IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Machine Learning Approach for Detecting Unemployment using the Smart Metering Infrastructure

Casimiro A. Curbelo Montañez[1], William Hurst[1]

[1]Department of Computer Science, Liverpool John Moores University, Liverpool, L3 3AF UK

Corresponding author: Casimiro Aday Curbelo Montañez (e-mail: contact@acurbelo.com).

This research project is funded by the EPRSC - EP/R020922/1.

ABSTRACT Technological advancements in the field of electrical energy distribution and utilization are revolutionizing the way consumers and utility providers interact. In addition to allowing utility companies to monitor the status of their network independently in autonomous fashion, data collected by smart meters as part of the wider advanced metering infrastructure, can be valuable for third parties, such as government authorities. The availability of the information, the granularity of the data, and the real-time nature of the smart meter, means that predictive analytics can be employed to profile consumers with high accuracy and approximate, for example, the number of individuals living in a house, the type of appliances being used, or the duration of occupancy, to name but a few applications. This paper presents a machine learning model comparison for unemployment prediction of single household occupants, based on features extracted from smart meter electricity readings. A number of nonlinear classifiers are compared, and benchmarked against a generalized linear model, and the results presented. To ensure the robustness of the classifiers, we use repeated cross validation. The results revealed that it is possible to predict employability status with Area Under Curve (AUC) = 74%, Sensitivity (SE) = 54% and Specificity (SP) = 83%, using a multilayer perceptron neural network with dropout, closely followed by the results produced by a distance weighted discrimination with polynomial kernel model. This shows the potential of using the smart metering infrastructure to provide additional autonomous services, such as unemployment detection, for governments using data collected from an advanced and distributed Internet of Things (IoT) sensor network.

INDEX TERMS Classification, consumer profiling, machine learning, smart meter.

I. INTRODUCTION

The emergence of complex modern electrical systems, such as the smart grid, has revolutionized the way electricity is generated, distributed and monitored [1]. Within the smart grid implementation, the Advanced Metering Infrastructure (AMI) is a system that integrates smart meters, communication networks, and data management systems. The real-time availability of the data generated by smart meters creates a sustainable and energy efficiency process. Its sophistication allows for bidirectional communication between the utility companies and the consumers. Smart meters are electronic devices and key components in the AMI that measure electricity, gas or even water usage at the installed facility. The smart meters communicate the information to both, utility company and the consumer without the need for an operator or any micromanagement. Measured information is shown via an in-home display (IHD), facilitating accurate billing and making consumers aware of their energy usage. Nevertheless, the benefits of smart meters are not restricted to this. The large volumes of data generated by these smart devices are being mined by different entities, such as energy utility companies, government authorities and researchers. The derived insights are used to drive business strategy, identify factors influencing domestic energy consumption [2], and provide predictive maintenance within industrial automation. More recently, they have been used in an IoT setting to detect daily living activities in patients with progressive neurodegenerative disorders such as dementia [3] using nonintrusive load monitoring (NILM) [4].

Consequently, new challenges have also resulted from the incorporation of these new technologies [5]. Particularly regarding how the information collected is secured and

privacy maintained [6], [7]. In this sense, utility companies are skeptical to share their data to the public, which hinders research in smart meter data analytics. Nonetheless, over the past years, numerous anonymized or semi-anonymized datasets from both, household and small-medium enterprise (SME) sources, have been made publicly available [8].

An example of anonymized and publicly available load datasets for research purposes is provided by the Commission for Energy Regulation (CER), the regulator organism for the electricity and gas sectors in Ireland [9]. The CER launched the Electricity Customer Behaviour Trials (CBTs) during 2009 and 2010. Their aim was to assess the impact of smart meters in consumers energy usage behavior, based on different parameters such as demographics, lifestyles, and residence characteristics [10].

This dataset was used by P. Carroll et al. [2] to produce official statistics about factors influencing domestic energy consumption. Their research determines household composition from smart meter data via classification analysis, using generalized linear model (GLM) and neural networks (NNs). Results revealed that neither approach was capable of classifying households with high accuracy using solely smart meter data, with both models performing similarly. Despite this outcome, smart meter data can still be used to provide useful insights in the context of official statistics.

For example, the CER dataset was also used by S. Arora and J. Tailor to forecast electricity consumption from smart meter data [11]. The authors used different implementations of kernel density (KD) and conditional kernel density (CKD) methods to generate probability density estimates from smart meter records. Their approach aimed to assist consumers and energy suppliers in reducing electricity usage and help towards designing advanced time-of-use pricing strategies respectively. The results demonstrated that utilizing CKD methods outperformed unconditional KD estimators when accommodating seasonality in energy consumption, also taking the undelaying variability into consideration.

Machine learning techniques have become widely used for classification and regression tasks in many areas of research [12]–[14], including electricity data analysis [15], as alternatives to more traditional statistical methods. Currently, many machine learning applications to smart meter data have focused on forecasting consumers' electricity consumption, where approaches such as support vector regression (SVR) and NNs has been used [16], [17]. Alternatively, consumer categorization based on load profiling has also been an active area of research [18]. In these types of studies, load patterns or electricity consumption behaviors are extracted from residential, commercial and industrial electricity consumers to categorize them based on load pattern similarities. This is performed generally using unsupervised clustering algorithms [19].

Employability is a key indicator of the health of an economy. A higher level of unemployed workers typically means less total economy production, which can lead to social and political disturbance [20]; while causing serious distress on the economy. In this sense, governments have the responsibility to ensure that citizens receive appropriate counselling and support when looking for a job. Methods for detecting unemployment levels commonly rely on longitudinal data sets created using surveys from official authorities along with statistical techniques, such as Markov models [21]. Additionally, unemployment rates and their consequences at multiple scales have been observed using data from smart phones, including call detail records (CDRs) [22], Global Positioning System (GPS) log data [23] or Google searches [24]. These types of logs are comprised of time series data (e.g. weekly, daily or, hourly) that has become increasingly prevalent for supporting economic statistics-based research, with the aim of modelling unemployment rates [23], [24]. However, little or non-existing literature is available on statistical analysis using non-traditional data such as smart meter data to identify unemployment levels. Therefore, instead of forecasting unemployment, we provide an approach to classify consumers based on electricity consumption using smart meters and machine learning techniques.

The aim of this study is, thus, to conduct data preprocessing and apply analytic techniques in a smart meter data stream, to predict if a consumer is unemployed. Unlike conventional meters, smart meters collect information automatically, with high frequency. This, in turn, enables accurate consumer profiling by extracting residents' behavioral patterns. To achieve this, sixteen features are extracted from the electricity usage of participating consumers. Their performance when discriminating between employed and unemployed status are tested using different linear and nonlinear classifiers. Logistic regression methods are the most commonly used parametric models for the analysis of binary outcome variables. Thus, GLM, an extension of traditional linear models [25], is used as baseline model before conducting experiments with more complex nonlinear approaches. A total of six models are evaluated in this study. This is in addition to classification methods commonly used in the study of smart meters data analysis, such as NNs and GLM. Specifically, we utilized gradient boosting machines (GBM), classification and regression trees (CART), random forest (RF), and Distance Weighted Discrimination (DWD) with Polynomial Kernel. For each classifier, hyperparameters are tuned using a grid search approach, while model evaluation is performed using five repeats of 10-fold cross validation (CV). To the best of our knowledge this is the first study of its kind, in which the CER data has been used to study unemployment detection in single household consumers using smart meter readings; while comparing standard statistical models with state-of-the-art machine learning models such as NNs and DWD with Polynomial Kernel.

