IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 23 April 2024, accepted 29 April 2024, date of publication 9 May 2024, date of current version 21 May 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3399390

APPLIED RESEARCH

Optimal Electricity Load Interruption Based on Time Series Classification With Super Learner and Feature Filtering

SOLOMON OLUWOLE AKINOLA[1], PETER OLUKANMI[1], QING-GUO WANG[1], AND TSHILIDZI MARWALA[2]
[1]Institute for Intelligent Systems, University of Johannesburg, Johannesburg 2006, South Africa
[2]Department of Office Rector, United Nations University, Tokyo 150-8925, Japan
Corresponding author: Solomon Oluwole Akinola (oluwolea@uj.ac.za)

ABSTRACT Load-shedding is vital for managing electrical power shortages and avoiding grid collapse. However, excessive electricity demand poses an imminent threat to the overall stability of power grid system (PGS) and its ability to run safely and reliably. Load-shedding strategies can be complicated and inadequate to manage electrical power system efficiently. The study proposed a data-driven load-shedding time series classification (TSC) technique employing a heterogeneous ensemble super learner (eSL) to categorize load-shedding based on contributing features. The model investigated challenges with binary classification while using a multidimensional time series for South Africa's hourly load-shedding stages in MW collected from PGS data. Considering that load-shedding is planned and predicted based on contributing features, we use these features as strong indicators to classify expected outcomes for load-shedding or no load-shedding. Validation tests for the suggested technique included the precision recall curve, the confusion matrix, the class likelihood ratio, the Brier skill scores and critical difference factor (CDF). Logistic regression (LR) produced the highest CDF average score, while support vector classifier (SVC) had the highest balanced precision (90.694%). The recursive feature elimination (RFE) model exhibited the most significant true negative and true positive counts, at 50.59% and 40.84%, respectively, and the highest proportion of valid classifications.

INDEX TERMS Ensemble, super learner, recursive feature elimination, time series classification.

NOMENCLATURE

X	Independent or predictor for $x_1, \ldots, x_n$ observations in dataset.
Y	Dependent variable for $y_1, \ldots, y_n$ dataset.
$P(Y\|X)$	The probability of $Y = 1$ for predictor variables X.
$E(Y\|X)$	Conditional expectation of Y given X.
n, N	The total number of observations or classes.
s	Split or branch node for routing to a left or right child node in a classification tree.
$lc(s), rc(s)$	Left and right child node when a split occur.
$f(s)$	A single decision tree single split feature.
$\theta(s)$	Threshold value.
l	Leaf node to store vote for a class.
τ	Number of leaves in a tree.
ι	Loss function.
α	Penalty/regularization parameter.
ρ_i	Percentage for misclassification.
$f(x), \hat{y}$	Approximation function.
i, j, k	Instance observation.
κ	Kernel function.
ϕ	Weight vector.
P_r, TP_{Rate}	Sensitivity or true positive rate.
R_r, TN_{Rate}	Specificity or true negative rate.
$F_1 score$	Harmonic mean.
$LR+$	Positive likelihood ratio.
k_n	Number of neighbors.
Obj	Objective function.

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

V_v	Validation set.
T_v	Training set.
χ	ESKOM dataset input.
$\hat{\Psi}_j$	Base learner.
Z	Prediction matrix.
$\hat{\Psi}_{SL}$	Ensemble super learner.
w_0	Intercept.
w_i	Coefficient weight.
v_j	Data sample.
p_i	Probability of each class.

I. INTRODUCTION

Electrical power is generated through a complex network of renewable and fossil fuel sources, leading to uncertainty in electricity supply to meet demand [1]. In the event of limited electricity production sources, electricity demand results in interruptions of the supply due to load-shedding. Excessive load-shedding from decreasing generating sources can lead to catastrophic grid system collapse [2], [3]. Various approaches based on machine learning (ML) have been investigated to balance electricity demand and generation. However, as the number of ML models that address load supply disruption continues to grow, there are limitations in the categorization task related to the disruption of the electric load supply based on developing characteristics. South Africa, like many developing countries in Africa, faces power shortages and irregular electricity supply. A load-shedding strategy by the South African energy agency ESKOM is in place to help with supply shortages, with daily load-shedding events varying from "stage 1" (about 1,000 MW) to "stages 8" (around 8,000 MW) [4].

