STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, June 2025, pp 2412–2433.
Published online in International Academic Press (www.IAPress.org)

An Application of Ensemble Stacking in Machine Learning to Predict Short-term Electricity Demand in South Africa

Claris Shoko [1,*], Caston Sigauke [2], Katleho Makatjane [1]

[1] *Department of Statistics, University of Botswana, Gaborone, Botswana*
[2] *Department of Mathematical and Computational Sciences, University of Venda, Thohoyandou, South Africa*

Abstract The massive increase in the collected data and the need for data mining and analyses has prompted the need to improve the accuracy and stability of traditional data mining and learning algorithms. This study proposes a robust stacking-ensemble algorithm for predicting the hourly electricity demand in South Africa. The structure of the proposed model is in two layers: the base model and the meta-model. Four machine learning models, that is, the gradient boosting machine (GBM), the deep neural network (DNN), the generalised linear model (GLM), and the random forest (RF), make up the base models. Output from the base models is integrated using ensemble stacking to form the meta-model. The stacking-ensemble (SE) model predicts South Africa's hourly electricity demand. The performance of the models is tested in different forecasting horizons. The prediction performance of the stacking-ensemble model is compared with the prediction performance of each of the base models using the root mean square error (RMSE), the mean absolute error (MAE), and the mean square error (MSE). In addition, the Giacomini-White test is used to identify the dominant model. Results showed that the RF model produced the most accurate predictions in all the forecasting horizons. The order of dominance is as follows: RF $>$ SE $>$ GBM $>$ GLM. Thus, RF demonstrates the highest predictive capability, dominating the other models. The stacking-ensemble model produced the second most accurate results, with its results in the shortest forecasting horizon almost equal to that of the RF model. Thus, in this context, the stacking ensemble performs better than 3 of the 4 meta models. The proposed model produces a reasonable and accurate prediction of hourly electricity demand, which is strategically significant in planning and formulating electricity load-shedding strategies in South Africa or any other country.

Keywords Deep neural network; electricity demand; gradient boosting machine; random forest; stacking-ensemble; Electricity demand; South Africa.

AMS 2010 subject classifications 62-07, 62M45

DOI: 10.19139/soic-2310-5070-2170

1. Introduction

Effective management of energy systems requires accurate prediction of energy demand (load) over some time. To enable real-time planning, system administrators use automated short-term forecasting methods. In unregulated energy markets, such forecasts are also important for market participants to support energy trading [36]. Short-term electricity demand forecasting plays a significant role in power system planning, including economic scheduling of generating capacity, scheduling of fuel purchases, and power system management. Accurate electricity forecasting is important to the residential sector and a major contributor to peak loads in most electricity systems. Overestimating electricity demands misleads planners and wastes resources with expensive expansion plans. Such overestimation also increases operating costs since electricity, unlike other energy sources, cannot be stored on a large scale [29]. However, underestimation of electricity demands will result in failures and shortages.

*Correspondence to: Claris Shoko (Email: shokoc@ub.ac.bw). Department of Statistics, University of Botswana, Gaborone, Botswana.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

The hourly electricity demand load is associated with complexities that statistical models fail to capture because of the assumptions of these models. Some of these assumptions include, among others, normality and linearity. However, hourly electricity demand load has complex nonlinear trends with non-normal density plots characterised by heavy tails. Hence, traditional time series forecasting approaches, including Auto-regressive Integrated Moving Average (ARIMA) and Exponential Smoothing, among others, are not the best models for handling nonlinear data since they assume that time series data should be linear [16].

Machine learning (ML) approaches provide the best performance in forecasting nonlinear trends in a given data set [5]. Coupled with deep learning algorithms, machine learning methods have attracted much attention in time series and forecasting literature because of their ability to handle complexities in time series data and, in particular, can handle big data sets. Unlike ordinary time series models, machine learning and deep learning models do not have any limitations and, hence, are widely applied to the prediction of complex time series trends. However, single forecasting may not adequately address the linear and nonlinear problems that characterise hourly electricity demand load [35]; it is, therefore, crucial to derive a model that electricity system developers will use to track, predict, and forecast hourly, daily, weekly, monthly, etc., electricity demand load in real-time.

Time series forecasting is a vast and active research area that has attracted significant interest from various fields. Because of this, much literature has concentrated on methods that can produce accurate forecasts in various real-world scenarios [14]. To achieve desired, accurate results and/or increase the accuracy of obtained results, literature generally points to two main approaches: (1) developing and proposing new forecasting models. (2) Hybridising existing forecasting models. Hybridisation is used because no single, comprehensive model can simultaneously capture all the patterns in the data. The complexity of real-world systems, with unpredictable mixed patterns, cannot be handled by a single model, and there is no doubt about that. Combining different models is one of the most popular solutions proposed in the literature, aiming to take the strength of individual models in pattern modelling and recognition applied in many time series forecasting articles. Ensemble forecast models have proved to be the best-performing models compared to the single forecast models. Published studies are numerous per the various models that researchers came up with. Because of this, many surveys cover machine learning for electricity time series forecasting studies. Recent advances in the field include deep learning (DL) models, which significantly outperform their traditional ML counterparts in terms of performance [18].

Short-term forecasting has a superior impact on the safety and financial implications of the energy network. Since the energy sector is stochastic and uncontrollable, the current study aims to improve the prediction and forecasting accuracy of hourly electricity demand load by combining a deep neural network with a generalised linear model and integrating it with a random forest. Determining whether a linear or nonlinear underlying process generated a studied series is often difficult. Whether one method is more efficient than another for out-of-sample predictions makes it difficult for forecasters to choose the right technique for their specific situation [22]. Typically, various models are evaluated, and the one with the most accurate results is selected. However, the final chosen model is not necessarily the best for further use due to several potentially influencing factors, such as sampling variance, uncertainty, and structural change, to mention a few. Combining different methods normally simplifies the model selection problem without additional effort. Second, time series are rarely linear or nonlinear in the real world. This often includes linear and nonlinear patterns. If this is the case, neither deep neural networks, generalised linear models, gradient boosting machines, nor random forests can be adequate in modelling and forecasting time series, especially hourly electricity demand load used in this study, since the DNN can have vanishing and exploding gradients during backpropagation, which leads to slow convergence or instability in training and memorising training data instead of generalising patterns, which now leads to poor performance on new unseen data. At the same time, the generalised linear model assumes a linear relationship between predictors and response variables. If the true relationship is highly nonlinear, GLMs may not capture it effectively. Furthermore, [37] outlined that GLMs might not perform well when dealing with complex relationships or interactions between variables, as they are inherently less flexible than other models. Although random forests are designed to mitigate over-fitting, they can still be prone to it, especially if the number of trees in the forest is too high and is sensitive to noisy data, which may affect their ability to generalise well to new, unseen data.

The GBMs have several hyperparameters that need to be tuned for optimal performance. Finding the right combination of hyperparameters is challenging and often requires extensive experimentation. This is because

the computation of the GBM is expensive, especially when dealing with large datasets such as hourly electricity demand load or a high number of trees in the ensemble. This makes it impractical for real-time or resource-constrained applications and makes it too sensitive to outliers (i.e., extreme values) in the training data, leading to skewed predictions. Combining DNN, GLM, and RF models will lead us to accurately model complex hourly electricity demand load structures. By harnessing the unique capabilities of each model type, this combined approach not only enhances predictive performance but also offers greater flexibility and interpretability in understanding electricity consumption patterns. Third, it is almost universally agreed in the forecasting literature that no single method is best in every situation. This is largely because a real-world problem is often complex, and any single model may not be able to capture different patterns equally well.

1.1. Literature review

Previous studies have generally performed simulated predictions based on individual machine learning models, showing their respective superiority. Although a single forecast model can improve forecast accuracy by adjusting parameters and selecting forecast factors in forecasting, the single model has model structure uncertainty and is difficult to adapt to different basins [25]. Numerous studies have shown that combining multiple single forecast models to build an ensemble forecast model can effectively exploit the advantages of different models and improve the reliability and accuracy of runoff forecasts [34]; [33]. Ensembles are usually used to overcome three problems associated with base learning algorithms: the statistical problem, the computational problem, and the representational problem [32], especially when complex nonlinear patterns characterise the time series data. The main aim of ensemble techniques is to develop a meta-classifier, a combination of multiple classifiers created from the same data to improve performance [9, 28].

Stacking is a popular ensemble learning technique that effectively mitigates bias and variance by combining weaker models to create a stronger one and has gained widespread use in machine learning. Stacking, also known as a stacked generalisation or stacking ensemble, is an ensemble machine learning technique that combines the predictions of multiple base models (also called learners or base learners) to improve the overall predictive performance of a model. Stacking aims to leverage the strengths of different models by allowing them to ”vote” on the final prediction, which often results in more accurate and robust predictions than individual models. The basic idea behind stacking is to train several diverse base models on the same dataset and then use the predictions of these base models as input features to a higher-level model called the meta-learner or blender. This meta-learner is responsible for learning how to combine the predictions of the base models best to make the final prediction. It learns from the errors or discrepancies of the base models and tries to correct them, resulting in a more accurate and robust final prediction.

Burger et al. [5] used a stacking-ensemble learning method by considering a weighted linear combination of forecasts from multiple sub-models. Sujan et al. [35] used a stacking ensemble for short-term electricity consumption forecasting using random forests, Long Short Term Memory, and Deep Neural Networks as base models. They ensembled the base models using Gradient Boosting and Extreme Gradient Boosting.

Lu et al. [20] used a stacking-ensemble approach to predict daily runoff to the Fuchun River Reservoir in the Qiantang River basin. They proposed a two-layer model comprising the random forest model, adaptive boosting (AdaBoost), and extreme gradient boosting (XGB) as the base models. Their results showed a significant outperformance of the proposed model compared to the base models.