The remainder of this article is structured as follows: Section II describes the data preprocess and classification approaches used in the proposed method. In Section III

hyperparameter tuning and classification results are presented. In Section IV a discussion of performance/validation of the machine learning comparison is provided. Finally, conclusion and future directions are outlined in Section V.

II. MATERIALS AND METHODS

In this section, the dataset employed in this study is introduced and the features extracted from it described. Likewise, the different machine learning models used for classification are presented. Further details about how models were tuned and evaluated are also provided.

A. DATA DESCRIPTION

The analysis conducted in this paper is applied to the data collected in the electricity smart metering technology trials carried out by Electricity Supply Board (ESB) networks as part of the CER Smart Metering Project in Ireland, publicly available at [9]. The data is comprised of over 6.000 smart meters for residential homes, SMEs and other locations. Electricity load data was recorded by the smart meters at half hourly intervals during the trial over 18-months. Each smart meter data usage file is composed of three columns: i) unique household meter ID, ii) time stamp, and iii) electricity readings, for 30 minutes intervals in kWh. An example of three consecutive hour readings for smart meter ID 1000 is shown in Table 1. The time stamp information is provided in Julian's day format; hence, a data manipulation step was required to obtain the actual date and time for the electricity readings.

The dataset was also accompanied by a pre- and post-trial survey with detailed information about household members. Information such as sex, number of occupants or employability status was available and used in this study as features or to label the smart meter data. Labelling the data allows the application of supervised machine learning approaches, such as classification.

TABLE 1
SMART METER ELECTRICITY DATA SAMPLE (30 MINUTE INTERVALS)

Meter ID	Time Stamp	Reading (kWh)
1000	19707	0.046
1000	19708	0.044
1000	19709	0.042
1000	19710	0.084
1000	19711	0.091
1000	19712	0.094

B. DATA PREPROCESSING

A subset of 803 different single occupants' smart meter data was selected from the dataset. The subset is the result of the exclusion of multiple occupant data; as statistical models have provided better insights in households of single persons [2] where energy readings are not affected by other family members. Furthermore, smart meter data for a twelve-month period between January and December 2010 was considered when conducting the experiments.

For the purpose of this study a binary class was created from the employment status reported in the residential trial survey: employed and unemployed. Therefore, the employed class includes information about all occupants that reported to be employed, self-employed (with employees) and self-employed (with no employees). Conversely, the unemployed class includes information about individuals who reported to be unemployed (actively seeking work), unemployed (not actively seeking work) and retired. Individuals looking after relative or family (carers) were discarded from further experiments. Hence, four smart meters were removed, resulting in a final subset of 799 smart meters employed to conduct classification analysis. Of these, 299 belong to the employed class whereas 500 to the unemployed class (see Figure 1), which represents an imbalanced class problem. This is common in most real-world classification problems, where classes do not make up an equal portion of the dataset.

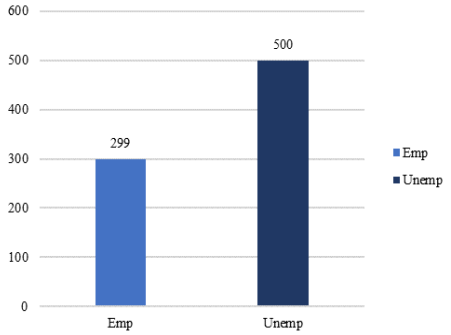


Figure 1. Binary class distribution

Individual consumer load profiles can be seen as a unique fingerprint to conduct a classification of unemployment, thanks to the granularity provided by smart meter half-hourly readings. However, the raw data captured by the smart meters is stochastic, so that the application of feature extraction is needed. This step provides a reduced representation of the data while retaining the key information but facilitating the computation. Hence, a total of 16 features were considered, including the consumers' sex, and 15 features extracted from the half-hourly meter readings to summarize each consumer's unique load profile. These features are standard descriptive statistical measures, while others have been actively used in other research [26]. Table 2 provides a short description of the features extracted for the total number of half hourly readings over a time frame of one year.

D. MODEL COMPARISSON

In binary classification problems, the aim is to predict class labels based on a given set of attributes. This is carried out by fitting a statistical model during a training stage, and then using that model to make predictions. For an observed training dataset of n pairs, where $\{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \mathbb{R}^p$, and the binary outcome is $y_i \in \{0,1\}$, the classifier will fit a discriminant

function *f*, to construct a rule to classify data point x_i to either 0 or 1 according to $f(x_i)$. There are several significant binary classification methods, such as kernel methods, ensemble methods, and neural networks [27].

TABLE 2
SUMMARY OF EXTRACTED FEATURES

Index	Feature	Description
1	Sex*	Sex reported by household occupant
2	Morning	Morning average consumption
3	Afternoon	Afternoon average consumption
4	Evening	Evening average consumption
5	Night	Night average consumption
6	Monday	Monday average consumption
7	Tuesday	Tuesday average consumption
8	Wednesday	Wednesday average consumption
9	Thursday	Thursday average consumption
10	Friday	Friday average consumption
11	Saturday	Saturday average consumption
12	Sunday	Sunday average consumption
13	Av	Annual average consumption
14	Med	Annual median consumption
15	SD	Annual standard deviation consumption
16	Var	Annual variance consumption

* Selected from trial survey

The experiments conducted in this paper consist of a model comparison to test how accurately the classifiers are able to discriminate between employed and unemployed individuals based on the features extracted from the smart meter readings. A list with the proposed machine learning models to be compared is given in Table 3.

TABLE 3
CLASSIFIERS COMPARED IN THIS PAPER

Classifier	Category
Generalized linear model with lasso and elastic-net regularization.	Linear
Stochastic Gradient Boosting Machines.	Nonlinear
Classification and Regression Trees.	Nonlinear
Random Forest.	Nonlinear
Multilayer Perceptron with Dropout.	Nonlinear
Distance Weighted Discrimination with Polynomial Kernel.	Nonlinear

1) GENERALISED LINEAR MODEL

In order to compare models of different nature, this research uses an extension of traditional linear models, GLM [25] with Lasso and elastic-net regularization [28], as a baseline model. Depending on the distribution and function of choice, GLM can be used for classification or prediction. However, since the response is categorical and binary, in this paper, classification analysis is performed using a binomial distribution.

In simple linear regression problems, it is assumed that the response variable *y* (independent observations) is related to a set of explanatory variables *x* (call predictors) by

$$y = \beta_0 + \beta x_i + \in_i \tag{1}$$

also expressed as

$$E(Y_i) = \beta_0 + \beta x_i \tag{2}$$

where β_0 is the intercept term, β is the parameter vector, $\in \sim N(0, \sigma^2)$ is a Gaussian random variable that represents noise in the model, while each data point is identified by the index *i*.