Predictive ML techniques can help with systematic and effective load-shedding management [5]. ML algorithms are powerful in extracting insight from data, often performed with learners for a covariate task, predictive function, or causal impact [6]. Adopting Industry 4.0 is a complementary scientific approach that focuses on data analysis, computational intelligence, and the identification of indicators for decision-making. In this study, ML predictive classifiers identify discrete class labels using stacked heterogeneous aggregated learners, leading to an insightful classification. Feature engineering is indispensable in the ML pipeline, as it optimizes computation, improves performance, and limits noise or irrelevant features [7]. Recent studies have explored binary wrapper, grey wolf optimization, particle swarm optimization, stability criteria, wrapper-based feature selection, adaptive teaching and learning for feature selection optimization, and its application in electricity optimization, dimensionality reduction, and numerosity reduction [8], [9], [10], [11], [12], [13].

A promising ML approach for load-shedding is the ensemble approach. The ensemble technique aggregates the knowledge of weak learners. The ensemble technique results in higher convergence, robustness to outliers, and optimal regularization compared to a single predictor [6]. To validate, aggregating learners from numerous options requires consistent sampling, as it is improbable to find in advance the most appropriate combination for a specific task [6]. The ensemble super learner is a proven technique [6], [14], [15]. The eSL is a data-adaptive approach with proven use cases and confirms significance in maximum likelihood estimates. The aggregation of learners evolved from the stacked generalization model [16]. Further experimentation demonstrates the capability of stacking predictors for meta-learning [6], [14], [15], [17], [18], [19], [20], [21], [22], [23], [24], with variations extending eSL functions for a specific set of tasks. eSL solves some of the bottlenecks common with individual models, such as an expectation space that is overly large for the quantity of available training data, an analytical challenge that guarantees a global optimum, and an individual model that lacks a well-defined approximation for model distribution outcomes. This study focuses on stacked eSL for load-shedding task [6], [17]. Details of the schema are established in section II.

The study explores the classification of electrical load supply interruptions using a stacked heterogeneous learner. It extends the weighted information gain measure by combining heterogeneous techniques in base learner classifiers. The following are the key contributions of this paper:

1) Identify biclass electricity load-shedding, which is strongly associated with a meta-learner classification technique based on layered eSL. The load-shedding method was determined using ESKOM data. The best load-shedding option was determined through hourly categorization of contributing ESKOM features data.
2) The load-shedding contributing indicator for the meta-learners' load-shedding classification from feature representation in this research takes into account the base learner from the stacked heterogeneous ML learner aggregate using the logistic regression approach. The time-dependent class of load-shedding indicates that it can ensure the steady functioning of significant load distribution action.
3) The ESKOM PGS emergency load-shedding TSC task offers a guided tool to improve load-shedding decision-making and overall PGS efficiency when necessary. Load-shedding is a response to high power demand or low generation. Severe load-shedding can result in revenue losses and poor industrial output. The goal is to avoid or reduce load-shedding sufficiently so that the PGS can be deployed optimally.

The rest of the paper is organized as follows: Section II emphasizes associated concepts, techniques and discusses super learner modeling and prediction. Section III describes and discusses the result. Finally, Section IV concludes the paper.

II. RELATED THEORIES AND METHODS

A. BASE LEARNER

In the construction of eSL model, we used logistic regression, decision tree classifier, support vector classifier, extreme gradient boosting classifier, k-nearest neighbors classifier, AdaBoost classifier, bagging classifier, random forest classifier, and extremely randomized trees classifiers as base learners and logistic regression for the meta-learner. The process starts at the root node in logistic regression and continues through all base models. This approach assesses the model with k-fold cross-validation. An array is formed by stacking out-of-fold forecasts. Each base model is adapted to the training dataset, and the resulting estimate is kept for the meta-analysis.