Buildings use a lot of energy, which causes problems for the environment. It’s crucial to predict how much energy will be used to boost efficiency and make smart choices [8]. The authors used clustering to look at energy use patterns, finding groups for low, high, and weekend use, which showed how people behave. The study used several machine learning methods such as artificial neural networks, K-nearest neighbour, decision trees, random forest, extreme gradient boosting, and gradient boosting trees. A stacking ensemble approach with a genetic algorithm improved the prediction accuracy.

Chen et al. [6] introduced a novel stacking ensemble forecast model, TimeGAN-SEFM, aimed at boosting ultra-short-term PV power forecasting accuracy. The developed model uses a dual-layer feature-weighted clustering algorithm to categorise data by weather types and uses a particle swarm optimisation-based BP neural network to enhance model diversity. Evaluated with data from a 20 MW PV power station in North China, TimeGAN-SEFM

demonstrated significant improvements in forecasting accuracy, facilitating better alignment between PV energy supply and demand.

Fekih and Challouf [11] proposed an integrated approach to optimise grid management, combining photovoltaic (PV), wind, and grid energies to reduce costs and improve sustainability. The integrated approach focused on a scheduling algorithm using mixed integer programming (MIP) for dynamic resource allocation and employed a stacking algorithm for accurate forecasting, achieving an RMSE of less than 0.1. The authors argued that the proposed modelling framework enhances grid management for better renewable energy utilisation and sustainability.

The study by Yang et al. [39] introduced a modified stacking ensemble learning approach for short-term wind power forecasting, enhancing traditional models through improved tree-based and neural network learners. The authors argued that the proposed method reduced MAPE from 8.3% to 7.5% in 15-minute predictions, demonstrating superior accuracy across various conditions.

This study develops a stacking-ensemble algorithm for forecasting hourly electricity demand in South Africa. The base models are the gradient boosting machine (GBM), the generalised linear model (GLM), the deep random forest model (RF), and the deep neural network (DNN) model. These base models can handle complex nonlinear patterns in a time series, and their combined effort helps improve the prediction power. This study differs from previous studies in that the performance of fitted models is analysed at different forecasting horizons. In addition, explanatory variables are ranked according to their relative influence on the electricity demand. In addition to the use of performance metrics, root mean square error (RMSE), mean absolute error (MAE), etc., the Giacommini-White test is used to rank the models in terms of their predictive capability.

1.2. Contribution and research highlights

1.2.1. Contribution Based on the literature survey in Section 1.1, the contribution of the study is that the study proposes a novel stacking ensemble algorithm that combines different base models for an improved accuracy level, accurately forecasting day-ahead hourly predictions while embedding inherent complexity within electricity demand.

1.2.2. Research highlights The highlights of this study are:

- A novel stacking-ensemble algorithm is introduced in this paper for predicting hourly electricity demand using South African data. A two-layer structure is given which is: the base model layer (GBM, DNN, GLM, RF) and the meta-model layer formed through ensemble stacking of base model outputs.
- The performance of the proposed stacking-ensemble model is tested across various forecasting horizons. Evaluation metrics include RMSE, MAE, and MSE including the use of Giacomini-White test for evaluating the predictive capabilities of the models. Comparison with individual base models reveals that the RF model consistently produces the most accurate predictions.
- The stacking-ensemble model emerges as the second most accurate after the RF model, with results in the shortest forecasting horizon closely rivalling the model. Notably, the stacking ensemble outperforms three of the four base-models.
- The South Africa power utility company is currently facing supply constraints. The proposed model provides reasonable and accurate predictions of hourly electricity demand, holding strategic importance for planning and devising electricity load-shedding strategies in South Africa and other countries with similar challenges.

The rest of the paper is organised as follows. The methodology is discussed in Section 2. The empirical results are presented in Section 3, and the discussion of the results is given in Section 4. The conclusion is given in Section 5

2. Materials and methods

This section of the study presents the methods and procedures followed. Data from ESKOM (The South African Power Utility) is used for analysis in this study. From the data, we extracted the electricity demand load from the 1^{st}

of April 2019 to the 7^{th} of November 20, giving us a total of 40369 observations. The minimum hourly electricity demand is 587.1, with a maximum of 2375.9. The average electricity demand in South Africa is 1518.7 per hour, approximately equal to the median (1501.1). All the analysis is done using the R package "h2o". The package has the following features (Source: https://cran.r-project.org/web/packages/h2o/index.html):

- Uses data-distributed and parallelised Java-based algorithms for the ensemble.
- All training and data processing is performed in the high-performance H2O cluster rather than R memory.
- Supports regression and binary classification.
- Multi-class support in development.

2.1. Machine learning algorithms

Machine learning (ML) is the scientific study of algorithms and statistical models computer systems use to perform a specific task without being explicitly programmed. Learning algorithms in many applications that we make use of daily. Every time a web search engine like Google is used to search the internet, one of the reasons it works so well is because of a learning algorithm that has learnt how to rank web pages. These algorithms are used for various purposes like data mining, image processing, predictive analytics, etc., to name a few. The main advantage of machine learning is that once an algorithm learns what to do with data, it can do its work automatically [21]. In this study, four machine learning algorithms are used. Namely, Stochastic Gradient Boosting, convolutional deep neural network, generalised linear model, and Random Forest. Furthermore, the prospect of the vast applications of machine learning algorithms has been made.

2.1.1. Stochastic gradient boosting The terms "gradient boosting machine" and "gradient tree-boost" were initially coined by Friedman when he first introduced gradient boosting in his work [12]. Gradient boosting is a machine learning technique for solving classification and regression problems [27]. This method incrementally constructs weak predictive models by optimising an arbitrary differentiable function. The statistical framework of gradient boosting can be described as an optimisation problem aimed at minimising model errors through a step-by-step addition of weak learners using a gradient descent approach [12, 27].

Traditional gradient descent aims to minimise parameters like covariate coefficients or neural network weights by evaluating loss or error and adjusting the weights accordingly [13]. In gradient boosting, weak learners in the form of decision trees are employed instead of these parameters. A parameterised tree is added to the model, which decreases error and residual losses by adjusting the tree's parameters in the direction indicated by the gradient [13]. The gradients help identify errors within these weak learners.

One notable drawback of gradient boosting is its greedy nature, which can lead to overfitting on the training data [13, 15]. One variation of gradient boosting is stochastic gradient boosting (SGB), where a random subsample of the training dataset is selected without replacement [12, 13]. The general formulation of SGB can be found in Equation (1).

$$F(x) = \sum_{m=1}^{M} \beta_m h(x; \gamma_m), \tag{1}$$

where $h(x; \gamma_m) \in \mathcal{R}$ are functions of x with parameters γ_m and β_m which limit over fitting [13, 15].

2.1.2. Deep Neural Network Our network is based on the convolutional DNN defined by [17]. Our proposed neural network architecture is shown in Figure 1. This architecture consists of 2 convolutional layers with max pooling and 1 fully connected layer at the end. Each layer uses a rectified linear unit as a nonlinear transformation. Three of the convolutional layers have, in addition, max pooling. We refer the reader to [17].

We adapt the above generic architecture for localisation. Instead of using a softmax predictor as a last layer, we use a regression layer which generates $DNN(x; \Theta) \in \Re^N$, where Θ are the network parameters, and N is the total number of hours a day. Since the network output has a fixed dimension, we then predict hourly electricity demand load of a fixed size $N = d \times d$. After being resized to the monthly time series, the resulting binary pattern represents one or several patterns: it should have a value of 1 at a particular month if this month lies within the bounding box of a time series of a given class and 0 otherwise.

The network is trained by minimising the L_2 error for predicting a ground truth time series $m \in [0,1]^N$ for a month x

$$\min_{\Theta} \sum_{(x,m)\in D} \left\| (\text{Diag}(m) + \lambda I)^{1/2} \left(DNN(x;\Theta) - m \right) \right\|_2^2, \tag{2}$$

the sum ranges over a training set D of months containing bounding hours and weeks, represented as binary patterns. Since our base network is highly non-convex and optimality cannot be guaranteed, it is sometimes necessary to regularise the loss function by using varying weights for each output depending on the ground truth. The intuition is that most patterns are small relative to the time series size, and the network can be easily trapped by the trivial solution of assigning a zero value to every output. To avoid this undesirable behaviour, increasing the weight of the outputs corresponding to non-zero values in the ground truth is helpful by a parameter $\lambda \in \Re^+$. If λ is chosen small, then the errors on the output with ground truth value 0 are penalised significantly less than those with one, encouraging the network to predict non-zero values even if the signals are weak.