The regularisation penalty, or elastic net regularisation penalty, combines Lasso (l_1) and Ridge regression (l_2) penalties parametrised by the *alpha* (α) and *lambda* (λ) parameters. These penalties are introduced to the model to avoid overfitting, reduce variance of the predictor error, and handle correlated predictors [29]. Therefore, elastic net regularisation penalty is defined by the weighted sum of l_1 and l_2 norms of β and is defined in (3) where $\lambda P_\alpha(\beta)$ is subtracted from the optimised likelihood.

$$\lambda P_\alpha(\beta) = \lambda \left[\alpha \|\beta\|_1 + \frac{1}{2}(1-\alpha)\|\beta\|_2^2 \right] \tag{3}$$

For N observations, the problem to be optimised is thus:

$$\min_{\beta,\beta_0} -\left[\frac{1}{N} \sum_{i=1}^{N} y_i \cdot (\beta_0 + x_i^T \beta) - \log(1 + e^{(\beta_0 + x_i^T \beta)}) \right] + \lambda \left[\alpha \|\beta\|_1 + \frac{1}{2}(1-\alpha)\|\beta\|_2^2 \right], \tag{4}$$

where the negative binomial log-likelihood is used by the objective function for the penalised logistic regression.

In this paper, GLM was implemented using the *glmnet* R package [30] and the parameters alpha and lambda tuned for optimal model selection.

2) DECISION TREES

Decision trees [31] are classification and regression techniques based on recursive partitioning [32]. They are widely used in data mining due to their ability to represent results in a simple and interpretable tree format. A tree representation is adopted to create a training model that predicts target variables (class) by learning decision rules inferred from the training data. As depicted in Figure 2, decision trees construct from a root node, internal nodes, and terminal nodes or leaves. A single root node is assigned to the entire training data in the tree. Each internal node corresponds to an attribute, and individual terminal nodes correspond to a class label. The process of growing the tree is conducted by splitting the source data, from the root node, into different branches (subsets) based on the attribute value following a splitting rule. The process is repeated following a binary

recursive partitioning manner at each node, until no additional splits can be formed.

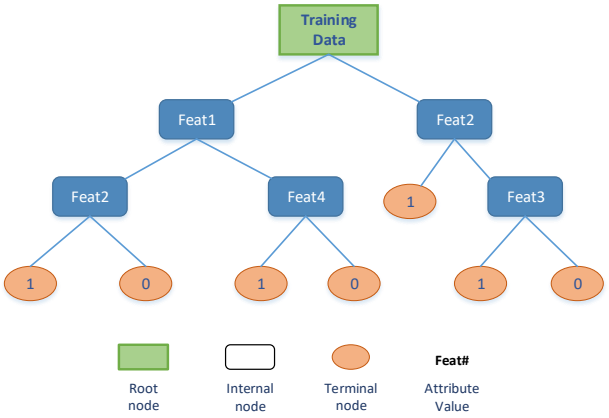


Figure 2. Decision tree workflow

Several approaches have been developed to build decision trees [33]. One of the most popular is the classification and regression trees (CART) algorithm of Breiman et al. [32]. In this paper, classification and regression trees classifier is implemented using the *rpart* R package [34], where only one parameter, the complexity parameter (cp), was tuned.

Decision tree-based models using the '*average over an ensemble of trees*' rather than a single tree have been developed to provide additional advantages to those offered by decision trees [35]. Two popular ensemble tree-based methods are GBM [36], also known as stochastic gradient boosting (SGB), and RF [37].

GBM is supported by two powerful techniques, gradient-based optimisation and boosting [38]. It computes the gradient to minimize the model's loss function in the training data. Whereas, the boosting algorithm adds new weak-base learner models to the ensemble in a gradual, additive and sequential manner, providing a more accurate estimate of the response variable (stronger learner). There are a number of different approaches to prevent GBM models from overfitting, including regularization through shrinkage [39]. Shrinkage, also known as the learning rate, is used to reduce the impact of each new model added to the ensemble.

To implement GBM in our experiments, we used the *gbm* R package [40]. In this occasion, the parameters listed in Table 4 were tuned to find the best models.

TABLE 4
GBM TUNING PARAMETERS

Parameters	Description
n.trees	Total number of trees to fit
interaction.depth	Maximum depth of each tree
shrinkage	Controls the learning rate
n.minobsinnode	Minimum number of observations in the terminal nodes of the trees

In comparison, random forest is an optimal approach for constructing ensembles. The strength of RF derives from using random subsamples of the training data (bootstrap aggregation or bagging) and randomising the algorithm for learning cased-level classifiers. The random forest is constructed by generating several bootstrap samples using the original data. For each bootstrap sample, the tree is grown, and a random subset of predictor variables is selected to split the tree node. The best split is calculated using these randomly selected candidate variables. This process is continued until the tree is fully grown without pruning, resulting in a forest of decision trees.

The RF is implemented using the ranger R package [41]. Parameters listed in Table 5 were tuned to achieve the optimum classifier performance.

TABLE 5
RF TUNING PARAMETERS

Parameters	Description
mtry	Number of variables to possibly split at in each node.
splitrule	Splitting rule. For classification: *extratrees* or *gini* are available.
min.node.size	Minimal node size

5) MULTILAYER PERCEPTRON WITH DROPOUT

Artificial neural networks are machine learning models that imitate biological neurons in the human brain to conduct function approximation and pattern recognition from a set of samples [42]. The neurons are arranged into layers, and each layer is fully connected with neurons in the next layer. One of the most frequently applied ANN architectures is the feedforward ANN (FNN) also known as the multilayer perceptron (MLP) [43]. The goal of the MLP is to find a function f: X→Y, capable of approximating the values of output variables (Y) dependent on the set of input variables (X). At its most basic level, an MLP has an *input layer*, *hidden layer(s)*, and an *output layer* as depicted in Figure 3.

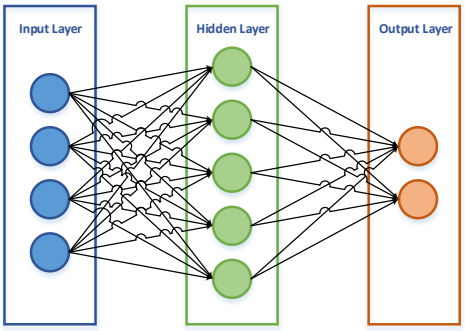


Figure 3. Illustration of a single hidden layer NN.

We used labelled training samples $(x^{(i)}, y^{(i)})$ from the CER smart meter data to train an MLP network for supervised learning tasks. A complex nonlinear hypothesis $h_{w,b}(x)$ is defined in (5) using a feed forward ANN, with weights (W) and bias (b) parameters fitted to the data, based on formal definitions in [44]. Taking a set of labelled samples $\{x_1,$

$x_2, \ldots x_n\}$ and a bias unit b (+1 intercept term) as input, single computational units or neurons output:

$$h_{W,b}(x) = f(W^T x) = f\left(\sum_{i=1}^{n} W_i x_i + b\right) \quad (5)$$

where f: $\mathbb{R} \mapsto \mathbb{R}$ represents the activation function. Activation functions, such as the sigmoid function, hyperbolic tangent (tanh) and rectifier linear unit (ReLU) are commonly used in many neural network configurations. However, in this paper, the hyperbolic tangent function, defined in (6), provided the most favorable results in the experiments conducted with MLP.

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z - e^{-z}} \quad (6)$$

A single hidden MLP layer, was implemented, generally computed as the activation or output value of node i in layer l:

$$h_{w,b}(x) = a_i^{(l)} = f\left(\sum_{j=1}^{n} W_{ij}^{(l)} a_j^{(l)} + b_i^{(l)}\right) \quad (7)$$

The number of epochs, that is, the number of times that the whole training set is shown to the MLP during training, was set to 100. The adaptive learning rate ADADELTA [45] was used for stochastic gradient descent optimization to balance the global and local search efficiencies. Typically, the weights for all neurons are learned via stochastic gradient descent.