1) LOGISTIC REGRESSION

The logistic regression (LR) technique applies a linear sequence of input data for binary classification tasks. LR possesses a suite of theoretical foundations with significant predictive accuracy in widely competitive domain classification tasks [25]. In LR, the independent variable X as $(x_1, \ldots, x_n)$ defines the dependent variable Y as a logit fit multiple linear regression. LR requires a functional form $P(Y|X)$ for the probability of $Y = 1$ for predictor variables X. In the following notation from [26], (1) and (2) expresses the LR as:

$$P(Y = 1 \mid X) = \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}, \tag{1}$$

$$P(Y = 0 \mid X) = \frac{\exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}, \tag{2}$$

where w_0 is the coefficient intercept and coefficients weight for observations $w_1, \ldots, w_n$ is selected from a maximizing conditional likelihood. Equation (2) originates from (1), as the sum of the two probabilities should be one. LR is a probabilistic function applied to the negative categorization power and frequency change rates with very high correction performance [27].

2) DECISION TREE CLASSIFIER

Decision tree classifier (DTC) is a non-parametric technique based on a rule-defining scheme for target labels from feature inferencing. DTC has a modest implementation scheme but may result in overfitting. There are many variants of the DTC from well-known models such as chi-square automatic interaction detector (CHAID) [28], Iterative Dichotomize (ID3), Quinlan iteration (C4.5 and C5.0) [29], classification and regression trees (CART) [30]. Gini loss and entropy are significant tuning parameters in a classification task. The Gini estimates the value of a split, log loss, or entropy is the information gain. Equations (3) to (6) define the left split, right split, Gini, and entropy. The technique to create a decision tree begins with a random training sample from the training dataset. A decision tree consists of split and tree nodes. Each node s is a looping procedure and starts by randomly choosing sample variables from all available variables. The root node is built and assigned the sample data. The choice of the optimal split feature and threshold is based on the Gini or entropy criterion by dividing the node into two child nodes and moving to the associated subsets. The p represents the percentage of samples attributed to class i.

$$x \in lc\,(s) \Leftrightarrow x_{f\,(s)} < \theta\,(s)\,, \tag{3}$$

$$x \in rc\,(s) \Leftrightarrow x_{f\,(s)} \geq \theta\,(s)\,, \tag{4}$$

$$\text{Gini index}: G(E) = 1 - \sum_{i=1}^{n} p^2{}_i, \tag{5}$$

$$\text{entropy}: H\,(E) = -\sum_{i=1}^{n} p_i log p_i. \tag{6}$$

In an input space, the tree formulation [31] is built repeatedly. Every branch node is a branch (split) node. Branch makes a divide decision and sends the data sample x to either the left child node $lc(s)$ or the right child node $rc(s)$. When employing axis-aligned split options, the split rule is based on a single split feature $f(s)$ and a threshold value $\theta(s)$. If the value of x feature $f(s)$ is less than a threshold $\theta(s)$, it is routed to the left child node. Otherwise, it is directed to the right child node. All leaf nodes are in the branches. Leaf node l store votes for the classes $y^l = (y_1^l, \ldots, y_n^l)$, where n is the number of classes. The CART decision tree was adopted in the experimentation for the ESKOM data classification.

3) SUPPORT VECTOR CLASSIFIER

SVC maps input sequences to a high-dimensional space. SVC is a classifier capable of handling non-linear tasks. SVC is implemented with a hyperplane as decision boundaries. The outermost boundary defines the hyperplane [32]. SVC is a type of kernel support vector machine (SVM) for the classification task. For further information, see [33].

The kernel approach improves SVM by allowing kernel functions to solve optimization issues in a high-dimensional space. When utilizing SVM, training data is mapped into a new feature space using a kernel function. Then, SVM creates a considerable margin difference between training sets in the new feature space. Given the assigned series $(x_1, y_1, \ldots, x_n, y_n)$, x indicates the variables that constitute the covariates and $y \in \{-1, 1\}$ is the reaction, the value of the weight vector is ϕ. SVM uses a kernel function κ as defined in (7) to (9).