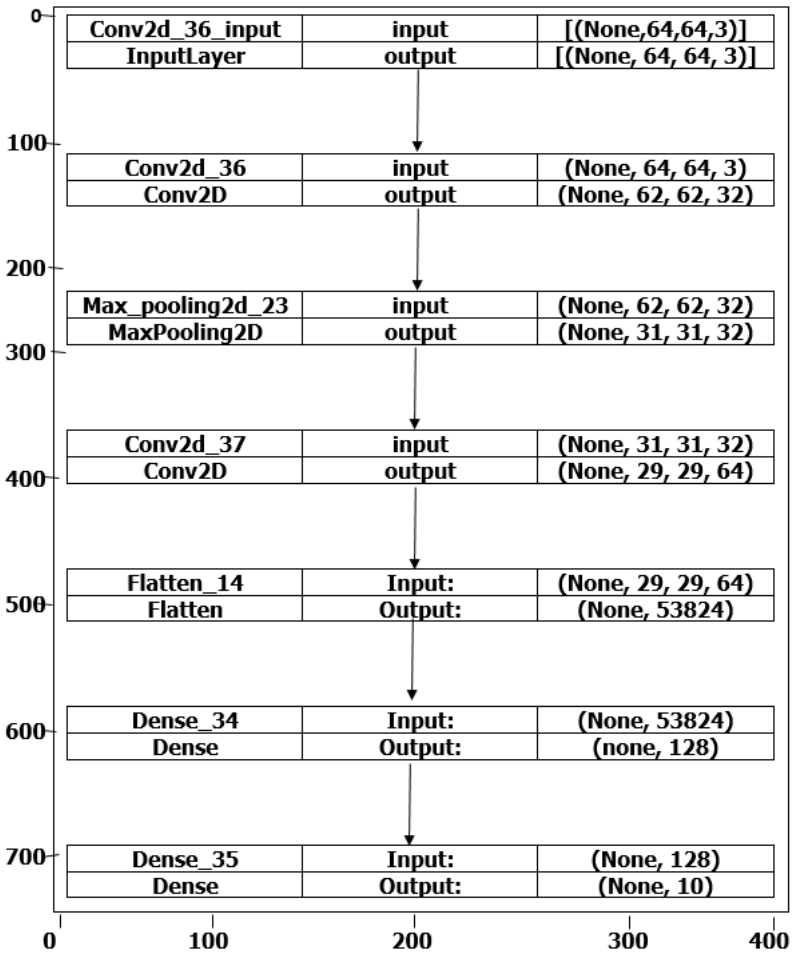

Figure 1. Proposed Convolutional Deep Neural Network. Source: Author Own Computation

2.1.3. Generalised linear model The generalised linear models provide a common approach to various response modelling problems. Normal, Poisson and binomial responses are commonly used, but other distributions can also be used. Apart from specifying the response, GLMs also need a link function to set, allowing further flexibility

in the modelling. The GLM can be fitted using a common procedure, and a mechanism for hypothesis testing is available [10]. The term "generalised" linear model (GLIM or GLM) refers to a larger class of models popularised by [24]. In these models, the response variable y_t is assumed to be from an exponential family distribution with mean μ_t, which is assumed to be some (often nonlinear) function of $x_t^T\beta$. Some researchers would call these "nonlinear" because μ_t is often a nonlinear function of the covariates, but [24] consider them to be linear because the covariates affect the distribution of y_t only through the linear combination $x_t^T\beta$.

Following [1] and [19], we assume that there is a discrete time-series data Y_t with $t \in \mathbb{N}$. Next, we model the conditional mean $E\left(Y_t \mid F_{t-1}\right)$ from discrete time-series data, for example, λ_t and $t \in \mathbb{N}$. Then, the general GLM model for modelling discrete time series data is given in Equation (3).

$$g\left(\lambda_t\right) = \beta_0 + \sum_{k=1}^{p} \beta_k \tilde{g}\left(Y_{t-ik}\right) + \sum_{l=1}^{q} \alpha_l g\left(\lambda_{t-jl}\right) + \eta^T X_t, \tag{3}$$

where $g : R^+ \to R$ is the link function, and $\tilde{g} : N_0 \to R$ is a transformation function, a vector parameter $\eta = \left(\eta_1, \cdots, \eta_{tr}\right)^T$. In $GLM\nu_t = g\left(\lambda_t\right)$ is called the linear predictor. The regression can be used for the past time response variables that are defined as $p = \{i_1, \cdots, i_p\}$ and i is integer $0 < i_1 \cdots < i_p < \infty$, with $p \in N_0$. In the GLM model for discrete time series data, regressing observed lag $Y_{t-i}, \cdots, Y_{t-ip}$ is possible. The same analogy with lag in observation defined Q is $Q = \{j_1, \cdots, j_q\}$ for j is $\{j_1 <, \cdots < j_p < \infty\}$ with j is the integer and $q \in N_0$ for the regressor variable on the lag for the conditional mean $\lambda_{t-j1}, \cdots, \lambda_{t-jp}$.

The model in Equation 3 depends on the link function used. Here is an example of the identity link function, $g\left(x\right) = \bar{g}\left(x\right) = x$. Then $p = \{1, \cdots, p\}$, $\{1, \cdots, q\}$ and $\eta = 0$. When $\eta = 0$, the covariates have no effect, then model in Equation 3 will then be

$$\lambda_t = \beta_0 + \sum_{k=1}^{p} \beta_k \left(Y_{t-k}\right) + \sum_{l=1}^{q} \alpha_1 \left(\lambda_{t-1}\right). \tag{4}$$

Equation 3 assumes that Y_t has a Poisson distribution. To assess the model's goodness developed by [7], we then use the integral probability transform (PIT) displayed as a histogram. The model is better if the histogram shape is approaching uniform distribution. Besides, according to [23] another way to assess the model's habit is to use a selection criteria model such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC), that is, the model with the smallest AIC and BIC values is the best.

2.1.4. Random Forest Random forests (RFs) are an ensemble learning method for classification and regression problems [2]. Random Forest is a collection of decision trees growing in randomly selected feature space subspaces. The principle of RFs is to combine a set of binary decision trees (Breiman's CART – Classification And Regression Trees [3]), each of which is constructed using a bootstrap sample coming from the learning sample and a subset of features (input variables or predictors) randomly chosen at each node. Thus, in contrast to the CART model building strategy, an individual tree in RF is built on a subset of learning points and on subsets of features considered at each node to split on. Moreover, trees in the forest are grown to maximum size, and the pruning step is skipped.

Given an ensemble of classifiers $h_1(x), h_2(x), \cdots, h_k(x)$, and with the training set drawn at random from the distribution of a random vector Y, X, define the margin function as

$$Mg(X, Y) = av_k I(h_k(X) = Y) \max_j av_k I(h_k(X) = j), \tag{5}$$

where $I\left(\bullet\right)$ is the indicator function. The margin measures the extent to which the average number of votes at X, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalisation error is given in equation (6).

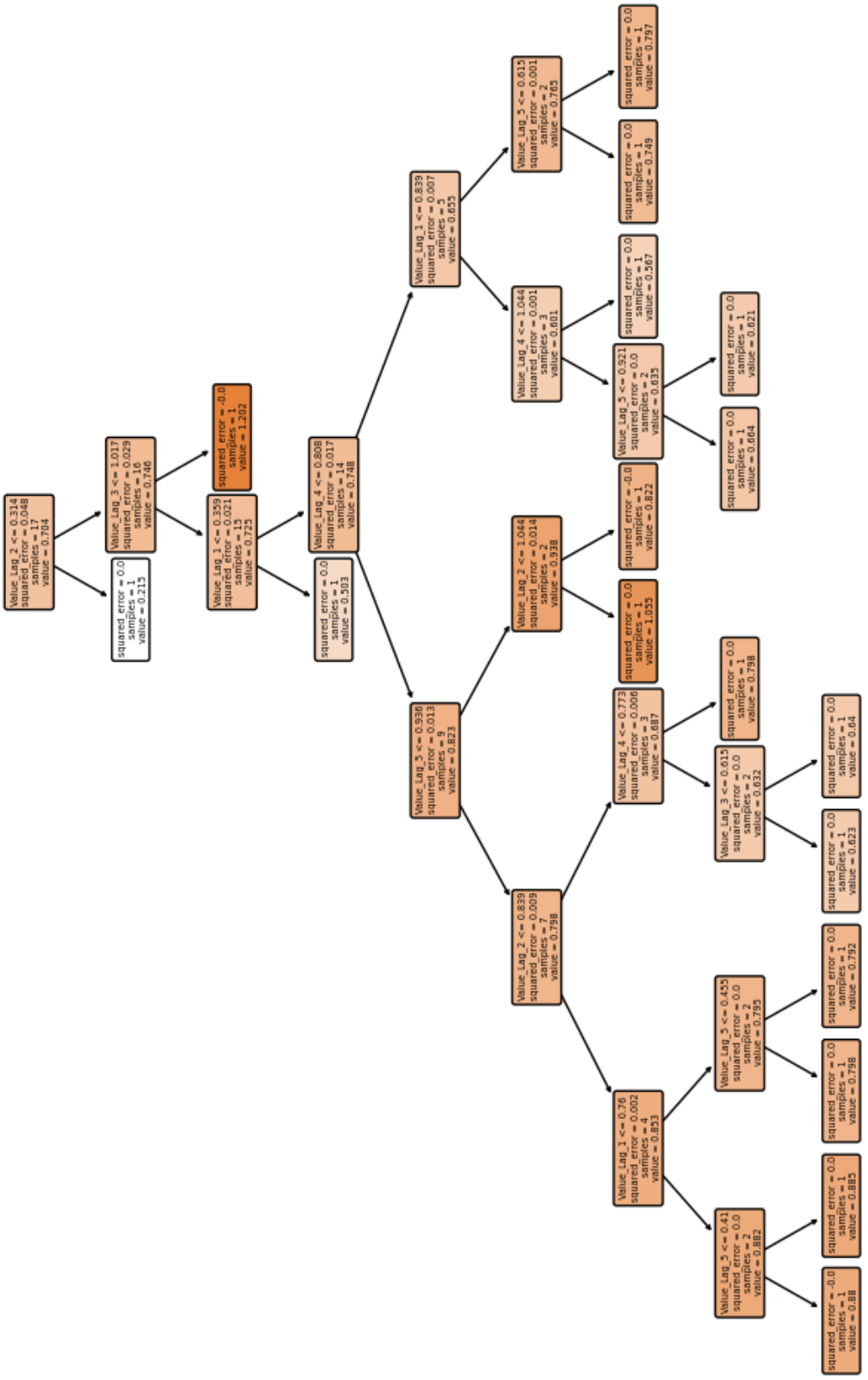

Figure 2. Proposed Random Forest. Source: Authors own computation

$$PE^{(}*) = P_{X,Y}\left(Mg\left(XY\right) < 0\right) \tag{6}$$

where the subscripts X, Y indicate that the probability is over the X, Y space.

In random forests, $h_k\left(X\right) = h\left(X, \theta_k\right)$. For a large number of trees, it follows from the strong law of large numbers and the tree structure that: As the number of trees increases, for almost all sequences $\theta_1, \cdots, PE^*$ converges to

$$P_{XY} = \left(P_\theta\left(h\left(X, \theta\right) = Y\right) \max_j {}_y P\theta\left(h\left(X, \theta\right) = j\right) < 0\right) \tag{7}$$

The convergence of Equation 7 explains why random forests do not over-fit as more trees are added but produce a limiting value of the generalisation error. Figure 2 represents our proposed random forest with one estimator and five random states.

