N\)), \(f_N(x, \Theta_i)\) denotes the predicted value of point \(x\) for a single \(i\)-th tree (i.e. constructed using the bootstrapped samples [48]), and \(X\) and \(Y\) are explanatory and response variables, respectively.

- **eXtreme Gradient Boosting Regression**: eXtreme Gradient Boosting Regression is an ensemble tree algorithm with the aim of minimising the objective function given as [49].

\[
\sum_{i=1}^{N} l(\hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)
\]  

where \(N\) is the size of data, \(\hat{y}_i^{(t-1)}\) denotes the prediction output at the \((t-1)\)-th (i.e. previous) iteration of the \(i\)-th instance, \(y_i\) denotes the observation, \(f_t\) denotes the independent tree structure at the \(t\)-th iteration, \(l\) denotes the loss function measuring the difference between \(y_i, \hat{y}_i^{(t-1)}\), and \(\Omega(f_t)\) denotes the penalty term for the model complexity. The term \(\Omega(f)\) can be calculated as

\[
\Omega(f) = \gamma L + \frac{1}{2} \alpha \|\omega\|^2
\]  

where \(L\) is the number of leaves in the tree, \(\omega\) is the score of leaves, and \(\gamma\) and \(\alpha\) are regularisation coefficients [50].

The motivation behind the preliminary use of these four algorithms is the capability to handle the non-linear regression (LR) problem and continuous variables, that is by using SVR [22] and RFR [51]. Besides, the algorithm's capability for analysing data suffering from the multicollinearity problem, which refers to the linear relationship among explanatory variables, is considered, that is by using RFR [51] and RR [52]. This is derived from the fact that non-linearity and multicollinearity exist in the obtained training data, as can be seen in Figures 3, A1 and A2. Additionally, the algorithm having state-of-the-art advantages in ML research and recently being increasingly used in power engineering is included, that is XGBR [27]. Simulation setup of the listed ML algorithms for the next steps is implemented using the Scikit-learn package in Python [53]. For this, the principles of the ML algorithms represented by Equations (3)–(9) are considered. This set-up process is discussed in the next section.
4.2 Model evaluation and selection

Model evaluation aims to estimate how well the performance of the specified model generalises to unseen data, the so called generalisation error [44]. To this end, it is required that model fitting and evaluation are executed on the independent datasets, that is training and validation data. Otherwise, the performance estimation can be biased due to overfitting [44].

Model selection refers to a process of tuning the hyperparameters of a specific ML algorithm to the best settings corresponding to the best-performing model [54]. A ML algorithm is generally designed with several hyperparameters; those are tuning parameters with adjustable values and have decisive effects on the performance of the algorithm [44]. Given a specific ML approach, applying different hyperparameters will lead to different models. Thus, it is desirable to select the model with the best estimation accuracy by tweaking the optimal hyperparameters [44].

To perform model evaluation and selection (as well as the algorithm evaluation and selection), nested CV is well-suited as it demonstrates several advantages [44]. This advanced method is a nesting of two $k$-fold CV loops: the inner and the outer CV loop, which form a $k \times k$ set up. The $k$-fold CV of the outer loop splits the entire historical dataset into $k$ random, non-overlapping subsets: one subset serves as the testing data for the algorithm selection and the remaining $(k - 1)$ subsets are merged to form the training data used in the inner loop [55]. The inner loop based on the $k$-fold CV implements model evaluation and selection by crossing over the training and validation steps in an iterative manner, that is over the original training data derived from the outer loop $k$ times. For model evaluation purpose, the $k$-fold CV of the inner loop further splits the provided training data into $k$ random, independent subsets. The $k$ subsets are then interchangeably used so that one subset serves as the validation data and the remaining $(k - 1)$ subsets are merged and serve as the training data [55]. This split prevents the leaking of test data during the training process, thus avoiding overfitting and reducing the bias of the models’ performance estimates [54]. For model selection purpose, the inner loop runs a specific ML algorithm with each fixed hyperparameter configuration over the training data $k$ times in a loop. Each iteration of the loop uses a distinct subset of training and validation data; therefore, when that loop finishes, each hyperparameter configuration is associated with $k$ different models and $k$ different performance estimates [44]. Later, the multiple resulting estimates of every model are averaged to produce the ultimate generalisation error, ranked against each other to select the best model, that is with the lowest error. This loop is essential because the algorithm cannot learn the hyperparameter settings by itself during the fitting process, meaning these settings need to be specified a priori and then evaluated separately [44]. More details on the nested CV method can be found in [44, 54].