$$f(x) = \sum_{i=1}^{n} \phi_i \kappa(x_i, x) + b, \tag{7}$$

$$\sum_{i,j=1}^{n} \phi_i \phi_j \kappa(x_i, x_j) + \alpha \sum_{i=1}^{n} \rho_i, \textit{ and} \tag{8}$$

$$y_i f(x_i) \geq 1 - \rho_i, \tag{9}$$

ρ_i represents the percentage of incorrect categorization of x_i, and α is the penalty parameter for miss classification.The optimal hyperplane is a function of $\phi > 0$ and bias $b. f(x)$ is the function and translates the training vectors x into a higher dimensional space. Using the function $f(x)$, SVM computes a linear hyperplane that distinguishes the training data into a higher dimensional space. SVC has been applied in energy fault detection based on active power variants [34] and in large-scale image recognition problems [35], [36].

4) EXTREME GRADIENT BOOSTING CLASSIFIER

Chen and Guestrin [37] proposed an extension of gradient boosting called extreme gradient boosting (XGB) to include an objective function for scalable tree boost. XGB is a function of functions. The objective function includes the regularization term and the training loss. In (10) to (12),

$$f_i^{(t)} = \sum\nolimits_{n-1}^{t} f_n(x_i) = f_i^{(t-1)} + f_t(x_i), \quad (10)$$

$$Obj^{(t)} = \sum\nolimits_{j=1}^{n} \iota(\widehat{y}, y) + \sum\nolimits_{j=1}^{t} \alpha(f_i), \quad (11)$$

$$\alpha(f) = \delta\tau + \frac{1}{2}\sum \mu^2, \quad (12)$$

Given the predictive function $f_t(x_i)$ at the time step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ represent optimized functions at consecutive time steps t and $t-1$. To avoid overfitting while maintaining computational performance, (11) assesses the quality of the model. Chen and Guestrin [37] study express the objective functions $Obj^{(t)}$ allow for a mix of regularization and predictive terms, as well as parallel execution during training. ι is the loss function for estimating the distance between the $\widehat{y_i}$ and y_i, α is the regularization function represented in (12), δ is the minimal loss required to divide the leaf node further, σ is the regularization parameter, τ is the number of leaves in the tree, and μ is the branch score vectors. XGB, which has been widely used to solve a number of ML problems and made outstanding performance in many domains including mitigation schemes for accurate attack detection and efficient network resource utilization [38], remaining useful life of transformer insulation paper [39], fault diagnosis of diesel engine [40], and radar emitter classification [41].

5) K-NEAREST NEIGHBORS CLASSIFIER

K-nearest neighbor classifier (kNNC) is a non-parametric classification technique widely applicable in classification tasks [42]. The number of kNNC neighbors are identified, where k_n is significant. kNNC uses a vector set as the center of a circle, its circumference being determined by the variable k_n. k_n denotes the number of neighbors within the radius of the circle. A definitive value of k_n would be preferable if the input data contained outliers. In (13), following [43] notations, v_j denotes a collection of data samples, whereas (v_j, o_j) denotes a combination of the data vector and label $(o_j \in [1, C])$ and C specifies the variable set's optimum categories in the dataset. The kNNC technique is then applied to categorize a new vector $\vec{\eta}$, kNNC is estimated in (13) as

$$\underset{j}{argmin}(dist)(\vec{v}_j, \vec{\eta})\forall j = 1, \ldots, n. \quad (13)$$

$\underset{j}{argmin}$ defines the set of j values that result in the minimal likelihood value with the distance measure $dist(.)$. There are three distance matrices applied in the kNNC. Equations (14) to (16) represent the Euclidean ($dist_{Euc}$), Manhattan ($dist_{Man}$), and Minkowski ($dist_{Min}$) distances, notable for the kNNC techniques widely used for ML tasks.

$$d(y, x) = \sqrt{(n\sum_{i=1}^{n}(x_i - y_i)^2)}, \quad (14)$$

$$d(y, x) = \sum_{i=1}^{n} |x_i - y_i|, \quad (15)$$

$$d(y, x) = (\sum_{i=1}^{n}(x_i - y_i)^p)^{1/p}, \quad (16)$$

where $d(y, x)$ is the neighbor distance $dist(.)$ as $dist_{Euc}$, $dist_{Man}$, or $dist_{Min}$, for two vectors x and y given length n. p is the integer power order between two points. The Minkowski distance is transformed into Euclidean distance when p is 1 and the Manhattan distance when p is 2.