Table 1 gives a summary of the strengths and weaknesses of the proposed machine learning algorithms including their computational efficiencies.

Table 1. Strengths and weaknesses of the proposed machine learning algorithms.

Algorithm	Strengths	Weaknesses
Gradient Boosting Machines	1. High predictive accuracy 2. Handles mixed data types well 3. Robust to outliers	1. Prone to overfitting if not tuned 2. Computationally intensive 3. Requires careful parameter tuning
Deep Neural Networks	1. Can model complex nonlinear relationships 2. Excellent for unstructured data like images 3. Highly scalable and flexible	1. Requires large amounts of data 2. High computational cost and time 3. Difficult to interpret and explain
Generalized Linear Model	1. Easy to interpret and explain 2. Efficient with smaller datasets 3. Provides coefficients that show relationships	1. Assumes linear relationships among variables 2. Limited flexibility for complex data patterns 3. Poor performance in high-dimensional spaces
Random Forest	1. Robust to overfitting 2. Handles missing values effectively 3. Good performance across various tasks	1. Less interpretable than linear models 2. Requires more computational resources 3. May not perform well on imbalanced datasets

2.2. Stacking ensemble learning method

In recent years, performance improvements have made ensemble models more relevant. It is obtained from various tasks such as classification or regression problems [30]. It consists of methods that combine different learning models to improve the results of each model. Usually, two phases are employed. In the first phase, a set of base learners is obtained from training data, while in the second phase, the learners obtained in the first phase are combined to produce a unified prediction model. Thus, multiple forecasts based on the different base learners are constructed and combined into an enhanced composite model superior to the base individual models. Integrating all good individual models into one improved composite model generally leads to higher accuracy. Figure 3 shows the schema of the Stacking-Ensemble algorithm used for forecasting electricity demand in South Africa.

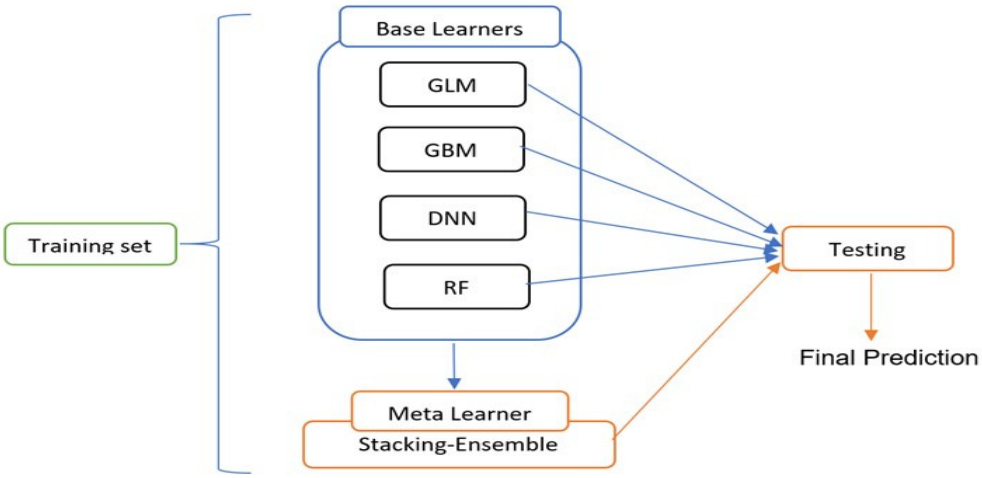

Figure 3. Schema for the Stacking-Ensemble Algorithm for predicting electricity demand in South Africa.

The most used and well-known basic ensemble methods are bagging, boosting and stacking. This paper uses a stacking approach since it is the most suitable for the base learners' problem considered in this work. In the following section, we will specify which learning algorithms have been used in the proposed scheme, as shown in Figure 3. Therefore, the stacking-ensemble approach to time series forecasting is when heterogeneous base models are trained and tested on the dataset in parallel or stacking. The assumption is that prediction from the base learners is weak. In stacking, the algorithm takes the outputs of the base learners as input and attempts to learn the best way of combining the heterogeneous predictions to give a better output. In this study, we use the Generalised Linear model (GLM), Random Forest (DRF), Gradient Boosting Machine (GBM), and Deep Neural Network (DNN) model as the base learners. (See Figure 3). The stacking-ensemble approach to time series forecasting is when heterogeneous base models are trained and tested on the dataset in parallel or stacking. The assumption is that prediction from the base models is weak.

In stacking, the algorithm takes the outputs of the base models as input and attempts to learn the best way of combining the heterogeneous predictions to give a better output. In this chapter, we use the Generalised Linear model (GLM), Random Forest (DRF), Support Vector Machine (SVM) using different kernel functions, Gradient Boosting machine (GBM), and Deep Neural Network (DNN) model as the base models. Predictions from these base models are combined using either the Linear Quantile regression Averaging (LQRA) or the Weighted Averaging methods. A comparison of the performance of these models is done using the Key Performance Indicators KPIs). The KPIs are discussed at the end of this section. The meta-model combines predictions from these base models using either the stacking-ensemble approach. A comparison of the performance of these models is done using the Key Performance Indicators KPIs).

Ensemble learning methods which linearly combine the predictions of multiple models are generally referred to as stacking or stacked generalisation methods and can often outperform any one of the trained sub-models (see, for instance, [4] and [38]). To produce a multivariate prediction $\hat{y}\sum \in \Re^m$, [5] emphasised that the ensemble model is defined as the weighted sum of each prediction $\hat{y}\sum 0 \in \Re^m$, from each submodel s, as given by equation (8).

$$\hat{y}\sum_{s=1}^{M}\theta_s\hat{y}_s \tag{8}$$

with variable $\theta_s \in \mathbf{R}$, the weighting coefficient of sub-model, s where M is the number of sub-models and subscript $s = 1, \cdots, M$ indexes the coefficients, i.e. $[\theta_1, \cdots, \theta_M = \theta \in \mathbf{R^M}]$. Note that we are not calculating the weighted mean of the sub-models. Therefore, we do not require that the values of the weighting coefficients sum to 1 or that the individual weights are positive.

Unlike [5], who employed the ordinary least square method, we now use a maximum likelihood with ℓ_2 regularisation (Ridge) approach for learning the weighting coefficients θ. By solving a quadratic optimisation problem, we can identify the weighting coefficients that minimise the error between the observations and the weighted sum of the submodel predictions. The optimisation problem and stacking ensemble model at time step t are given by equation (9).

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \sum_{j=1}^{m} \left(y_{i,j} - \sum_{s=1}^{M} \theta_s \hat{y}_{s,i,j} \right)^2 + \lambda \sum_{s=1}^{M} \theta_s^2 \tag{9}$$

$$\hat{y} \sum, t = \sum_{s=1}^{M} \theta_s^* \hat{y}_{s,t} \tag{10}$$

with variables $\theta^* \in \mathbf{R^M}$, the optimal weighting coefficients given by $y_i \in \mathbf{R^m}$, the i^{th} observed multivariate response $\hat{y}_{s,i} \in \mathbf{R^m}$, the i^th prediction from sub-model $s, \hat{y} \sum, t \in \mathbf{R^M}$, the ensemble model prediction at t, and $i = 1; \cdots ; N$, where N is the number of data samples used for training and m is the length of yi. Subscript $j = 1; \cdots ; m$ indexes the j^{th} response. Lastly, λ is a weighting term for the regularisation penalty.

2.3. Evaluation metrics

This study subsection discusses the error metrics for model selection and assessing the model performance. Unlike [22], who used only four metrics, we use five in this study.

2.3.1. Mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |(y_i - \hat{y}_i)| \tag{11}$$

2.3.2. Root mean square error RMSE signifies an absolute error.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{12}$$

2.3.3. Root mean squared log error RMSLE represents a relative error, whereas.

$$\text{RMSLE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\log[1 + y_i] - \log[1 + \hat{y}_i])^2} \tag{13}$$

2.3.4. Mean residual deviance The mean residual deviance (MRD) is the deviance divided by the degrees of freedom, which measures the average lack of fit of the model per data point. A smaller mean residual deviance indicates a better fit of the model to the data. The formula for the mean residual deviance is typically:

$$\text{MRD} = \frac{\text{Deviance}}{\text{Degrees of freedom}} \tag{14}$$

2.3.5. Predictive accuracy The Giacomini-White test formally tests whether the differences in forecast accuracy are statistically significant. If the test yields significant results, it provides evidence that one model is statistically superior to the other, which is useful for selecting the best forecasting model in a given context.

3. Empirical results

In this study, hourly electricity demand data from the 4^{th} of January 2019 to the 25^{th} of July 2023 is used for analysis. The data is split into two sets: training and testing sets. The stacking-ensemble algorithm is developed from the base models: deep neural networks, deep random forests, gradient boosting machines, and generalised linear models. Further detail is presented below.

3.1. Exploratory data analysis

The average RSACD for the period under study was 1518.7 with a standard deviation of 234.9, a median of 1501.1, a skewness of 0.261, and a kurtosis of 2.625. This shows that the data is non-normal and tends to have outliers. Figure 4 presents a univariate data analysis for the variable of interest: the daily contracted electricity demand for South Africa. The time series, density, normal Q-Q, and box and whisker plots are presented in this figure.