Dropout regularization is a technique that prevents neurons from co-adapting, which reduces overfitting. This approach has been successful in many domains including object classification, speech recognition or analysis of biology data [46]. Dropout achieves this by randomly selecting a fraction of neurons in each layer and dropping them out of the training process by setting the neuron values to zero. When performing tests, no neurons are dropped, rather their weights are scaled appropriately based on:

$$W_{test}^{(l-1)} = pW^{(l-1)} \quad (8)$$

where l is the layer, and the neurons are dropped with probability p (i.e. a value of p = 0.5 indicates that 50% of the neurons are dropped at an iteration). Thus, during test, the incoming weights to the layer l are scaled by p, according to (8).

For the implementation of MLP with dropout, *keras* R package is employed [47]. The main parameters tuned are listed in Table 6.

6) DISTANCE WEIGHTED DISCRIMINATION

Distance weighted discrimination [48] is a classification (discrimination) method, developed originally to overcome the limitations of support vector machines (SVM) in high dimension, low sample size (HDLSS) context; although, its use can be extended to other scenarios. DWD overcomes the data piling problem [49] in high dimensional situations, while improving generalizability, and uses interior-point methods for second-order cone programming (SOCP) problems.

TABLE 6
MLP TUNING PARAMETERS

Parameters	Description
size	Number of hidden units
dropout	Dropout rate
batch_size	Number of patterns shown to the MLP before the weights are updated.
lr	Learning rate
rho	Gradient moving average decay factor
decay	Learning rate decay over each update
cost	Cost
activation	Activation function to use

This is possible as DWD identifies the hyperplane that minimizes the sum of inverse distances from each data point. Marron et al. [48] formulated DWD as the separating hyperplane that minimizes the total inverse margins of all the data points, outlined in (9):

$$\min_{w_0, w, d_i, \eta_i} \left(\sum_{i=1}^{n} \frac{1}{d_i} + c \sum_{i=1}^{n} \eta_i \right), \quad (9)$$

subject to $d_i = y_i(w_0 + x_i^T w) + \eta_i \geq 0$, $\eta_i \geq 0$, $\forall i$, and $w^T w = 1$. Where di is equivalent to the Euclidean distance, w is a unit normal vector, and η_i are nonnegative slack variables introduced to ensure that all $y_i(w_0 + x_i^T w) + \eta_i$ are non-negatives, in case that the classes are not separable. Equation (9) was reformulated lately by Marron et al. where the q^{th} power ($q > 0$) of the inverse distances was introduced to replace the reciprocal in the standard DWD optimization problem. Hence, the new generalized formulation of DWD is as follows:

$$\min_{w_0, w} \left(\sum_{i=1}^{n} \frac{1}{d_i^q} + c \sum_{i=1}^{n} \eta_i \right), \quad (10)$$

subject to $d_i = y_i(w_0 + x_i^T w) + \eta_i \geq 0$, $\eta_i \geq 0$, $\forall i$, and $w^T w = 1$, which is equivalent to (9) when q = 1.

Distance Weighted Discrimination with polynomial kernel was implemented using the *kerndwd* R package [50]. Several parameters were tuned, as indicated in Table 7.

TABLE 7
DWD TUNING PARAMETERS

Parameters	Description
lambda	A user supplied lambda sequence
qval	The exponent index of the generalized DWD
degree	The degree of the polynomial, bessel or ANOVA kernel function
scale	The scaling parameter of the polynomial kernel function

7) HYPERPARAMETER OPTIMISATION

Hyperparameter optimization aims to find an optimal set of hyperparameters that minimise the generalisation error E for a

given learning algorithm. This in turn, produces classifiers with a high predictive performance [51]. Methods for optimizing hyperparameters in machine learning approaches include grid search, Bayesian optimization, random search, and gradient-based optimization. Grid search is a commonly documented approach in literature [52], and the method considered in the approach put forward in this paper.

Typically, machine learning models are trained using a training set and validated using a holdout or validation set. To ensure that overfitting does not occur, repeated cross validation is used during the modelling stage. This resampling technique allow for the examination of the classification performance and a decision, from a range of results, on which classifier produces the highest result. Therefore, to evaluate and validate the models, we created a train/test data split; 80% of the data was used for training whereas 20% was used for testing. A 10-fold (k = 10) CV repeated five times was used to validate the models and for the selection of the best tuning parameters. To do this, the training set is randomly divided into 10 nearly equal segments. Next, one of the folds is used as a validation set (hold out set) and the classifier is fit on the remaining *k-1* folds. This step is repeated five times. Finally, the result of the *k*-fold cross-validation is obtained by summarizing the average performance across hold-out predictions. In our case, 50 results (five repeats of 10-fold CV) are generated for each best model. Then, the averaged distributions (50 results) between the models is compared.

Therefore, in this paper, each model is tuned using a grid search approach, and evaluated using 10-fold CV resampling with five repetitions.

8) MODEL PERFORMANCE

Model performance is measured using numerical and graphical approaches [53]. In binary classification, informative measures of generalizability are derived from a 2x2 contingency table to calculate sensitivity (SE), specificity (SP) and accuracy (Acc) among other metrics.

Sensitivity, or true positive rate (TPR), is used to quantify how effectively the classifiers correctly recognize actual positive cases (i.e. employed occupants). Whilst SP, or true negative rate (TNR), represents the classifier's ability to correctly recognize actual negative cases (i.e. unemployed occupants). Therefore, SE and SP can be defined as in (11) and (12) respectively, where TP = True positive, FP = False positive, TN = True negative and FN = False negative.

$$\mathrm{SE} = \mathrm{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\mathrm{SP} = TNR = \frac{TN}{TN + FP} \tag{12}$$

The proportion of predicted positives that are actual positives is called precision, or positive predictive value (PPV). Conversely, a negative predictive value (NPV) is the proportion of predicted negatives that are actual negatives. Therefore,

$$\mathrm{Precision} = PPV = \frac{TP}{TP + FP} \tag{13}$$

$$NPV = \frac{TN}{FN + TN} \tag{14}$$

Classification accuracy, defined in (15), represents the percentage of total items classified correctly and is utilized frequently to assess the quality of predictive models, where $N = TP+TN+FP+FN$.

$$\mathrm{Acc} = \frac{TP + TN}{N} \tag{15}$$

However, this performance measure could be misleading, particularly in large class imbalance datasets, since overall accuracy varies with class frequency [53], [54]. Thus, to overcome this limitation, balanced accuracy (bAcc) [55] is considered and used in this paper along with other measures to evaluate model performance. Formally, bAcc can be defined as follows:

$$bAcc = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) \tag{16}$$

The receiver operating characteristic (ROC) curve is a standard technique used as a graphical performance measure to summarise the predictive performance of binary classification models [54], [56]. The ROC curve plots the TPR against false positive rate (FPR) measurements produced by a classification model, where each point on the ROC curve corresponds to a classifier. Additionally, the ROC is summarised commonly by a single measure known as the area under the ROC curve (AUC). AUC measures the probability that test values from a randomly selected pair of binary class samples are correctly ranked and is thus a convenient global measure for the quantification of classification accuracy. For an algorithm that perfectly classifies, the AUC will be 1, whereas a classifier that randomly assigns labels, will be 0.5 [57]. To measure and report AUC properly, it is important to determined its confidence interval (CI) [58]. In this paper, thus, CIs are computed using the R package pROC [59].