6) ADABOOST CLASSIFIER

AdaBoost classifier (ADC), which stands for adaptive boosting, is an ensemble learning technique used in ML for problems associated with regression and classification. In 1995, Yoav Freund and Robert Shapire invented the AdaBoost algorithm [44]. The ADC's primary principle is to iteratively train a weak classifier on a training dataset, with each subsequent classifier assigning more weight to the misclassified data points. Combining the weak classifiers used for training with the weights assigned to the models based on their accuracy results in the final ADC model. The model with the lowest accuracy is given a lower weight and the weakest model with the best accuracy is given the highest.

7) BAGGING CLASSIFIER

Bootstrap aggregator, commonly called bagging, is an ML ensemble meta-technique established to increase the stability and accuracy of ML algorithms. Bagging is a technique introduced by Breiman in 1996 [45]. Bagging is employed for ML classification and regression, which reduces variation and helps prevent overfitting. Usually, a decision tree is another practical use case for bagging. A specific instance of the model averaging approach is bagging.

8) RANDOM FOREST CLASSIFIER

Random forest (RF) combines tree predictors in which the values of a random vector sampled independently and with the same distribution for all trees in the forest are used to predict the values of each tree [46]. The RF classifier meta estimator predicts accurately while handling overfitting via sub-sampling dataset with averaging.

9) EXTREMELY RANDOMIZED TREES CLASSIFIER

An extremely randomized trees classifier (ERTC), or a highly randomized tree classifier, varies from a decision tree in techniques and construction but is similar to the RF. ERTC is an extreme technique with fully randomized tree inferencing from different constructions [47]. ERTC implements a meta-estimator with a randomized node split from all the data. ERTCs are characterized by low variance and faster node splits.

B. META-LEARNER

The meta-learner combines the base predictors into a stacked weighted model with assigned weights for an optimal combined super learner prediction [6], [15]. The meta-learner, also known as the aggregator, is the next level following the base learner. The meta-learner collects base model forecasts into meta characteristics. The learning process combines these assumptions to provide the final forecast. The meta-learner undergoes training using the validation dataset's forecast results as well as the model's predictions. The meta-learner's purpose is to determine the optimum approach from the weighted rules in order to reduce errors during prediction.

C. ENSEMBLE SUPER LEARNER

The eSL is a mixture of two layers. The first layer is the base learners, and the second layer is the meta-learner, creating an ensemble of learners' prediction algorithms. Heterogeneous prediction models with given weights result in the optimal aggregation for a prediction function [15]. The weights of the candidate learners are calculated using a ten-fold cross-validation to minimize the loss function. The eSL method transforms a training dataset into a prediction dataset with k-fold partitions.

According to Latha et al. [17], the super learner approach was reliable for compressive value forecasting in high-performance concrete. In another study, Lee et al. [14] executed heterogeneous combinations to predict the genotoxic description for different Multi-Walled Carbon Nano Tubes. Casas and Vanerio [20] used the super learner for data analysis strategy to detect traffic anomalies. In another study, imbalanced datasets classification task showed better performance results with a super learner [22]. In [23], an empirical study for vehicle-type traffic surveillance classification provided compelling results with the super learner.

In this study, the meta-learner is a LR and extended base learner. $Obj_i = (X_i, Y_i), i = 1, 2, 3, \ldots, n$ is the objective function to estimate the LR $\psi_0(X) = E(Y|X)$, where $X(X \in \chi)$ and Y are the input parameters and the result of interest, respectively. The outcome of the regression is described as the minimization of the predicted loss $E[\iota(Obj, \psi)]$ and expressed in (17),