This study involves adopting nested CV with a $10 \times 10$ set up for the model evaluation and selection for the ML algorithm list (prepared in Section 4.1). The $k$-fold CV outer loop with $k = 10$ partitions the whole dataset into 10 parts utilised for the inner loop. For an algorithm, a range of settings is initially specified as the bounded domain of hyperparameters. Then, from that domain, a set $H$, with size $|H| = 15$, of the combinations of hyperparameter settings are randomly sampled using randomised search technique [44]. To facilitate the random sampling, the hyperparameters’ settings are defined as the uniform distributions of values, as shown in Table 2. The $k$-fold CV

| ML algorithms | Hyperparameters | Bounded domain               |
|---------------|-----------------|------------------------------|
| RR            | Maximum number of iterations ($T_{max}$), i.e. an integer | Uniform distribution over [1000, 10000] |
|               | Regularisation strength ($\alpha$), i.e. a float       | Log-uniform distribution over $[10^{-5}, 100]$ |
| SVR           | Kernel coefficient ($\kappa$), i.e. a float            | Log-uniform distribution over [1, 200] |
|               | Regularisation strength ($C$), i.e. a float            | Log-uniform distribution over $[10^{-5}, 100]$ |
|               | Epsilon parameter ($\varepsilon$), i.e. a float        | Log-uniform distribution over $[10^{-5}, 100]$ |
|               | Maximum number of iterations ($T_{max}$), i.e. an integer | Uniform distribution over [1000, 10000] |
| RFR           | Number of trees in the forest ($T_r$), i.e. an integer  | Uniform distribution over [100, 1000] |
|               | Maximum depth of the tree ($D_{max}$), i.e. an integer  | Uniform distribution over [10, 500] |
|               | Minimum number of samples required to split an internal node ($S_{max}$), i.e. an integer | Uniform distribution over [1, 20] |
|               | Minimum number of samples required to be at a leaf node ($L_{min}$), i.e. an integer | Uniform distribution over [1, 20] |
| XGBR          | Subsample ratio of columns when constructing each tree ($\theta$), i.e. a float | Uniform distribution over [0, 1] |
|               | Regularisation strength ($\alpha$), i.e. a float        | Log-uniform distribution over $[10^{-5}, 100]$ |
|               | Parameter to shrink the contribution of each tree ($P_r$), i.e. a float | Uniform distribution over [0, 1] |
|               | Maximum depth of the individual regression estimators ($D_{max}$), i.e. an integer | Uniform distribution over [100, 1000] |
|               | Number of gradient boosted trees ($T_r$), i.e. an integer | Uniform distribution over [100, 1000] |

Abbreviations: ML, machine learning; RFR, random forest regression; RR, ridge regression; SVR, support vector regression; XGBR, eXtreme gradient boosting regression.
inner loop with \( k = 10 \) in the initiated nested CV is later applied for each combination of hyperparameter values \( h \in |H| \). The goal is to find the hyperparameter combination that achieves the lowest generalisation error among all the investigated combinations in \( |H| \). For this, a set \( S \) with \( |S| = 10 \) of various subsets of training and validation data is generated. In total, 6000 simulations are required for a list of four ML algorithms. A performance metric used for the model evaluation and selection is root mean square error (RMSE), which quantifies the average magnitude of the error between the real values and the computed values, as calculated by [56],

\[
RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (y_t - \hat{y}_t)^2}
\]  

where \( y \) and \( \hat{y} \) denote the real and computed values, respectively, \( T \) denotes the total number of steps of the time-series validation data, and \( t \) denotes the time step in \( T \). The best hyperparameters are chosen with the lowest RMSE. The process can be schematically presented, as shown in Figure 5. For the demonstration purpose, the numerical results in terms of the optimal hyperparameter configurations associated with the averaged RMSE as a generalisation error for the SVR algorithm are summarised in Table 3. It is worth mentioning that since the inner loop operates over 10 distinct subsets of the simulated historical data, which are formed on account of the outer loop, 10 combinations of the best hyperparameters along with their estimated performance are also achieved. It is observed that the hyperparameters are properly customised with varying levels of settings among the subsets. With the optimal hyperparameters from the inner loop, the algorithm will be trained and evaluated in the outer loop, which is presented in the next section.