The F1 metric, also known as the F-score, or F-measure, takes precision and recall into account and represents the harmonic mean between the two as shown in (17).

$$\mathrm{F}_1 = 2 \times \left(\frac{\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}}\right) \tag{17}$$

Each of these model performance metrics have been used in other research investigations in order to evaluate binary classification experiments [13].

III. RESULTS

In this section, our experimental results are reported. Six different classifiers were implemented to discriminate between employed and unemployed status, using sex and 15 descriptive statistics features listed in Table 2. All the analyses carried out in this work were conducted using the free software environment for statistical computing and graphics, R [60].

A. HYPERPARAMETER SELECTION

The selection of an approximately optimal configuration for each classifier is addressed via grid search hyperparameter tuning. Based on empirical analysis, the model specific tuning values reported in Table 8 produced the models with the best AUC.

TABLE 8
SELECTED TUNING PARAMETERS

Model	Best
GLM	α = 0.6123 λ = 0.003673802
GBM	n.trees = 3110 interaction.dedepth = 10 shrinkage = 0.1402978 n.minobsinnode = 11
CART	cp = 0
RF	mtry = 13 splitrule = extratrees min.node.size = 14
MLP	Size = 13 dropout = 0.2 batch_size = 210 lr = 0.2668208 rho = 0.153046 decay = 0.3743612 cost = 5.917387 activation = tanh
DWD	lambda = 0.0002112 qval = 0.29436 degree = 2 scale = 0.072

The different grid search used to select the optimal tuning parameters for each model are shown in Figure 4. In some instances, only one or two parameters were tuned, see GLM and CART in Figure 4 a) and c) respectively. Conversely, models such as GBM, RF, DWD and, especially MLP have a large dimensional hyperparameter search space, resulting in several model configurations being tested as shown in Figure 4 b), e) and d).

B. MODEL COMPARISSON

In this section, we highlight the performance of the proposed machine learning comparison using extracted features and CV.

1) CROSS VALIDATION CLASSIFIER PERFORMANCE

Results from resampling using 10-fold cross-validation repeated five times are provided in Table 9. For each model, the calculated average SE, SP and AUC values across hold-out predictions are reported.

TABLE 9
CROSS-VALIDATION MODEL PERFORMANCE

Model	Sensitivity	Specificity	AUC
GLM	0.407	0.894	0.752
GBM	0.487	0.761	0.678
CART	0.487	0.736	0.665
RF	0.452	0.852	0.722
MLP	0.535	0.864	0.771
DWD	0.535	0.846	0.774

The dot plot in Figure 5 depicts the spread of the estimated results for the ROC, SE and SP for the different classifiers, using a 95% confidence interval (CI). The models are sorted from highest to lowest ROC, where overlaps of the spreads between models can be observed, especially in DWD and MLP, and GBM and GLM.

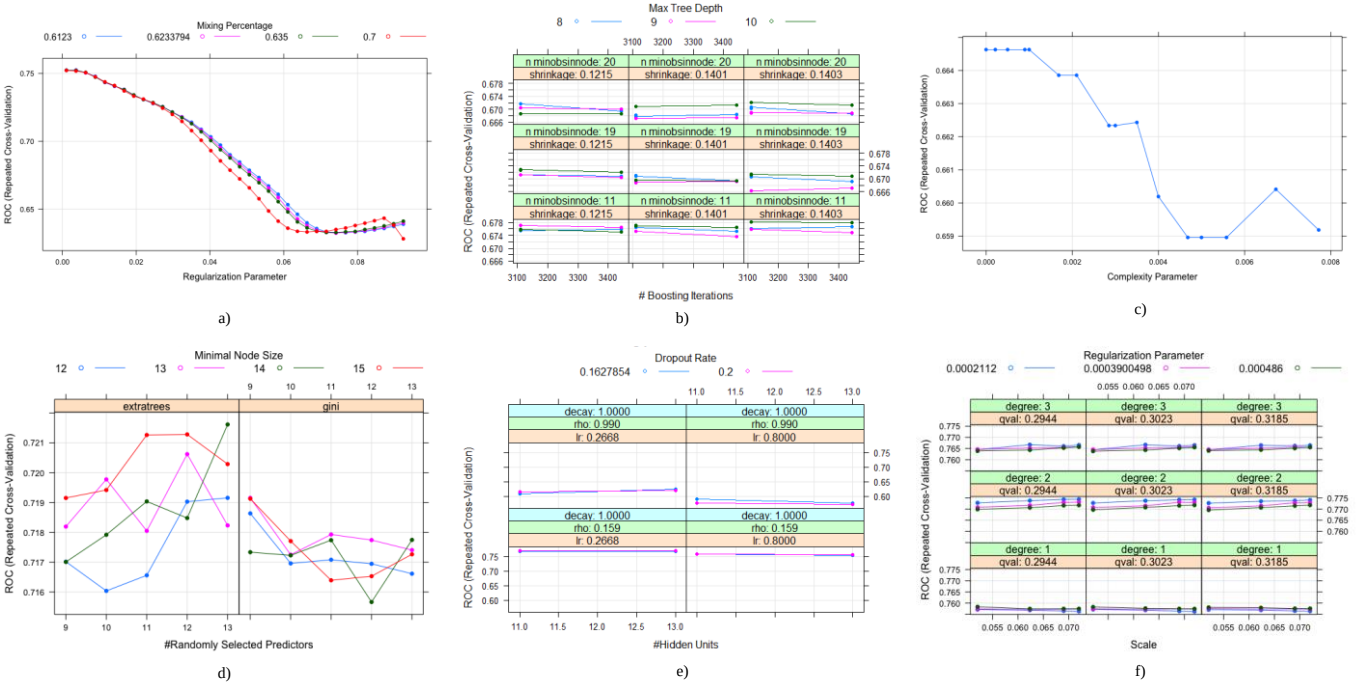


Figure 4. From a) to f) grid search for GLM, GBM, CART, RF, MLP and DWD respectively.

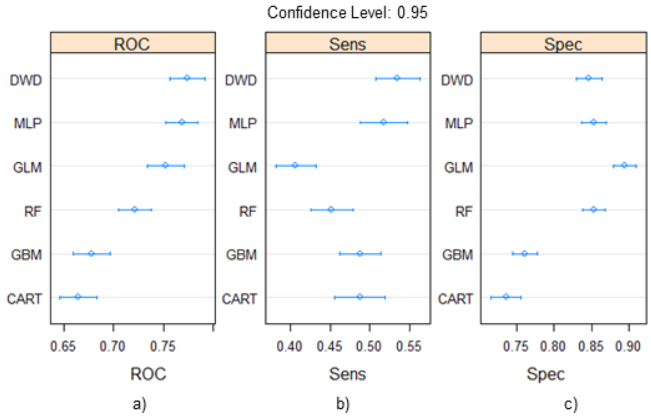


Figure 5. **Resampling distributions comparison between models for a) ROC, b) SE and c) SP respectively.**

2) TEST SET CLASSIFIER PERFORMANCE

The performance metrics for the test set are shown in Table 10.