$$\psi_0(X) = \underset{x}{argmin} E[\iota(Obj, \psi)], \tag{17}$$

where the loss function is ι. χ is the input from the ESKOM dataset and n is the total number of observations. Each k-fold validation and training set are indicated as V_ν $(\nu = 1, 2, 3, \ldots, k)$ and T_ν $(\nu = 1, 2, 3, \ldots, k)$. Assume $\hat{\Psi}_j (j = 1, 2, 3, \ldots, J)$ is a collection of J base learners derived from standard approaches. In the νth fold, each base model, $\hat{\Psi}_j$, is fitted using T_ν and the results in the associated set are produced in (18). Each base learner's forecasts are organized in layers to form a prediction matrix $Z = \hat{\Psi}_{j,T_\nu}(V_\nu)$. Equation (19) establishes a collection of weighted sets of possible base learners, annotated using a weight vector ϕ. In (20), The following phase determines the weight vector ϕ and avoids cross-validated errors between the overall acceptable weight vector sets as well as for the ground truth result Y. The final eSL $\hat{\Psi}_{SL}(X)$ in (21), is created by combining the ideal weight vector $\hat{\phi}$ with $\hat{\Psi}_j(X)$ using $m(z|\phi)$.

$$\hat{\Psi}_{j,T_\nu}(V_\nu), (j = 1, 2, 3, \ldots, J), \tag{18}$$

$$m(z|\phi) = \sum_{j=1}^{J} \phi_j \hat{\Psi}_{j,T_\nu}(V_\nu), \sum_{j=1}^{J} \phi_j = 1, \tag{19}$$

$$\hat{\phi} = \underset{\phi}{argmin} \sum_{i=1}^{n} (Y_i - m(z_i|\phi))^2, \tag{20}$$

$$\hat{\Psi}_{SL}(X) = \sum_{j=1}^{J} \hat{\phi}_j \hat{\Psi}_j(X). \tag{21}$$

D. FEATURE ENGINEERING WITH CONFORMITY

We performed noise estimation on the ESKOM data to enhance feature selection. A case for distribution overlap and label issues was examined. For conformity, a random selection of characteristics was made based on the available filter, wrapper, embedded, and hybrid techniques [48]. A higher number of features may lead to model overfitting and considerably more computationally demanding. To limit the overhead and computational cost and simplify the complexity of the model, we considered feature filters from available grounded techniques. Again, the expensive overhead for the stacked eSL was considered for practicality and experimentation.

1) ALL USABLE FEATURES

All ESKOM feature variables, without redundant variables, excluding manual load reduction (MLR), interruptible load sheds (ILS), and excluding (Excl ILS), time stamps, and residual forecast before national lockdown, were used in the feature selection process. Details of the ESKOM features can be obtained from the ESKOM data portal.

2) OLS BACKWARD ELIMINATION (BE)

The ordinary least square (OLS) elimination progresses computational competence for feature selection [49]. BE is a type of filter technique that considers the central characteristics of the features. BE is computationally less expensive for high dimensional data than the hybrid or wrapper techniques.

3) HYBRID VARIANCE THRESHOLD, SELECT K-BEST, AND XGB (VT_KBEST)

The hybrid combines the strength of multiple filtering and embedded selection techniques. The initial filter reduces features with a low variance threshold. It is assumed that high-variance features are ideal features compared to low-variance features. Further filtering with SelectKBest [50] reduces variables from all features, and gradient boosting reduces the dimension for optimal feature selection. The main advantage of the hybrid technique is the combination of the strength of different selection techniques [50], [51], [52].

4) LASSOCV EMBEDDED METHOD (CVE)

The least absolute shrinkage and selection operator (LASSO) is a regularization involving penalizing model parameters and avoiding over-fitting [53]. Features are eliminated subject to the sum of the absolute value of the coefficients and reduced to zero. LASSOCV incorporates cross-validation (CV) [52] folds, further improving the selection process. LASSO is a computationally expensive embedded feature selection technique for feature elimination.

5) RECURSIVE FEATURE ELIMINATION (RFE)

RFE removes features using attributes with assigned weights. The least valuable features are recursively pruned from the list for the desired list [54]. RFE is a wrapper feature selection technique and is computationally expensive, employing greedy search and a more significant number of datasets or features, but the accuracy is reliable. RFE base and RFE Opt were considered for experimentation. RFE base is the first level filtering, and RFE Opt further reduced the RFE base feature list.