### 4.3 Machine learning algorithm selection

The ML algorithm selection involves finding the algorithm that has the lowest generalisation error and the best stability capability among all the compared algorithms. The generalisation error represents the performance estimates of the algorithm in new, unseen data, that is not in the training set. Meanwhile, the stability capability quantifies how well the algorithm performs on multiple different test datasets [44]. The \( k \)-fold CV enables both estimation of the generalisation performance and testing of the stability of the algorithm. The rationale is that this method retains the independent test dataset for the performance estimate, and it iterates the performance testing \( k \) times in different datasets. These features make the \( k \)-fold CV a common approach to complete the algorithm selection [44].

Owing to the \( k \)-fold CV in the outer loop, nested CV is applicable to the ML algorithm selection. Accordingly, nested CV with a \( 10 \times 10 \) set up established in Section 4.2 is employed in this section. Specifically, in every iteration \( k \) of the outer loops, every algorithm fits the model along with its optimal hyperparameters determined from the inner loop on the training data and then evaluates the model performance on the withholding test data. The performance metrics are based on RMSE using Equation (10). Similar to the inner loop, 10 resulting generalisation errors of each algorithm are averaged.

---

**Table 3** Summarised results for model evaluation and selection of SVR

| Data subset | Optimal hyperparameters | \( IT_{\text{max}} \) | Mean RMSE |
|-------------|-------------------------|-----------------|-----------|
| 1           | 6.44 30.57 1.71         | 9122            | 3.04      |
| 2           | 15.7  1.16  4.44        | 3440            | 3.66      |
| 3           | 6.92  6.26  \( 1.7 \times 10^{-4} \) | 6959        | 3.77      |
| 4           | 54.5  0.006  0.74       | 7879            | 3.2       |
| 5           | 5.26  38.08 0.37        | 3976            | 3.72      |
| 6           | 65.8  0.28  0.66        | 3599            | 2.92      |
| 7           | 95.4  2.02  5.86        | 2571            | 2.84      |
| 8           | 79.6  84.6  4.05        | 9608            | 1.95      |
| 9           | 38.1  63.9  0.84        | 8015            | 2.48      |
| 10          | 7.07  1.32  \( 3.3 \times 10^{-4} \) | 5547        | 4.87      |

**Abbreviations:** RMSE, root mean square error; SVR, support vector regression.
The RMSE results across each fold of the outer loop for individual ML algorithms are displayed in the box plot, as shown in Figure 6. The numerical values of RMSE and the computational time are listed in Table 4. As can be observed from both the figure and the table that, RFR and XGBR clearly outperform SVR and RR as the first two algorithms have lower mean RMSE values from around 3.5 to 4.5 times than that of the last two algorithms. Moreover, it can be seen from the figure that the error spreading of RFR and XGBR significantly decreases compared to SVR and RR. Between RFR and XGBR, the RMSE results are relatively comparable as both these models are based on the ensemble tree algorithm. In view of the computational time, SVR has the longest time, followed by RFR and XGBR, and RR is the fastest algorithm. This is due to the fact that the SVR model is built by solving a quadratic programming problem and selecting the best support vectors from the dataset, which is typically computationally expensive [57]. RFR consists of a large number of regression trees in which each tree is grown by drawing a set of bootstrapped samples. This makes the training of RFR relatively time-consuming, nevertheless faster than SVR as a result of growing the trees in parallel [48]. XGBR has significantly lower computational time than RFR because XGBR adopts an improved tree learning algorithm (for managing sparse data) and a quantile sketch scheme (for considering weights) [49]. By only solving the linear equations and sorting them, RR requires the lowest simulation time [58] but at the expense of significantly reduced accuracy. Hence, these findings suggest that RFR and XGBR algorithms are of worth for further study on the defined regression problem.