TABLE 10
TEST SET MODEL PERFORMANCE

Model	bAcc	SE	SP	PPV	NPV	F1	AUC
GLM	0.611	0.373	0.848	0.595	0.694	0.458	0.712
GBM	0.680	0.542	0.818	0.640	0.750	0.587	0.700
CART	0.596	0.525	0.667	0.484	0.702	0.504	0.653
RF	0.643	0.458	0.828	0.614	0.719	0.524	0.739
MLP	0.685	0.542	0.828	0.653	0.752	0.593	0.740
DWD	0.687	0.576	0.798	0.630	0.760	0.602	0.738

The ROC curves depicted in Figure 6 are used as a graphical performance measure to summarize the predictive performance of the six models. The cut-off values for the false and true positive rates using the test set are shown in each of the ROC curves for the different implemented classifiers. Additionally, CIs for the AUC are represented graphically by the blue light areas in the plots and their numerical variations printed.

IV. DISCUSSION

Smart meters are a powerful source to mine consumer information due to i) the growth in uptake, ii) the low installation cost and iii) the ease of installation. As previously mentioned, this has opened a new market of personalized intelligent services to analyze the recorded meter data. This paper provides a comparison of six machine learning algorithms for predicting user employability from smart meter electricity data, using an anonymized and publicly available dataset. As derived from [2], statistical models provide better insights in households of single persons. Therefore, only data from smart meters installed in houses with a single occupant was used to conduct the experiments.

As depicted in Figure 1, the binary class derived from the CER dataset is not perfectly balanced. This is a real-world problem where it is difficult to collect an ideal number of classes consisting on having approximately 50% of the individuals belonging to class one and 50% to class two.

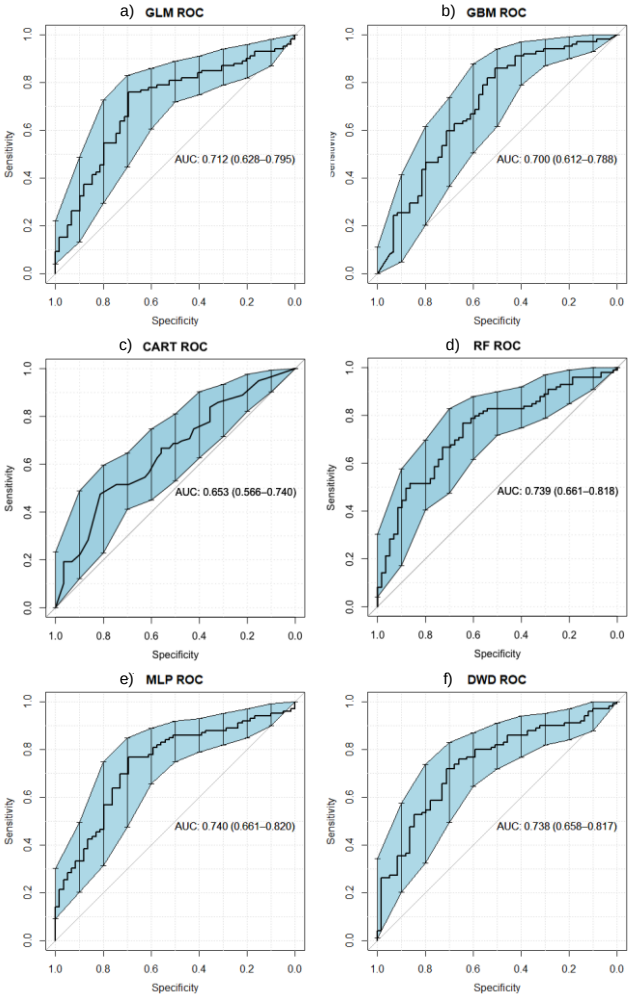
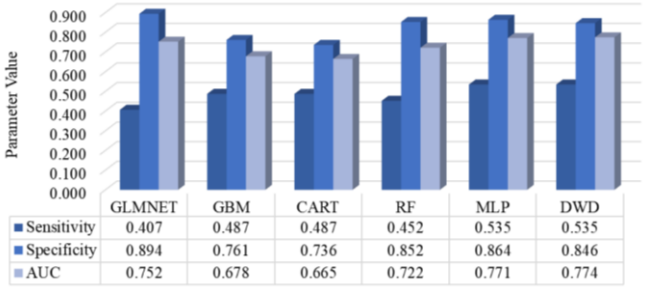


Figure 6. **From a) to f), model ROC curves of the different classifiers. The light blue areas are the confidence intervals. The AUC values and their variations are printed in the middle of the plot.**

In the literature, however, there is no clear consensus about the cut-off value below which a dataset is considered to be imbalanced [61]. In our case, 37.4% of the smart meters belong to employed consumers and 62.6% to unemployed. Instead of considering common strategies in supervised learning to overcome class imbalance [62] such as resampling or using Synthetic Minority Over-sampling Technique (SMOTE), we evaluated the models using bAcc, AUC, PPV, NPV and F1 performance metrics, which are research standard approaches to provide an overview of how well the model performs, even in situations of class imbalance. Additionally, models were tuned using repeated 10-fold CV on the training data. This ensures that the utilized machine learning algorithms are best adapted for the given problem.

A summary with the performance of the six classifiers in the CV modelling stage is shown in Figure 7. As depicted, DWD and MLP are the best models with ~77% AUC, achieved by both models, followed by the GLM. We observed a limitation in all the classifiers to classify correctly employed individuals (low sensitivities). This is most likely a result of

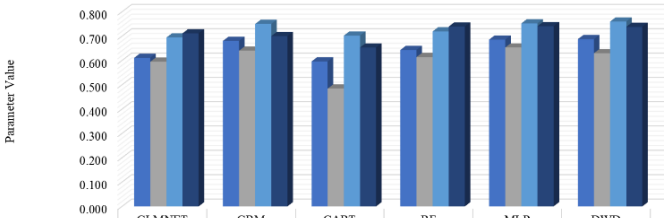
the class imbalance highlighted earlier. Despite this limitation, two of the classifiers (MLP and DWD) performed better than the rest (See Figure 7.).



	GLMNET	GBM	CART	RF	MLP	DWD
Sensitivity	0.407	0.487	0.487	0.452	0.535	0.535
Specificity	0.894	0.761	0.736	0.852	0.864	0.846
AUC	0.752	0.678	0.665	0.722	0.771	0.774

Figure 7. **CV results summary.**

When predicting new classes using the test set, all classifiers except CART yielded overall reasonable performance. Reported AUC values were higher than 50%, indicating that the classifiers are not randomly assigning labels to the samples. As in the training stage, the MLP and DWD remained the best models. However, it is observed that there are signs of overfitting. Overfitting occurs when models memorize training data but do not generalize to new cases [63]. To reduce the effect of overfitting, various regularization techniques are applied in the different models, such as elastic net regularization in GLM and dropout regularization in MLP. In Figure 8, a summary with the performance of each classifier is shown. We observed that the capacity of the classifiers to identify employed consumers (PPV) was always lower in comparison to unemployed consumers (NPV).