6) PARTICLE SWARM OPTIMIZATION (PSO)

Kennedy and Eberhart proposed particle swarm optimization (PSO) in 1995 [8]. PSO is best utilized to determine the highest or lowest value of a function specified in a multilayer vector space. PSO has the potential to determine the highest or lowest value of a function specified in a multilayer vector space. The PSO algorithm will return the minimum-producing parameter.

E. LABEL CURATION

ESKOM MLR, ILS, and Excl ILS features are continuous variable representations. The discretization process involves aggregating these sets of variables into logical binary bins. Given the task a categorical problem. Discarding the granularity of the data results in a significant loss of unconsolidated information. The inflection point is suitable for the electrical load interruption task, which is indicated as load-shedding and no shedding. Applying [13], the set of classes Y is produced by using a training set made up of n signals with the values $x_{(1)}, \ldots, x_{(n)}$ in the input space X, which is p-dimensional. The ESKOM MLR, ILS, and Excl ILS were aggregated for electricity power interruption and labeled for $x_{(1)}$ to $x_{(n)}$ is for ESKOM load-shedding or no shedding categories. The ESKOM feature space X excludes the label variables.

F. EVALUATION INDICATORS

1) BALANCED ACCURACY

Balanced accuracy is a performance estimate for imbalanced datasets by computing the recall average obtained from a specific class [55]. In (22), the true positive (TP) is the positive instance, which the identified categorization models accurately; true negative (TN) negative instance categorization as shown by the classification model, false positive (FP) is the negative instance that is classified incorrectly in the positive class, and false negative (FN) are positive instances that are incorrectly classified in a negative class [56].

$$\text{balanced-accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right), \tag{22}$$

2) CONFUSION MATRIX

The confusion matrix is derived from the classification assessment and a combination of metrics [57], [58]. The confusion matrix illustrates the number of correct classifications on the sloping side of the matrix. In (23) and (24), metrics precision (sensitivity: P_r), also known as TP_{Rate}, is the amount of TP over the number of TPs added to the number of FP. Recall (specificity: R_r) described as true negative rate TN_{Rate}. The recall and precision standard metric for a particular performance estimate is described as the harmonic mean ($F1score$): See (25) [56]. All metrics have various benefits and drawbacks and are considered differently in balanced and imbalanced datasets. Hence, it is critical to consider the class distribution of the dataset to choose appropriate metrics for meaningful performance evaluations.

$$P_r = \frac{TP}{(TP+FP)}, \tag{23}$$

$$R_r = \frac{TP}{(TP+FN)}, \tag{24}$$

$$F_1score = \frac{(2 * P_r * R_r)}{(P_r + R_r)}. \tag{25}$$

3) AREA UNDER THE CURVE OF A PRECISION-RECALL CURVE

The area under the curve of a Precision-Recall curve (PR-AUC) is an illustration of the performance of the ML model with precision (specificity) and recall (sensitivity). PR-AUC is tractable for imbalanced classes, and the plot is more accurate compared to receiver operating characteristic (ROC) curves [58]. PR-AUC is a criss-cross plot that is less velvety and convex than the ROC curve. A typical issue with PR-AUC lies in the interpretability between points on the PR curve, resulting in numerical integration under the curve complex [56]. Again, PR-AUC algorithms that optimize the area under the ROC curve are not guaranteed to optimize the area under the PR curve [59].

4) BRIER SCORE LOSS AND BRIER SKILL SCORE

The Brier score loss estimates the probabilistic accuracy of forecasts. The use case for Brier score loss is when there is an occurrence of an event or no occurrence. In a binary task, the best Brier score loss value is 0, and the worst achievable score is 1. Hence, a lower Brier score loss shows a more accurate prediction. Brier skill score compares two Brier score losses by comparing the benchmark and innovative models. The Brier Sill Score is an important metric used to uncover the goodness of fit from a Brier score loss model across all probabilistic predicted on the holdout set. The Brier score is estimated in (26) as:

$$S_B = 1/n \sum_{(i=1)}^{N} (\hat{y}_i - y_i)^2, \tag{26}$$

where S_B is the Brier score, N is the number of observations, $\hat{y}_i$ and y_i are the predicted and experimental values, respectively. In (27),

$$SS_B = (S_B - S_N)/S_B, \tag{27}$$

where SS_B is the Brier skill score, S_N is the Brier score loss of the new model, and S_B is the Brier score loss for the benchmark model. The Brier skill score focuses on the relative metric lacking in Brier score loss. A negative score shows a weaker model than the base model, 0 implies equality and a positive value means the performance of the new model is superior to the experimental model.