5 | MODEL DEPLOYMENT FOR TRANSFORMER LOADING ESTIMATION

Given the ML algorithms selected from the previous section, this section focuses on the procedure of deploying the algorithms to estimate the transformer loading, as depicted in Figure 1. The evaluation of the algorithm performance is also analysed. It is worth mentioning that the algorithm evaluation completed in Section 4.3 involves the small sets of test data, and only RMSE served as the performance metric. Therefore, it is desirable to carry out further tests on a larger dataset and several performance metrics for the selected algorithms.

5.1 | Model set up

The model set-up stage prepares the input data for ML algorithms and implements the model evaluation and selection. Firstly, the input data preparation requires gathering the voltage measurements of the selected household SMs as specified in Sections 3.2 and 3.3 from the operational network data. In this study, the operational data during the model deployment is simulated by using 20% of the stochastic training dataset withheld from the data generation in Section 3.1.1. As a result, a set of 7008 time-series voltage measurements with 15-min time resolution at the POC of five houses, including H3, H1, H31, H29, and H48, are obtained. Secondly, implementing the model evaluation and selection adopts the k-fold CV with \( k = 10 \) that is similar to the inner loops of nested CV, as explained in Section 4.2 to optimise the model hyperparameters. To this end, the randomised search technique is also utilised to randomly sample a set \( H \) with size \( |H| = 15 \), of the hyperparameter combinations from the predefined ranges listed in Table 2.

A difference from Section 4.2, however, is that the model selection process in this section excludes the \( k \)-fold CV outer loop with \( k = 10 \) of nested CV. It means that the model hyperparameters are tuned based on the entire set of the simulated historical data at once instead of being iterated several times in various folds. A total of 300 simulations, thus,
are required for two ML algorithms of RFR and XGBR. The results of optimal hyperparameters, along with the estimated model performance based on RMSE, are illustrated in Table 5. With these optimal hyperparameters, the RFR and XGBR models are trained on the entire historical data for the later performance estimates.

5.2 | Model performance analysis

After being trained with the optimised hyperparameters, the RFR and XGBR algorithms use the gathered input data of household voltage measurements to calculate the transformer loading. With an aim to analyse the model performance, three performance metrics are computed. First, by using RMSE as determined by Equation (2), the quantitative insights into the generalisation performance of the model are obtained [56]. Secondly, Pearson correlation coefficients $R$, as defined by Equation (10), are used to gain insights into the degree of linear dependence between the real value and the estimated value. Thirdly, the Kolmogorov–Smirnov (KS) statistical test is adopted in this study because it sufficiently operates with non-normally distributed data. The non-normally distributed data of $V_i$ with $\forall i \in N^4$ is shown in the histogram plots in Figures 3, A1 and A2. In the KS statistical test, the $K$ statistic indicates how close the proximity of the compared distributions of data will be. Hence, a lower $K$ statistic will be expected for higher estimation accuracy. Moreover, for a comparison of the RFR and XGBR models to non-ML models, the ordinary least squares LR model is applied as a benchmark for transformer loading calculation using the same input data and performance metrics.

The results of the three performance metrics are listed in Table 6. As can be seen, the ML models (i.e. RFR and XGBR) significantly outperform the non-ML model (i.e. LR). The ML models have the coefficients $R$ and RMSE around 5.2 and 7.8 times, respectively, higher than the non-ML model, while the $K$ statistic from the KS test is around 3.4 times lower. Owing to the features of the tree-based ensemble algorithm, RFR and XGBR are comparable in the estimation accuracy as their differences in the calculated three performance metrics are insignificant. These results show the efficacy of the regression-based methods for estimating the transformer loading.