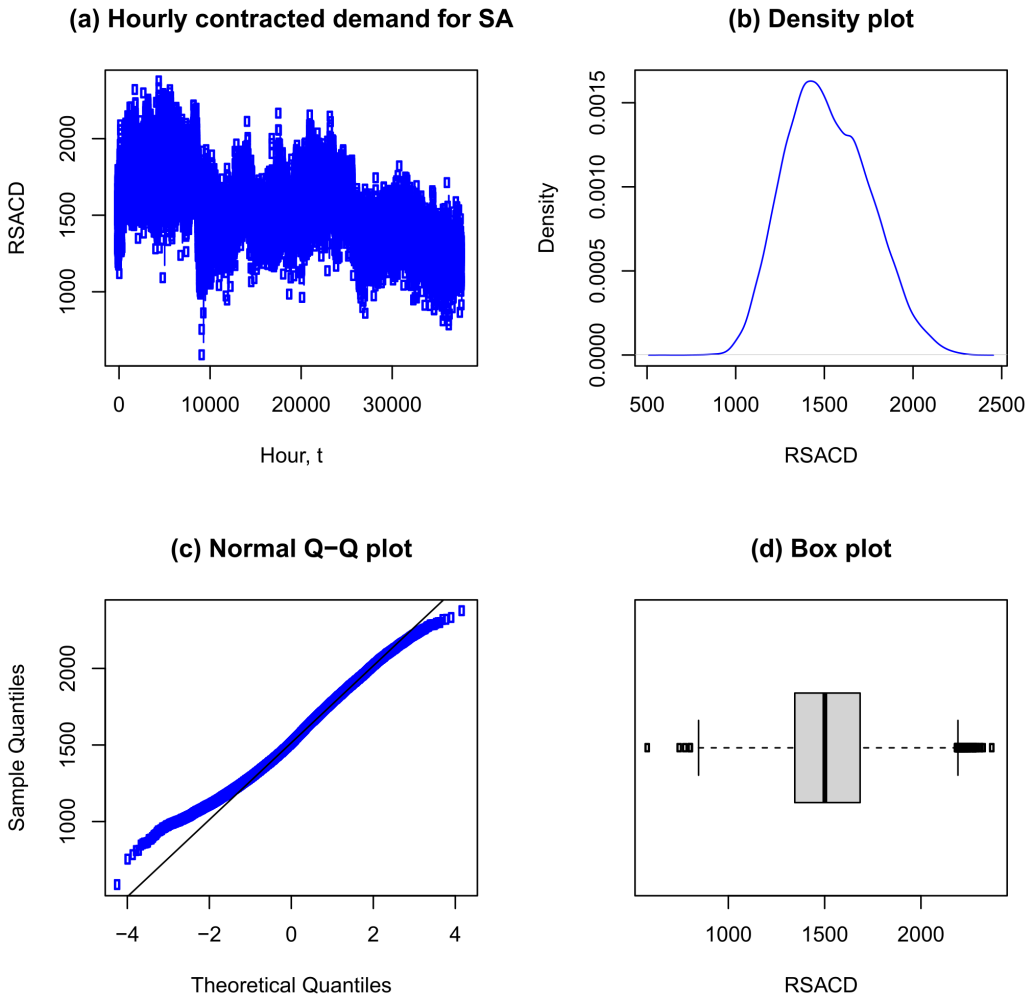

Figure 4. Univariate analysis for the hourly contracted electricity demand for South Africa.

The density plot shows that the hourly electricity demand data is not normally distributed. The normal Q-Q plot also shows some deviations from normality. The time series plot shows a complex nonlinear pattern. We extract a

nonlinear trend for the RSACD data by fitting the cubic smoothing spline function represented by the equation

$$n(t) = \sum_{t=0}^{n}(y_t - f(t))^2 + \lambda \int \{f''(t)\}^2 dt, \tag{15}$$

where λ is the smoothing parameter that is equal to -0.005473617 predicted using the generalised cross-validation (GCV=14212.13), and the nonlinear trend is illustrated in Figure 5. This trend helps to figure out the time series trend.

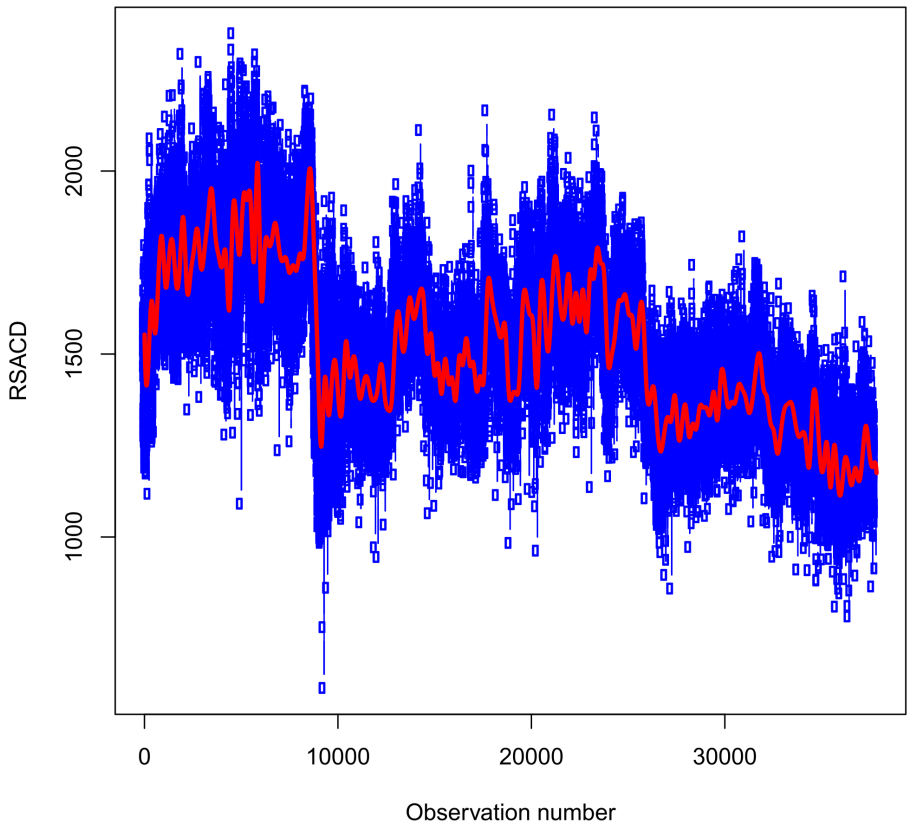

Figure 5. Hourly Contracted Electricity Demand superimposed with a nonlinear trend variable from a regression spline.

The nonlinear trend for the hourly contracted electricity demand time series data shows irregular cycles instead of regular seasonality. The peaks are sometimes smaller or larger. Thus, modelling this time series data requires the inclusion of the lag features to allow our forecasters to react dynamically to the changing conditions. This study includes lag variables as explanatory variables to allow varying amounts of recent history into the forecast.

3.2. Results

Hourly contracted electricity demand data is divided into two sets: the training set and the testing set. The training set has 37,465 observations, and the testing set has 376 observations. The stacking-ensemble algorithm is divided into two stages. In the first stage, the base models discussed in section 2 are trained and tested. For each of the fitted

models, the following variables were included: hourly electricity demand (response variable) and the explanatory variables are the hour of the day, Day (Monday, Tuesday,..., Sunday), Month (January, February,..., December), Winter (1 if winter and 0 otherwise), lag1, lag 2, lag 5 and lag 7. The inclusion of lags improved the performance of the fitted models.

In the next stage, we combine predictions from the base models using the stacking-ensemble approach and then test the performance of the final model on the test set. The analysis is performed using the 'h2o' package in R. The detailed analysis is presented in the subsections that follow. For each fitted model, 5-fold cross-validation is used to verify each machine learning model. Table 2 presents variable importance for the fitted models.

Table 2. Percentage Influence of each variable on the fitted base models.

	Percentage			
variable	**GBM**	**GLM**	**Random F**	**DNN**
noltrend	0.823901	0.622439	0.705184	0.203874
lag7	0.093493	0.144439	0.086965	0.093623
lag5	0.043197	0.083332	0.063969	0.091409
Hour	0.018315	0.060492	0.03007	0.174157
lag2	0.016877	0.063279	0.040066	0.072874
lag1	0.00175	0.012735	0.020126	0.056545
Month	0.001453	0.00044	0.036985	0.113188
MLR	0.000928	0.010584	0.005987	0.068239
Day1	0.000076	0.000869	0.007994	0.042383
Winter	0.000009	0.001391	0.002653	0.083707

The nonlinear trend variable has the highest percentage influence on all the fitted models. In most models (GBM, GLM, and RF), lag 7 and 5, respectively, are ranked as the 2nd and 3rd in importance. The results show that shifting the values 7 hours into the future is between 8.6% and 14.4% influential in forecasting the hourly electricity demand, and the percentage influence of shifting the values 5 hours ahead is between 4.3% and 9.14% depending on the model.

3.2.1. Comparative analysis of fitted models In order o come up with the best model that predicts hourly contracted electricity demand for South Africa, we used different forecasting horizons, that is, $\{(train : test)\} = \{(95\% : 5\%), (97\% : 3\%), (99\% : 1\%)\} = \{(35942 : 1899), (36737 : 1104), (37465 : 376)\}$. The performances of the fitted for each forecast horizon are presented in Table 3. The fitted models are the base models (GBM, GLM, RF, and DNN) and the stacking ensemble (SE). We used the mean absolute error (MAE), root mean square error (RMSE), root mean square log error (RMSLE), and mean residual deviance (MRD) to evaluate the performance of the models on the test set.

At the $\{(train : test)\} = \{(95\% : 5\%), (97\% : 3\%)\} = \{(35942 : 1899), (36737 : 1104)\}$ forecast horizons, the random forest model has the best performance. However, the SE model performs almost the same as the random forest model at the $\{(train : test)\} = \{(99\% : 1\%)\} = \{((37465 : 376)\}$ forecasting horizon. This means that the SE model is the best for short-term electricity demand prediction, and for long-term forecasting, the random forest model performs best. The $\{(train : test)\} = \{(95\% : 5\%)\} = \{((35942 : 1899)\}$ forecasting horizon has the smallest error metrics compared to the other two. For further comparison, we use forecast horizons $\{(train : test)\} = \{(95\% : 5\%), (99\% : 1\%)\} = \{(35942 : 1899), (37465 : 376)\}$ for long-term forecasting and short-term forecasting, respectively. Figures 6 and 8 present plots of forecasts from the fitted base models superimposed on the observed data from the training test. In addition, the density plots for the fitted models are presented in Figure 7 and Figure 9 for the 1% and the 5% forecast horizons respectively. Plots help in giving a visual appeal of the performance of each fitted model on the test data sets.