	GLMNET	GBM	CART	RF	MLP	DWD
Balanced Accuracy	0.611	0.680	0.596	0.643	0.685	0.687
PPV/Precision	0.595	0.640	0.484	0.614	0.653	0.630
NPV	0.694	0.750	0.702	0.719	0.752	0.760
AUC	0.712	0.700	0.653	0.739	0.741	0.738

Figure 8. **Summary of classifiers performance in the test set.**

Therefore, we achieved the best results in the test set using the MLP with dropout, 74% AUC with a 95% CI from 0.661 to 0.820, SE = 54%, SP = 83%, bAcc = 69%, PPV = 65%, NPV = 75% and F1 = 0.593 as observed in Table 10 and Figure 6. Furthermore, MLP was also the best model predicting employed consumers (PPV = 65%). These results were closely followed by those achieved by RF and DWD. Random forest achieved an AUC value of ~74%, similar to MLP and also DWD. Nevertheless, its bAcc (64%) was lower, while it performed worse than MLP and DWD when predicting positives and negative classes, PPV = 61% and NPV = 72% respectively. Conversely, although the AUC reported by DWD was slightly lower than MLP (See Table 10), this classifier performed better predicting unemployed people with a NPV = 76%. However, MLP and DWD performed similarly, so more experimental analysis will be necessary to conclude which, between the two, is better for the analysis of smart meter data.

The worst model performance was attained by CART, AUC = 65% (CI95% = 0.566-0.740), SE = 53%, SP = 67%, bAcc = 60%, PPV = 48%, NPV = 70% and F1 = 0.50 (See Table 10 and Figure 6). A PPV = 48% demonstrates the inability of CART to predict employed consumers. This result validates the use of ensemble algorithms, such GBM and RF, which are more powerful solutions and provide more advantages that simpler CART.

The ROC curve comparison, depicted in Figure 9, provides a more comprehensive view of the discriminative ability of the models. The corresponding SE and SP optimal values for each of the ROC curves are provided in Table 10 and depicted in Figure 7. The ROC curves for MLP, DWD and RF (top models) lie above the remaining models, where CART (dark blue) represents the lowest performing model in terms of AUC. Therefore, those models with ROC curves closer to the top left corner in Figure 9 show higher performance, where the AUC increases as the curve recedes from the diagonal line towards the top left corner of the graph.

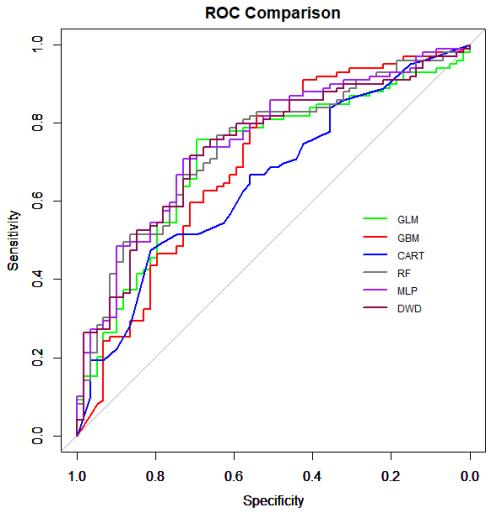


Figure 9. **Combined ROC curves plot for test set.**

Our main findings reveal that nonlinear approaches such as MLP, DWD and RF outperformed the use of GLM as a baseline linear model. This can be explained by the fact that smart meter data contains certain complexity or complex nonlinearities that can only be explained by nonlinear models, such as those presented in Table 3. Particularly, the use of distance weighted discrimination in smart meter data analytics has proven to be promising, since most analytics in this field are carried out using NNs, SVM, RF and other unsupervised clustering techniques.

To the best of our knowledge, this is the first time the CER data has been used to study employability in single household consumers using electricity meter readings. In addition, it is the first time that distance weighted discrimination with polynomial kernel has been compared with common linear and nonlinear models for employability prediction with promising results.

V. CONCLUSSION AND FUTURE WORK

The experiments presented in this paper describe an approach for employability classification using smart meter data, via a machine learning classification comparison. We provide a useful list of algorithms and their corresponding R packages, which can guide researchers when conducting binary classification experiments in smart meter data analytics. The gained insights highlight the potential of using nonlinear machine learning approaches, such as MLP and DWD along with smart meter data, to assist the identification of unemployed individuals. This has the potential for new personalized services to be created in public organizations aiming to reduce high levels of unemployment in society and, in turn, improve the economy.

Despite the encouraging results reported, the number of smart meter readings from single household occupants in the CER dataset was limited to 803 meters, while the class label for employment status was partially imbalanced. This has produced models capable of predicting non-employed individuals with reasonable accuracy, but limited when predicting the positive class, employed. In future work, multiclass classification will be conducted. To do so, the number of employment status label needs to be increased, particularly for employed class, in order to have a more balanced dataset. Separating between employed, non-employed and retired classes will provide more granular and detailed information about an individual´s employment status. This information can be used by the relevant authorities or utility providers and serves as a demonstration of the insights available when analyzing datasets generated by autonomous IoT systems within the Industry 4.0 revolution.

ACKNOWLEDGMENT

This research project has been funded by the EPRSC - EP/R020922/1. Data is available for request from the Commission for Energy Regulation (CER). (2012). *CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010* [dataset]. 1st Edition. Accessed via the Irish Social Science Data Archive. SN: 0012-00. www.ucd.ie/issda/CER-electricity

REFERENCES

[1] R. Rashed Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on Advanced Metering Infrastructure," *Int. J. Electr. Power Energy Syst.*, 2014.

[2] P. Carroll, T. Murphy, M. Hanley, D. Dempsey, and J. Dunne, "Household Classification Using Smart Meter Data," *J. Off. Stat.*, vol. 34, no. 1, pp. 1–25, Mar. 2018.

[3] C. Chalmers, P. Fergus, C. Curbelo Montanez, S. Sikdar, F. Ball, and B. Kendall, "Detecting Activities of Daily Living and Routine Behaviours in Dementia Patients Living Alone Using Smart Meter Load Disaggregation," *arXiv.org*. Mar-2019.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[5] A. Ghasempour, "Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges," *Inventions*, vol. 4, no. 1, p. 22, Mar. 2019.

[6] A. MacDermott, Qi Shi, M. Merabti, and K. Kifiyat, "Considering an elastic scaling model for cloud Security," in *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, 2013, pp. 150–155.

[7] P. McDaniel and S. McLaughlin, "Security and Privacy Challenges in the Smart Grid," *IEEE Secur. Priv. Mag.*, vol. 7, no. 3, pp. 75–77, May 2009.

[8] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges," *IEEE Trans. Smart Grid*, pp. 1–1, Feb. 2018.

[9] Commission for Energy Regulation (CER), "CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]. 1st Edition." [Online]. Available: www.ucd.ie/issda/CER-electricity.

[10] CER, "Electricity Smart Metering Technology Customer Behaviour Trials (CBT) Findings Report," Dublin, 2011.

[11] S. Arora and J. W. Taylor, "Forecasting electricity smart meter data using conditional kernel density estimation," *Omega*, vol. 59, no. 0, pp. 47–59, Mar. 2016.

[12] C. Curbelo *et al.*, "Machine learning approaches for the prediction of obesity using publicly available genetic profiles," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2743–2750.

[13] C. Curbelo, P. Fergus, A. Curbelo, A. Hussain, D. Al-Jumeily, and C. Chalmers, "Deep Learning Classification of Polygenic Obesity using Genome Wide Association Study SNPs," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[14] P. Larrañaga *et al.*, "Machine learning in bioinformatics," *Brief. Bioinform.*, vol. 7, no. 1, pp. 86–112, 2006.

[15] G. Capizzi, G. Lo Sciuto, C. Napoli, and E. Tramontana, "Advanced and Adaptive Dispatch for Smart Grids by Means of Predictive Models," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6684–6691, Nov. 2018.