Furthermore, the probability density function (PDF) based on kernel density estimate is adopted to visualise the performance of regression models based on RFR, XGBR, and LR algorithms on the transformer loading calculation. A comparison of the PDF of the database of transformer loading measurements (i.e. true values) and the values calculated by RFR, XGBR, and LR models is shown in Figure 7. Further, a comparison of the PDF of the errors between the measured and calculated values is demonstrated in Figure 8.

It is observed from Figure 7 that the RFR and XGBR profiles have similar patterns and statistical properties (i.e. the peak of the tail of the kernel density estimate) with the true data. In contrast, LR profiles show a large difference with the true data. As can be noticed from Figure 8, RFR and XGBR profiles have comparable statistical properties, for example the peak at 0 kVA and the short tails between approximately $-9$ and $5$ kVA. All these indicate that regression modes based on RFR and XGBR appear equally effective in accurately estimating the transformer loading when using the voltage measurements from SMs. The main driver of these similarities is

### Table 5

| ML algorithms | Hyperparameters | Optimal values | Mean RMSE |
|---------------|-----------------|----------------|-----------|
| RFR           | $T_r$           | 955            | 0.96      |
|               | $D_{max}$       | 357            |           |
|               | $S_{min}$       | 12             |           |
|               | $L_{min}$       | 2              |           |
| XGBR          | $\eta$         | 0.73           | 0.97      |
|               | $\alpha$       | 1.75           |           |
|               | $p$             | 0.8            |           |
|               | $D_{max}$       | 158            |           |
|               | $T_r$           | 574            |           |

Abbreviations: ML, machine learning; RFR, random forest regression; RMSE, root mean square error; XGBR, eXtreme gradient boosting regression.

### Table 6

| ML algorithms | RMSE   | Coefficients $R$ | $K$ statistic from KS test |
|---------------|--------|------------------|---------------------------|
| RFR           | 0.8782 | 0.9828           | 0.0499                    |
| XGBR          | 0.8767 | 0.9829           | 0.0515                    |
| LR            | 6.8771 | 0.1884           | 0.1752                    |

Abbreviations: KS, Kolmogorov–Smirnov; LR, linear regression; ML, machine learning; RFR, random forest regression; RMSE, root mean square error; XGBR, eXtreme gradient boosting regression.

![Figure 7](image-url)
that both RFR and XGB originated from the ensemble tree algorithm.

With an aim to further compare these two models, in addition to the computation time during the model fitting and evaluation, the computation time during the deployment phase is measured and displayed in Table 7. In this phase, the computation time is for re-tuning the model parameters and re-evaluating the model performance (as explained in Section 5.1). This table points out that the XGB-based regression model runs considerably faster than the RFR one as the former takes only one-third of the latter’s simulation time. Hence, considering the high effectiveness and lower execution time, it is evident that the XGB algorithm provides the best solution to the regression problem of transformer loading calculation.

6 | CONCLUSION

This article presented an innovative alternative to implementing transformer congestion monitoring in LV distribution networks by developing a data-driven approach implemented by the LECs. The proposed approach adopts the different regression models and leveraging SM data from residential customers. A thorough analysis is conducted to discover the relationships between the transformer loading (i.e. instantaneous apparent power) and residential SM readings (active and reactive power, and voltage magnitude). A comprehensive framework is provided to implement, validate and compare four different ML algorithms for the transformer loading estimation from the LECs’ perspective. The compared algorithms include RR, SVR, RFR, and XGB. The presented procedure covers all the essential stages in deploying a data-driven technique, consisting of training data generation, exploratory data analysis, feature selection, model selection and algorithm selection. In this respect, state-of-the-art advances in ML research have been integrated, such as the XGB algorithm and nested CV technique.

The obtained simulation results indicate that combining the regression models with voltage magnitude measurements from residential SMs can effectively estimate transformer loading, that is with the Pearson correlation coefficient $R$ and RMSE calculated for the real values and the estimated values of around 0.98 and 0.87, respectively, by using only a reduced set of SM data (5 out of 21 SMs used). Simultaneously, the proposed approach can preserve customers' privacy rights. Among four ML algorithms examined, XGB appears to be the best method as it achieves adequate accuracy on estimating the transformer loading but spends significantly less time for the execution (e.g., in total, one-third of the execution time of RFR). Hence, the introduced approach provides a solution to implementing transformer congestion monitoring in LV distribution networks. The approach can eliminate the need for sensors and communications dedicated to transformer congestion management due to the uses of available residential SMs. Furthermore, the approach enables the congestion monitoring capability of the LECs. Thus, the LECs can control their consumption or generation profiles to solve the congestion problems, enhancing their self-management in the regional network for safe operation.