Table 3. Model comparisons for three different forecasting horizons.

95% training ($n_{train} = 35942$) : 5% testing ($n_{test} = 1899$)				
Model	MAE	RMSE	RMSLE	MRD
GBM	55.54	71.82	0.047	5158.47
GLM	59.39	76.35	0.050	5829.70
RF	**52.66**	**68.86**	**0.045**	**4741.28**
DNN	58.24	75.14	0.050	5646.24
SE	54.33	69.95	0.046	4893.11
97% training ($n_{train} = 36737$) : 3% testing ($n_{test} = 1104$)				
GBM	57.08	73.50	0.048	5401.90
GLM	60.51	77.65	0.051	6029.39
RF	**53.41**	**70.23**	**0.046**	**4932.84**
DNN	59.79	76.90	0.050	5913.74
SE	55.31	71.75	0.047	5148.14
99% training ($n_{train} = 37465$) : 1% testing ($n_{test} = 376$)				
GBM	59.25	74.95	0.049	5616.82
GLM	61.23	77.58	0.050	6018.38
RF	55.01	**71.14**	**0.046**	**5061.57**
DNN	61.37	78.95	0.051	6232.75
SE	**54.68**	71.19	**0.046**	5067.42

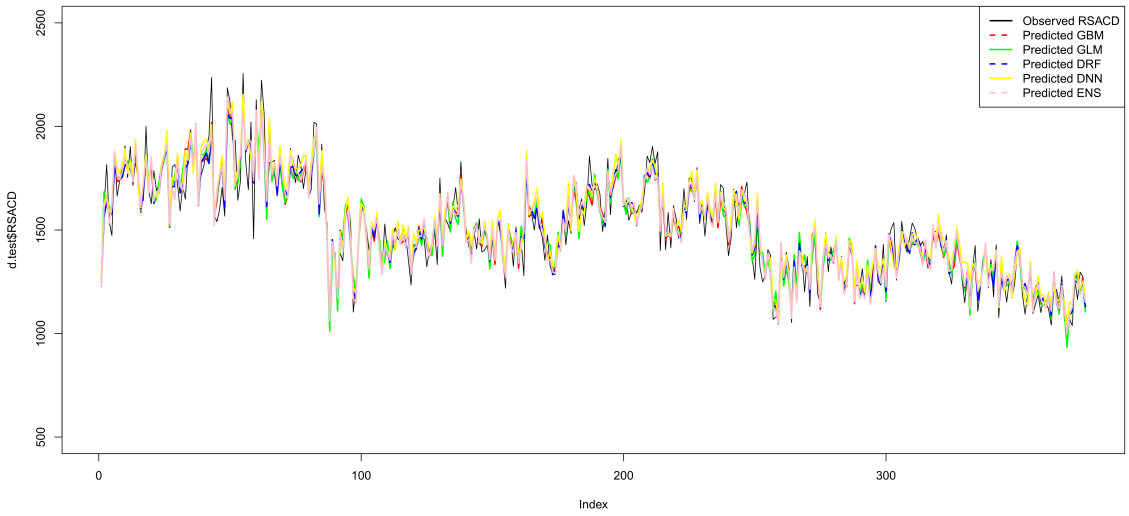

Figure 6. Short-term forecasts from fitted models superimposed on the observed set or test set ($n_{train} = 37465$): 1% testing ($n_{test} = 376$).

In Figure 6, forecasts from the Stacking ensemble give the best prediction of the test data set followed by the Random Forest model. This confirms the findings presented in Table 3. The density plots in Figure 7 show that the forecasts from the SE model are closer to the observed data set.

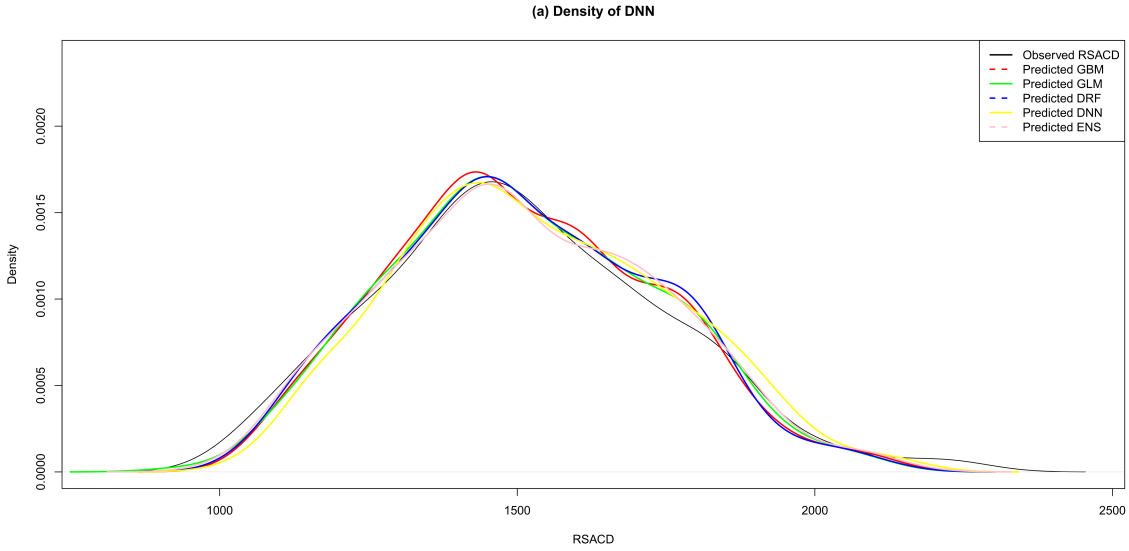

Figure 7. Density plots for the fitted models superimposed on the observed set or test set ($n_{train} = 37465$): 1% testing ($n_{test} = 376$).

In the long-term forecast horizon shown in Figure 8, the DNN and SE models failed to predict the lowest peak near the 600^{th} observation in the training set. Thus, compared to the RF model, the SE model is weaker in long-term predictions. The density plot in Figure 9 confirms the results.

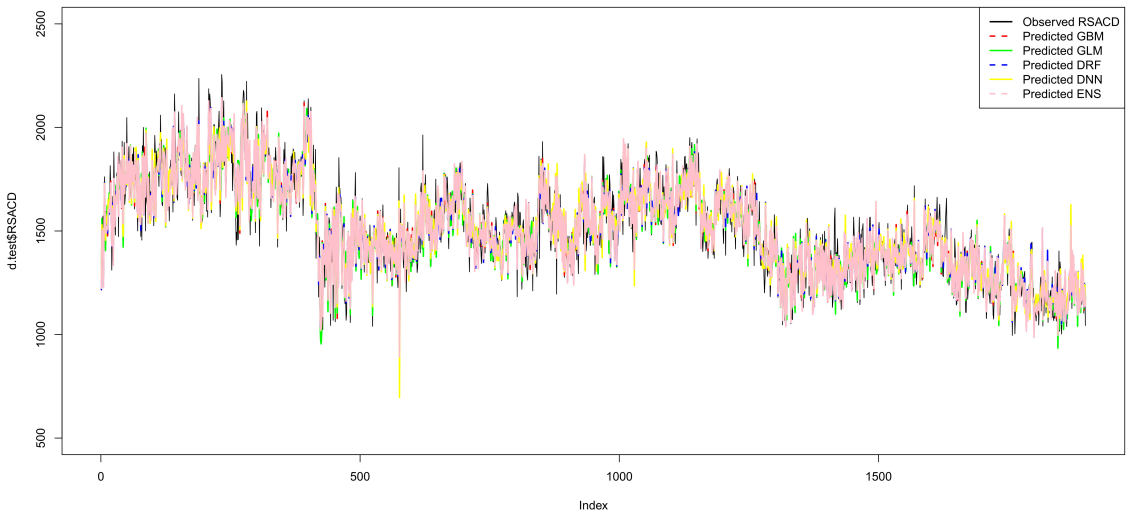

Figure 8. Long-term forecasts from fitted models superimposed on the observed set or test set ($n_{train} = 35942$): 5% testing ($n_{test} = 1899$).

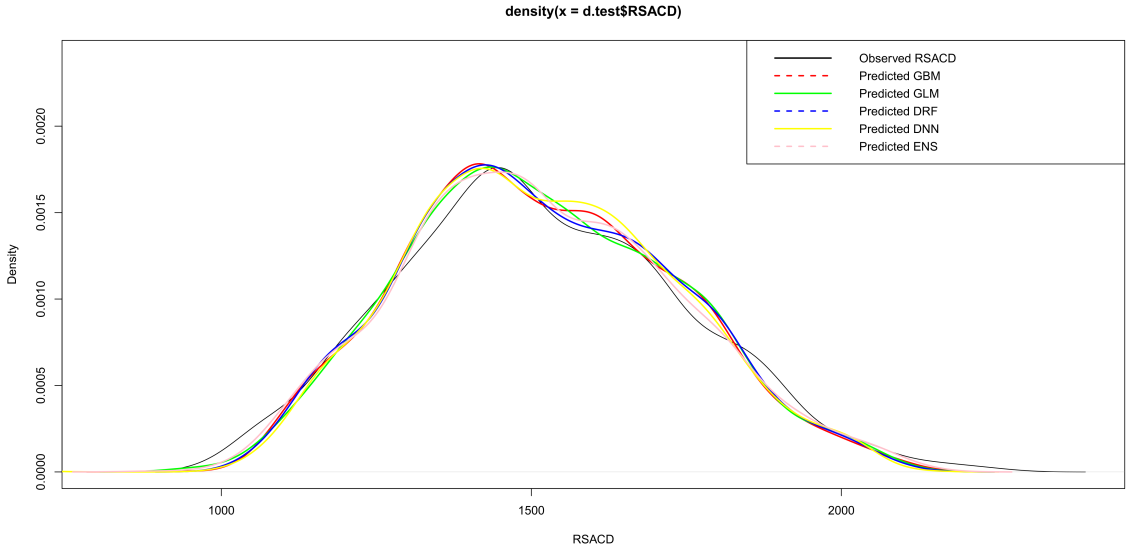

Figure 9. Density plots for the fitted models superimposed on the observed set or test set ($n_{train} = 35942$): 5% testing ($n_{test} = 1899$) .