[16] T. Zufferey, A. Ulbig, S. Koch, and G. Hug, "Forecasting of Smart Meter Time Series Based on Neural Networks," in *Dare*, vol. 10097, Springer International Publishing, 2017, pp. 10–21.

[17] K. Grolinger, A. L'Heureux, M. A. M. Capretz, and L. Seewald, "Energy Forecasting for Event Venues: Big Data and Prediction Accuracy," *Energy Build.*, vol. 112, pp. 222–233, Jan. 2016.

[18] A. M. Tureczek and P. S. Nielsen, "Structured Literature Review of Electricity Consumption Classification Using Smart Meter Data," *Energies*, vol. 10, no. 5, p. 584, Apr. 2017.

[19] Z. Jiang, R. Lin, and F. Yang, "A Hybrid Machine Learning Model for Electricity Consumer Categorization Using Smart Meter Data," *Energies*, vol. 11, no. 9, p. 2235, Aug. 2018.

[20] S. Raphael and R. Winter-Ebmer, "Identifying the Effect of Unemployment on Crime," *SSRN Electron. J.*, no. September 1998, 1999.

[21] S. R. G. Jones and W. C. Riddell, "The Measurement of Unemployment: An Empirical Approach," *Econometrica*, vol. 67, no. 1, pp. 147–162, Jan. 1999.

[22] J. L. Toole, Y.-R. Lin, E. Muehlegger, D. Shoag, M. C. González, and D. Lazer, "Tracking employment shocks using mobile phone data," *J. R. Soc. Interface*, vol. 12, no. 107, p. 20150185, Jun. 2015.

[23] D. Moriwaki, "Nowcasting Unemployment Rates with Smartphone GPS Data," in *Multiple-Aspect Analysis of Semantic Trajectories. MASTER 2019. Lecture Notes in Computer Science*, 2020, pp. 21–33.

[24] J. Pavlicek and L. Kristoufek, "Nowcasting Unemployment Rates with Google Searches: Evidence from the Visegrad Group Countries," *PLoS One*, vol. 10, no. 5, p. e0127084, May 2015.

[25] P. McCullagh, "Generalized linear models," *Eur. J. Oper. Res.*,

vol. 16, no. 3, pp. 285–292, Jun. 1984.
[26] F. McLoughlin, A. Duffy, and M. Conlon, "Characterising domestic electricity consumption patterns by dwelling and occupant socio-economic variables: An Irish case study," *Energy Build.*, vol. 48, no. July 2009, pp. 240–248, May 2012.
[27] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. New York: Springer-Verlag New York, 2006.
[28] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent.," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.
[29] D. Cook, *Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI*, First Edit. O'Reilly Media, Inc, 2016.
[30] J. Friedman, T. Hastie, and R. Tibshirani, "glmnet: Lasso and elastic-net regularized generalized linear models." 2019.
[31] S. B. Kotsiantis, "Decision trees: a recent overview," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, Apr. 2013.
[32] C. J. Breiman L., Friedman J. H., Olshen R. A., and Stone, "Classification and Regression Trees." Wadsworth International Group, 1984.
[33] G. Stiglic, S. Kocbek, I. Pernek, and P. Kokol, "Comprehensive Decision Tree Models in Bioinformatics," *PLoS One*, vol. 7, no. 3, p. e33812, Mar. 2012.
[34] T. M. Therneau and E. J. Atkinson, "An Introduction to Recursive Partitioning Using the RPART Routines," Wadsworth, 2015.
[35] C. Strobl, J. Malley, and G. Tutz, "An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests.," *Psychol. Methods*, vol. 14, no. 4, pp. 323–348, 2009.
[36] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.
[38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY: Springer New York, 2009.
[39] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, no. DEC, 2013.
[40] Brandon Greenwell, Bradley Boehmke, and Jay Cunningham, "Generalized Boosted Regression Models." 2019.
[41] M. N. Wright and A. Ziegler, "ranger : A Fast Implementation of Random Forests for High Dimensional Data in C++ and R," *J. Stat. Softw.*, vol. 77, no. 1, 2017.
[42] T. Manning, R. D. Sleator, and P. Walsh, "Biologically inspired intelligent decision making," *Bioengineered*, vol. 5, no. 2, pp. 80–95, Mar. 2014.
[43] K. Chen and L. A. Kurgan, "Neural Networks in Bioinformatics," in *Handbook of Natural Computing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 565–583.
[44] A. Ng, "Sparse Autoencoder," in *CS294A Lecture notes*, 2011, pp. 1–19.
[45] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv:1212.5701*, Dec. 2012.
[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[47] D. Falbel *et al.*, "R Interface to 'Keras.'" 2019.
[48] J. S. Marron, M. J. Todd, and J. Ahn, "improving generalisability," *J. Am. Stat. Assoc.*, vol. 102, no. 480, pp. 1267–1271, Dec. 2007.
[49] X. Qiao, H. H. Zhang, Y. Liu, M. J. Todd, and J. S. Marron, "Weighted Distance Weighted Discrimination and Its Asymptotic Properties," *J. Am. Stat. Assoc.*, vol. 105, no. 489, pp. 401–414, Mar. 2010.
[50] B. Wang and H. Zou, "Distance Weighted Discrimination (DWD) and Kernel Methods," *R package version*, vol. 1. 2018.
[51] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[52] R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl, and A. C. P. L. F. de Carvalho, "Effectiveness of Random Search in SVM hyper-parameter tuning," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, vol. 2015-Septe, pp. 1–8.
[53] N. Salari, S. Shohaimi, F. Najafi, M. Nallappan, and I. Karishnarajah, "A Novel Hybrid Classification Model of Genetic Algorithms, Modified k-Nearest Neighbor and Developed Backpropagation Neural Network," *PLoS One*, vol. 9, no. 11, p. e112987, Nov. 2014.
[54] T. R. Hoens and N. V Chawla, "Imbalanced Datasets: From Sampling to Classifiers," in *Imbalanced Learning*, T. I. of E. and E. Engineers, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013, pp. 43–59.
[55] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3121–3124.
[56] T. Fawcett, "ROC Graphs : Notes and Practical Considerations for Researchers," *ReCALL*, vol. 31, no. HPL-2003-4, pp. 1–38, 2004.
[57] D. de Ridder, J. de Ridder, and M. J. T. Reinders, "Pattern recognition in bioinformatics.," *Brief. Bioinform.*, vol. 14, no. 5, pp. 633–47, Sep. 2013.
[58] C. Cortes and M. Mohri, "Confidence intervals for the area under the ROC curve," *Adv. Neural Inf. Process. Syst.*, vol. 17, pp. 305–312–312, 2005.
[59] X. Robin *et al.*, "pROC: an open-source package for R and S+ to analyze and compare ROC curves," *BMC Bioinformatics*, vol. 12, no. 1, p. 77, Dec. 2011.
[60] R Development Core Team, "R: A language and environment for statistical computing." R Foundation for Statistical Computing, Vienna, Austria, 2008.
[61] W. Sayeh and A. Bellier, "Neural networks versus logistic regression: the best accuracy in predicting credit rationing decision," pp. 1–28.
[62] Haibo He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[63] A. P. Piotrowski and J. J. Napiorkowski, "A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling," *J. Hydrol.*, vol. 476, pp. 97–111, Jan. 2013.