Although successful, the data-driven approach comes with its own limitations. Appplying the proposed approach hinges on the existence of the LECs and their ability to access the residential SMs. The LECs will use the SM readings as historical and real-time operational data for the regression model's preparation and deployment. In this respect, modifications of the existing regulatory framework about the LECs’ business model and the accessibility to the residential SMs will be required. In addition, the proposed approach is not adaptive to grid topology changes or increasing PV installation, which is inherent in grid operation and evolution. These changes will induce variations in the correlation between the transformer power and nodal voltage levels, which needs to be re-captured by the regression model. Periodically assessing the regression model performance and re-training the model afterwards are essential.

In this research, a typical LV distribution network with a radial configuration is considered. For LV distribution networks with mesh topology, the application of the proposed approach can face challenges. The relationships between the transformer power and the nodal voltage magnitude can be complicated for regression models. Thus, further research is required to consider the mesh networks more efficiently.

ACKNOWLEDGEMENT

None.
CONFLICT OF INTEREST
The corresponding author or any of the co-authors has no conflict of interest to disclose.

PERMISSION TO REPRODUCE MATERIALS FROM OTHER SOURCES
None.

DATA AVAILABILITY STATEMENT
Data available on request from the authors.

ORCID
Tam T. Mai https://orcid.org/0000-0002-5317-9572

REFERENCES
1. Mai, TT, et al.: Coordinated active and reactive power control for overvoltage mitigation in physical LV microgrids. J. Eng. 2019(18), 5007–5011 (2019)
2. Fonsen, R., et al.: Evaluating flexibility values for congestion management in distribution networks within Dutch pilots. J. Eng. 2019(18), 5158–5162 (2019)
3. Haque, A.N.M.M., et al.: Agent-based unified approach for thermal and voltage constraint management in LV distribution network. Elec. Power Syst. Res. 143, 462–473 (2017)
4. Veldman, E.: Power play impacts of flexibility in future residential electricity demand on distribution network utilisation. Thesis, Technische Universiteit Eindhoven (2013)
5. LienRetheler, I., et al.: Optimised curtailment of distributed generators for the provision of congestion management services considering discrete controllability. IET Gener., Transm. Distrib. 14(5), 735–744 (2020)
6. Kulmala, A., et al.: Hierarchical and distributed control concept for distribution network congestion management. IET Gener., Transm. Distrib. 11(3), 665–675 (2017)
7. Ferdowski, M., et al.: A scalable data-driven monitoring approach for distribution systems. IEEE Trans. Instrum. Meas. 64(5), 1292–1305 (2015)
8. European Commission: Directive (EU) 2018/2001 of the European Parliament and of the Council on the promotion of the use of energy from renewable sources. Off. J. Eur. Union 2018(328), 82–209 (2018)
9. Koistala, B.P., et al.: Energetic communities for community energy: a review of key issues and trends shaping integrated community energy systems. Renew. Sustain. Energy Rev. 36, 722–744 (2016)
10. Minniti, S., et al.: Local markets for flexibility trading: key stages. Energies. 11(11), 1–21 (2018)
11. Barbato, A., et al.: Lessons learnt from real-time monitoring of the low voltage distribution network. Sustain. Energy, Grids Netw. 15, 76–85 (2018)
12. Khan, M.A., Hayes, B.: PTP-based time synchronisation of smart meter data for state estimation in power distribution networks. IET Smart Grid. 3(5), 705–712 (2020)
13. Lo, Y.L., Huang, S.C., Lu, C.N.: Transformational benefits of AMI data in transformer load modeling and management. IEEE Trans. Power Deliv. 29(2), 742–750 (2014)