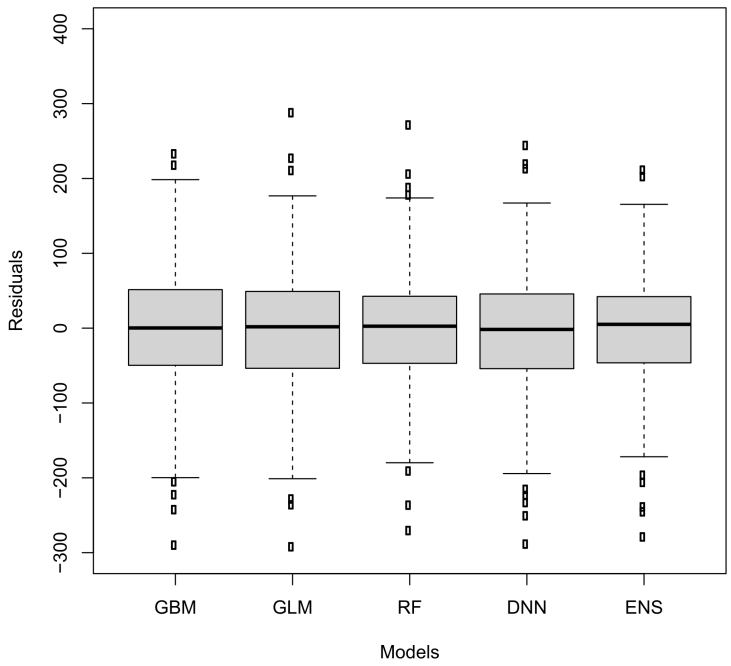

Figure 10. Box plots of the residuals of the models ($n_{train} = 37465$) : 1% testing ($n_{test} = 376$).

To test the effectiveness of the forecasting models, we present box plots illustrating the forecast error distributions for all the models that were fitted for the short-term and long-term forecasting horizons in Figure 10 and Figure 11

respectively. The visual representation suggests that RF exhibits the most tightly clustered error distribution when comparing models, implying that RF outperforms the other models and is the most favourable choice. However, the box plots for the longer time horizon have more outliers for all the fitted models.

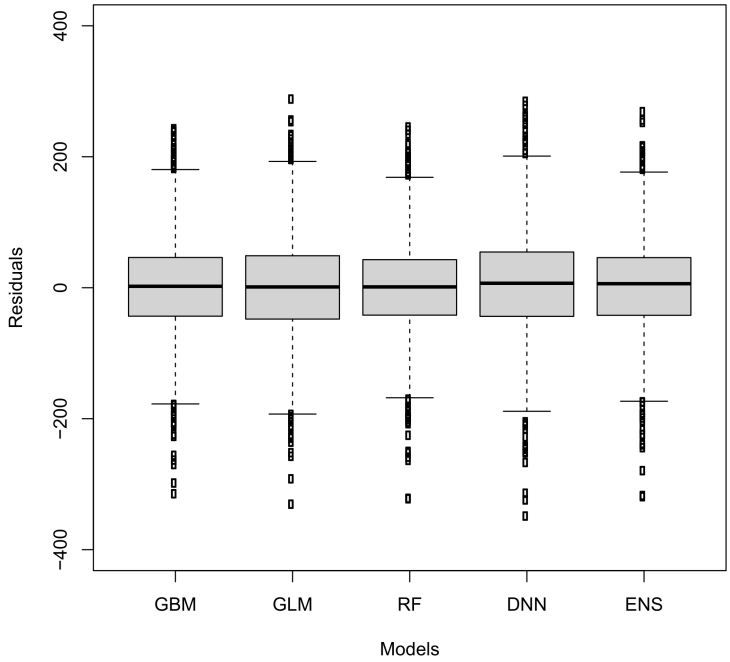

Figure 11. Box plots of the residuals of the models ($n_{train} = 35942$) : 5% testing ($n_{test} = 1899$).

Table 4. Model comparisons using the Giacomini-White test.

Giacommini-White test			
Null hypothesis	Test statistic	p-value	Result
GBM = GLM	3.57	0.059	Sign of mean loss is (-). GBM dominates GLM
GBM = RF	7.01	0.008	Sign of mean loss is (+). RF dominates GBM
GBM = DNN	0.658	0.417	Sign of mean loss is (-). GBM dominates DNN
GBM = SE	2.31	0.128	Sign of mean loss is (+). SE dominates GBM
GLM = RF	16.3	5.467 e^{-5}	Sign of mean loss is (+). RF dominates GLM
GLM = DNN	0.867	0.352	Sign of mean loss is (+). DNN dominates GLM
GLM = SE	6.47	0.011	Sign of mean loss is (+). SE dominates GLM
RF = DNN	6.63	0.010	Sign of mean loss is (-). RF dominates DNN
RF = SE	1.36	0.243	Sign of mean loss is (-). RF dominates SE
DNN = SE	3.70	0.054	Sign of mean loss is (+). SE dominates DNN

Note: The "Sign of the mean of the loss is (-)" is the sign of the mean of the forecast errors or loss function differences between the two models being compared. The loss function typically measures the difference between the predicted values and the actual observed values. If the mean of these differences is negative, it suggests that the first model has, on average, lower forecast errors (closer predictions to the actual values) than the second model. A negative mean indicates that the first model performs better in this case.

We also employed Giacomini-White (GW) tests to assess the models' statistical significance and predictive capabilities. The results of the GW tests are given in Table 4. To indicate dominance, we will use the notation $M_i > M_j$ for all $i \neq j$, signifying that model M_i dominates model M_j. Based on the information presented in Table 4, the order of dominance is as follows: RF $>$ SE $>$ GBM $>$ GLM. Consequently, RF demonstrates the highest predictive capability, as it dominates the other models.

Further comparison of the fitted models is presented using density plots for each of the fitted models. The density plots are presented in the Appendix section. The density plots show that the random forest model outperforms all the base models, including the stacking ensemble. However, the stacking ensemble is favourable compared to the deep neural network, gradient boosting, and the generalised linear model.

4. Discussion

This section of the study presents the discussion of the results and the limitation of the study.

4.1. General discussion

This study presents a novel approach to real-time electricity demand forecasting using the data from South African power utility ESKOM. We proposed a stacking ensemble algorithm combining the Gradient Boosting Machine, Generalised Linear, Random Forest, and Deep Neural Network models. The proposed models have been tested over three different forecasting horizons; $(training, testing) \in \{(95\%, 5\%), (97\%, 3\%), (99\%, 1\%)\}$. The results show that the longer the forecasting horizon, the better the performance of the models. The random forest model outperforms the GBM, GLM, DNN, and stacking ensemble models in the 5% and 3% forecasting horizons. The stacking ensemble model is the second-best-performing model in this context. However, the performance of the stacking ensemble model is almost similar to that of the Random forest model at the 1% forecasting horizon. This shows that the Stacking Ensemble performs better in short-term forecasting, but the Random Forest model is the best in long-term forecasting. For the longer forecasting horizon, the Giacomini-White test is used to test the performance of the models further, and the results show that the Random Forest model dominates all the other models, including the stacking ensemble. Most studies demonstrated that the stacking ensemble outperforms the meta-models [5] and [16]. This study demonstrated that the stacking ensemble performs the best in short-term forecasting horizons, with the Random Forest model producing more accurate performance as the forecast horizon gets wider.

The percentage influence of different variables on the hourly electricity demand for South Africa has been assessed. The variables include the constructed nonlinear trend, the most influential variable with over 70% influence for the GBM and the RF models; the lag 7 variable is the second most influential; the hour of the day and month also influence electricity demand.

4.2. Limitation

A limitation of this study is the use of data from a specific country, which is South Africa, in this case. The findings may not be that applicable to other areas because of differing characteristics of electricity demand due to the influencing variables such as climate, building design and energy policy, which affect energy consumption behaviours. Moreover, combining models such as deep neural networks, generalised linear models, random forests, and gradient-boosting machines through stacking ensemble techniques enhances predictive performance. However, this approach requires significant computational resources. Training multiple base models and a meta-model leads to long training times and increased memory usage, especially with large datasets such as hourly electricity demand used in this study or intricate model architectures. Finally, each model in the ensemble requires careful tuning of hyperparameters, which adds to the complexity and time required for model development. This process can be resource-intensive and may require extensive experimentation to identify optimal settings.

Some further research will, therefore, be comparative in various regions or include data from different countries to test whether the identified patterns persist. To overcome the limitations of the stacking algorithm, future studies should Utilise parallel processing capabilities to train multiple base models simultaneously. This can significantly

reduce training times and make better use of available computational resources. In addition, leveraging cloud-based platforms that offer scalable resources for model training, allowing for efficient handling of large datasets without local hardware constraints is another area of research that will be studied elsewhere.

5. Conclusion

This paper demonstrates the performance of base models (GBM, GLM, RF, DNN) and the meta-model (stacking ensemble) in models in forecasting hourly electricity demand for South Africa. The paper presents that although the stacking ensemble model performs better than the three baseline models, the random forest model is overall the best performer. The random forest model performs even better with longer forecasting horizons, while the stacking ensemble performs equally as the random forest model in the shortest forecast horizon. Results from this study assist electricity providers, end-users, and the South African government in addressing energy challenges in South Africa.