14. Pau, M., et al.: Low voltage system state estimation based on smart metering infrastructure. In: IEEE International Workshop on Applied Measurements for Power Systems (AMPS 2016) Aachen (2016)
15. Kumar, C.S., et al.: Robust distribution system state estimation with hybrid measurements. IET Gener., Transm. Distrib. 14(16), 3250–3259 (2020)
16. Ni, Y., et al.: Three-phase state estimation in the medium-voltage network with aggregated smart meter data. J. Electr. Power Energy Syst. 98, 463–473 (2018)
17. Nairn, K.: Smart meter measurement-based state estimation for monitoring of low-voltage distribution grids. Energies. 13, 5367 (2020)
18. Lu, S., et al.: Real-time low voltage network monitoring–ICT architecture and field test experience. IEEE Trans. Smart Grid. 6(4), 2002–2012 (2015)
19. Yildiz, B., et al.: Recent advances in the analysis of residential electricity consumption and applications of smart meter data. Appl. Energy. 208, 402–427 (2017)
20. Mai, T.T., et al.: Stochastic modelling of the correlation between transformer loading and distributed energy resources in LV distribution networks. In: International Conference and Exhibition on Electricity Distribution (CIRED), pp. 1–5. Berlin (2020)
21. Gandanedo, L.M., Feldheim, V., Deramats, D.: Data driven prediction models of energy use of appliances in a low-energy house. Energy Build. 140, 81–97 (2017)
22. Jain, R.K., et al.: Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl. Energy. 123, 168–178 (2014)
23. Yampulskaluk, N., et al.: Condition monitoring of wind power system with nonparametric regression analysis. IEEE Trans. Energy Convers. 29(2), 288–299 (2014)
24. Zafirakis, D., Tzanes, G., Kaldellisa, J.K.: Forecasting of wind power generation with the use of artificial neural networks and support vector regression models. Energy Procedia. 159, 509–514 (2019). https://doi.org/10.1016/j.egypro.2018.12.007
25. Hassanzadeh, M., Erenosoglu, C.Y.: A regression analysis based state transition model for power system dynamic state estimation. In: NAPS 2011–43rd North American Power Symposium. Boston (2011)
26. Tran, M.Q., et al.: Multi-area distribution system state estimation using decentralized physics-aware neural networks. Energies. 14, 3025 (2021)
27. Wang, S., Dong, P., Tian, Y.: A novel method of statistical line loss estimation for distribution feeders based on feeder cluster and modified XGBoost. Energies. 10, 2067 (2017)
28. Suganayadevi, M.V., Babulal, C.K., Kalyani, S.: Assessment of voltage stability margin by comparing various support vector regression models. Soft Comput. 20, 807–818 (2016). https://doi.org/10.1007/s00500-014-1544-x
29. Haque, A.N.M.M., et al.: Demand response for real-time congestion management incorporating dynamic thermal overloading cost. Sustain. Energy, Grids Netw. 10, 65–74 (2017)
30. Alahakoon, D., Yu, X.: Smart electricity meter data intelligence for future energy systems: a survey. IEEE Trans. Industr. Informat. 12(1), 425–436 (2016)
31. Farokhi, F.: Review of results on smart-meter privacy by data manipulation, demand shaping, and load scheduling. IET Smart Grid. 3(5), 605–613 (2020)
32. Wang, Y., et al.: Review of smart meter data analytics: applications, methodologies, and challenges. IEEE Trans. Smart Grid. 10(3), 3125–3148 (2019)
33. Vergara, P.P., et al.: A comprehensive assessment of PV inverters operating with droop control for overvoltage mitigation in LV distribution networks. Renew. Energy. 159, 172–183 (2020)
34. Ni, F., Nguyen, P.H., Cobben, J.F.G.: Basis adaptive sparse polynomial chaos expansion for probabilistic power flow. IEEE Trans. Power Syst. 32(1), 694–704 (2017)
35. Dobos, A.: PVWatts Version 5 Manual–Technical Report NREL/TP-6A20-62641. National Renewable Energy Lab (NREL), Golden (2014)
36. Royal Netherlands Meteorological Institute (KNMI): Hourly weather data in the Netherlands. http://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi (2020). Accessed 2 May 2020