Acknowledgements

The authors sincerely thank the anonymous reviewers for their helpful comments and suggestions on this paper.

Author Contributions:

These authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement:

The analytic data can be downloaded from https://github.com/csigauke?tab=repositories

Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Networks
GBM	Gradient Boosting Machine
GLM	Generalised Linear Model
MAE	Mean Absolute Error
RF	Random Forest
RMSE	Root Mean Square Error
RMSLE	Root Mean Squared Logarithmic Error
MRD	Mean Residual Deviance
SE	Stacked Ensemble

REFERENCES

1. Bosowski, N. and Manolakis, D. *Generalised Linear Models for Count Times Series*. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017. https://doi.org/10.1109/ICASSP.2017.7952962.
2. Breiman, L. *Random Forests*. Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

3. Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. Classification and Regression Trees. Chapman and Hall, 1984.
4. Breiman, L. *Stacked Regressions*. Machine learning, vol. 24, no. 1, pp. 49–64, 1996. https://doi.org/10.1007/BF00117832.
5. Burger, E. M., and Moura, S. J. *Building Electricity Load Forecasting via Stacking Ensemble Learning Method with Moving Horizon Optimization*. UC Berkeley: Energy, Controls, and Applications Lab, 2015. Available at https://escholarship.org/uc/item/6jc7377f (Accessed on November 23, 2023).
6. Chen, C., Chai, L. and Wang, Q. *Research on stacking ensemble method for day-ahead ultra-short-term prediction of photovoltaic power*. Renewable Energy, vol. 238, 121853, 2025. https://doi.org/10.1016/j.renene.2024.121853
7. Christou V., Fakianos K. *Estimating and Testing Linearity for Non-Linear Mixed Poisson Autoregression Electronic*. Journal of Statistics, vol. 9, no. 1, pp. 1357–1377, 2015. https://doi.org/10.1214/15-EJS1044
8. Dostmohammadi, M., Pedram, M.Z., Hoseinzadeh, S., and Garcia, D.A. *A GA-stacking ensemble approach for forecasting energy consumption in a smart household: A comparative study of ensemble methods*. Journal of Environmental Management, vol. 364, 121264, 2024. https://doi.org/10.1016/j.jenvman.2024.121264
9. Dzeroski, S. and Zenko, B. *Is Combining Classifiers with Stacking Better than Selecting the Best One*? Machine Learning, 54, 255–273, 2004. https://doi.org/10.1023/B:MACH.0000015881.36452.6e.
10. Faraway, J.J. *Generalised Linear Models*. International Encyclopaedia of Education, Third edition, pp. 178–183, 2010. Elsevier
11. Fekih Hassen, W. and Challouf, M. *Long Short-Term Renewable Energy Sources Prediction for Grid-Management Systems Based on Stacking Ensemble Model*. Energies, vol. 17, no. 13, 3145, 2024. https://doi.org/10.3390/en17133145
12. Friedman, J.H. *Stochastic Gradient Boosting*. Computational Statistics and Data Analysis, vol. 38, pp. 367–378, 2002. https://doi.org/10.1016/S0167-9473(01)00065-2
13. Friedman, J.H. *Greedy Function Approximation: A Gradient Boosting Machine*. Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001. https://www.jstor.org/stable/2699986
14. Hajirahimi, Z., and Khashei, M. *Hybrid Structures in Time Series Modelling and Forecasting: A Review.* Engineering Applications of Artificial Intelligence, vol. 86, pp 83–106, 2019. https://doi.org/10.1016/j.engappai.2019.08.018.
15. Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics., Second Edition, 2009.
16. Jorjeta, G. Jetcheva, Mostafa, M., Wei-Peng, C. *Neural Network Model Ensembles for Building-level Electricity Load Forecasts*. Energy and Buildings, vol. 84, pp. 214–223, 2014. https://doi.org/10.1016/j.enbuild.2014.08.004.
17. Krizhevsky, A., Sutskever, I., and Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks*. Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017. https://doi.org/10.1145/3065386.
18. Kumar, R., Kumar, P. and Kumar, Y. *Analysis of financial time series forecasting using deep learning models*. Paper presented at 2021 11^{th} International Conference on Cloud Computing, Data Science, and Engineering (Confluence), 2021. https://doi.org/10.1109/Confluence51648.2021.9377158
19. Liboshick, T., Fakianos, K. and Fried, R. *tscount: An R Package for Analysis of Count Time Series Following Generalised Linear Models*. Journal of Statistical Software, vol. 82, no. 5, 2017. https://doi.org/10.18637jss.v082.i05.
20. Lu, M., Hou, Q., Qin, S., Zhou, L., Hua, D., Wang, X. and Cheng, L. *A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting*. Water, 15, 1265, 2023. https://doi.org/10.3390/w15071265.org/
21. Mahesh, B. *Machine Learning Algorithms: A Review*. International Journal of Science and Research (IJSR). vol. 9, no. 1, pp. 381–386, 2020.
22. Makatjane, K. and Moroke, N. *Predicting Extreme Daily Regime Shifts in Financial Time Series Exchange/Johannesburg Stock Exchange-All Share Index*. International Journal of Financial Studies, vol. 9, no. 2, 18, 2021. https://doi.org/10.3390/ijfs9020018
23. Makoni, T., Mazuruse, G., and Nyagadza, B. *International tourist arrivals modelling and forecasting: A case of Zimbabwe*. Sustainable Technology and Entrepreneurship, vol. 2, no. 1, 100027, 2023. https://doi.org/10.1016/j.stae.2022.100027
24. McCullagh, P. and Nelder, J.A. *Generalised Linear Models*. Second Edition, Chapman and Hall, London., 1989. http://dx.doi.org/10.1007/978-1-4899-3242-6.
25. Moges, E., Demissie, Y., Larsen, L. and Yassin, F. *Review: Sources of Hydrological Model Uncertainties and Advances in Their Analysis*. Water, vol. 13, no. 1, 28, 2021. https://doi.org/10.3390/w13010028
26. Mohanad, S., Al-Musaylh, R.C., Deo, J.F. and Adamowski, Y. L. *Short-term Electricity Demand Forecasting with MARS, SVR and ARIMA Models Using Aggregated Demand Data in Queensland, Australia*. Advanced Engineering Informatics, vol. 35, pp. 1-16, 2017. https://doi.org/10.1016/j.aei.2017.11.002.
27. Mpfumali, P., Sigauke, C., Bere, A. and Mulaudzi, S. *Day Ahead Hourly Global Horizontal Irradiance Forecasting—Application to South African Data*. Energies, vol. 12, 3569, 2019. https://doi.org/10.3390/en12183569.
28. Opitz, D. and Maclin, R. *Popular Ensemble Methods: An Empirical Study*. Journal of Artificial Intelligence Research, vol. 11, pp. 169–198, 1999. https://doi.org/10.1613/jair.614
29. Qiu, J., Yang, H., Dong, Z. Y., Zhao, J. H., Meng, K., Luo, F. J., and Wong, K. P. *A Linear Programming Approach to Expansion Co-planning in Gas and Electricity Markets*. IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3594–3606, 2015. https://doi.org/10.1109/TPWRS.2015.2496203
30. Ren, Y., Zhang, L. and Suganthan, P.N. *Ensemble Classification and Regression: Recent Developments, Applications, and Future Directions*. In IEEE Computational Intelligence Magazine, vol. 11, no. 1, pp. 41–53, 2016. https://doi.org/10.1109/MCI.2015.2471235
31. Rochyati, I., Syafitri, U.D. and Sumertajaya, I.M. *Study of Forecasting Models on Foreign Tourists: Visit and International Passengers' Arrival without and with Covariates*. [thesis]. IPB University, 2019.
32. Sikora, R. *A Modified Stacking Ensemble Machine Learning Algorithm using Genetic Algorithms*. In Artificial Intelligence: Concepts, Methodologies, Tools, and Applications, pp. 395–405, IGI Global, 2017. doi.org10.4018/978-1-5225-1759-7.ch016

33. Shoko, C. and Sigauke, C. *Short-term Forecasting of COVID-19 using Support Vector Regression: An application using Zimbabwean Data*. American Journal of Infection Control, vol. 51, no. 10, pp. 1095–1107, 2023 https://doi.org/10.1016/j.ajic.2023.03.010
34. Shoko, C., Sigauke, C. and Njuho, P. *Short-term Forecasting of Confirmed Daily COVID-19 Cases in the Southern African Development Community Region*. African Health Sciences, vol. 22, no. 4, pp 534–550, 2022. https://dx.doi.org/10.4314/ahs.v22i4.60.
35. Sujan Reddy, A., Akashdeep, S., Harshvardhan, R., Sowmya, K.S. *Stacking Deep Learning and Machine Learning Models for Short-Term Energy Consumption Forecasting*. Advanced Engineering Informatics, 52, 101542, pp. 1–10, 2022. https://doi.org/10.1016/j.aei.2022.101542.
36. Taylor, J. W. *Triple Seasonal Methods for Short-term Electricity Demand Forecasting*. European Journal of Operational Research, vol. 204, no. 1, 139–152, 2010. https://doi.org/10.1016/j.ejor.2009.10.003
37. James, G., Witten T., Hastie, T. and Tibshirani, R. *An Introduction to Statistical Learning*. Springer publication, 2017.
38. Wolpert, D.H. *Stacked Generalization*. Neural networks, vol. 5, no. 2, pp. 241–259, 1992.
39. Yang, Y., Li, Y., Cheng, L. and Yang, S. *Short-Term Wind Power Prediction Based on a Modified Stacking Ensemble Learning Algorithm*. Sustainability, vol. 16, no. 14, 5960, 2024. https://doi.org/10.3390/su16145960