37. Ferdowski, M., et al.: Measurement selection for data-driven monitoring of distribution systems. IEEE Trans. Syst. J. 13(4), 4260–4268 (2019)
38. Morgenthaler, S.: Exploratory data analysis. Wiley Interscience. Rev. Comput. Stat. 1(1), 33–44 (2009)
39. Selman, H.J.: Chapter 4 exploratory data analysis. In: Experimental Design and Analysis, pp. 61–100. Carnegie Mellon University, Pittsburgh (2018)
40. Miao, J., Nia, L.: A survey on feature selection. Procedia Comput. Sci. 91, 919–926 (2016)
41. Xue, B., et al.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. 20(4), 606–626 (2016)
42. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. 40(1), 16–28 (2014)
43. Li, J., et al.: Feature selection: a data perspective. ACM Comput. Surv. 50(6), 163–193 (2017)
44. Raschka, S.: Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. arXiv (2018)
45. McDonald, G.C.: Ridge regression. Wiley Interdiscip. Rev. Comput. Stat. 1(1), 93–100 (2009)
46. Hawkins, J.: Support vector regression. In: Efficient Learning Machines, pp. 67–80. Apress, Berkeley (2015)
47. Sebrenik, E.: Random forests and kernel methods. IEEE Trans. Inf. Theor. 62(3), 1485–1500 (2016)
48. Ahmad, M.W., Reynolds, J., Rezgui, Y.: Predictive modelling for solar thermal energy systems: a comparison of support vector regression, random forest, extra trees and regression trees. J. Clean. Prod. 203, 810–821 (2018). https://doi.org/10.1016/j.jclepro.2018.08.207
49. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. San Francisco (2016)
50. Lu, H., et al.: Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: a case study of an intake tower. Energy. 203, 117756 (2020). https://doi.org/10.1016/j.energy.2020.117756
51. James, G., et al.: An Introduction to Statistical Learning With Applications in R. Springer (New York, USA 2013)
52. Li, G., Niu, P.: An enhanced extreme learning machine based on ridge regression for regression. Neural Comput. Appl. 22, 803–810 (2013)
53. Kramer, O.: Machine Learning for Evolution Strategies, vol. 20. Springer International Publishing, Switzerland (2016). https://doi.org/10.1007/978-3-319-33383-0_2
54. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, New York (2008)
55. Vehtari, A., Gelman, A., Gabry, J.: Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat. Comput. 27, 1413–1432 (2017)
56. Mocanu, E., et al.: Deep learning for estimating building energy consumption. Sustain. Energy, Grids Netw. 6, 91–99 (2016)
57. Maimon, O., Rokach, L., Edition, S.: Data Mining and Knowledge Discovery, Second edn. Springer, Boston (2016)
58. Vapnik, V.N.: Kernel ridge regression. In: Empirical Inference, pp. 105–116. Springer, Berlin (2013)
59. Mocanu, E., et al.: Unsupervised energy prediction in a smart grid context using reinforcement cross-building transfer learning. Energies. 116, 646–655 (2016). https://doi.org/10.1016/j.jenbuild.2016.01.030

How to cite this article: Mai, T.T., et al.: Exploring regression models to enable monitoring capability of local energy communities for self-management in low-voltage distribution networks. IET Smart Grid. 5(1), 25–41 (2022). https://doi.org/10.1049/stg2.12049
**FIGURE A1** A pair plot for the transformer phase Loading ($S_{trans}^j$) and voltage magnitudes ($V$) at the point of connections of the houses H25, H14, H51, H34, H52, H22, H21, and H54 connecting to the same phase. This pair plot consists of histogram plots along the diagonal, Pearson correlation coefficients below the diagonal, and scatter plots above it.
**Figure A2** A pair plot for the transformer phase A loading ($S_{trans}^A$) and voltage magnitudes ($V$) at the point of connections of the houses H5, H30, H4, H9, H49, H20, H55, and H46 connecting to the same phase. This pair plot consists of histogram plots along the diagonal, Pearson correlation coefficients below the diagonal, and scatter plots above it.