5) CLASS LIKELIHOOD RATIO

The class likelihood ratio is a statistical test to evaluate the optimal fit from statistical models. The class likelihood ratio is a famous test used in energy classification studies [60]. The Class likelihood ratio is a valuable metric for computing the positive and negative likelihood ratios. The metric is class invariant and ideal for class imbalance. There are two possible likelihood ratios for the predictive power of binary classification tasks (the positive LR+ and negative LR- likelihood ratios). A positive odds ratio was considered in the holdout test experiment. In (28), the positive likelihood ratio of (LR+) is the ratio of P_r sensitivity by the difference of R_r specificity from one.

$$LR+ = \frac{P_r}{(1 - R_r)}, \tag{28}$$

6) CRITICAL DIFFERENCE DIAGRAMS

Another intriguing tool for displaying retrospective test statistics is the critical difference diagram. The results within every component are first rated in a block design scenario, and the average rank for the entire result for each treatment is plotted along the x-axis. Groups of treatments that do not show statistically significant differences are then given a crossbar. Solid bars represent groupings in which there is little to no variance between classifiers. Difference tests were performed using paired Wilcoxon signed rank tests with Holm correction [61].

III. STACKED HETEROGENEOUS ENSEMBLE SUPER LEARNER

In completing the stacked eSL for the classification task defined for ESKOM electricity load interruption, we followed the recommended guidelines in [6]. However, the proposed stacked eSL model architecture is tailored to the characteristics of the data and the predictive tasks. The preliminary analysis of the ESKOM data follows a feature filtering technique by considering collaborating features to train a stacked eSL model. These attributes illustrate each projected pair with all other predicted pairings, using a grading of feature techniques (including filter, wrapper, embedded, and feature nature-inspired optimization feature filtering techniques). The choice for feature filtering was arbitrarily but stratified within the fundamentals required for ML model performance. The stacked eSL model is comprised of base and meta learners.

A. EXPERIMENTAL WORKFLOW

We model the ESKOM data with nine base learners in line with general practices and researcher heuristics. The aim is to classify load interruption (load-shedding) and no interruption (no shedding). The positive class is the load-shedding, and the negative class is no-shedding. *TP* is the number of predicted incidences of load-shedding that are load interruptions, *TN* is the number of predicted no-shedding that are non-interrupt incidences, *FP* is the number of no-shedding incorrectly classified as load interruptions, and *FN* is the number of load-shedding incorrectly classified as no interruption.

The implementation of eSL required a library of base learners and a metal learner. The proposed stacked eSL, which includes manual MLR, ILS, and Excl ILS from the ESKOM dataset, was suitable for classifying electric load-shedding. The implementation pipeline began with cross-validation [52] to distribute the ESKOM data into k-fold subsets using stratified 10-fold cross-validation. Python packages, including sci-kit-learn [62], xgboost [37], swarm optimization [63], NumPy [64], and pandas [65], were required to preprocess and develop models (see stage 2 in Fig. 1).

The workflow consists of three subsystems. The first stage illustrates the historical processes of ESKOM that involve electricity generation, interruption, data acquisition, and distribution. We include supply interrupt as specified for the present study pipeline subsystem. The second stage acquired ESKOM data and established feature engineering through the implementation process for cleaning, normalization, discretization of labels, and techniques for feature filtering. In the third pipeline, the scaled features passed from the second subsystem implement cross-validation, with ten folds. The base models' predictions were passed into the final meta-learner for the stacked eSL.

B. HYPERPARAMETER OPTIMIZATION AND PERFORMANCE MEASURES

Each base learner utilizes several hyperparameters that need to be configured before the learning process can begin. They are adjustable and can directly effect how well the model trains, therefore carefully consideration was required in the selection process to achieve the most significant results. The hyperparameters of the eSL model are given in Table 1.

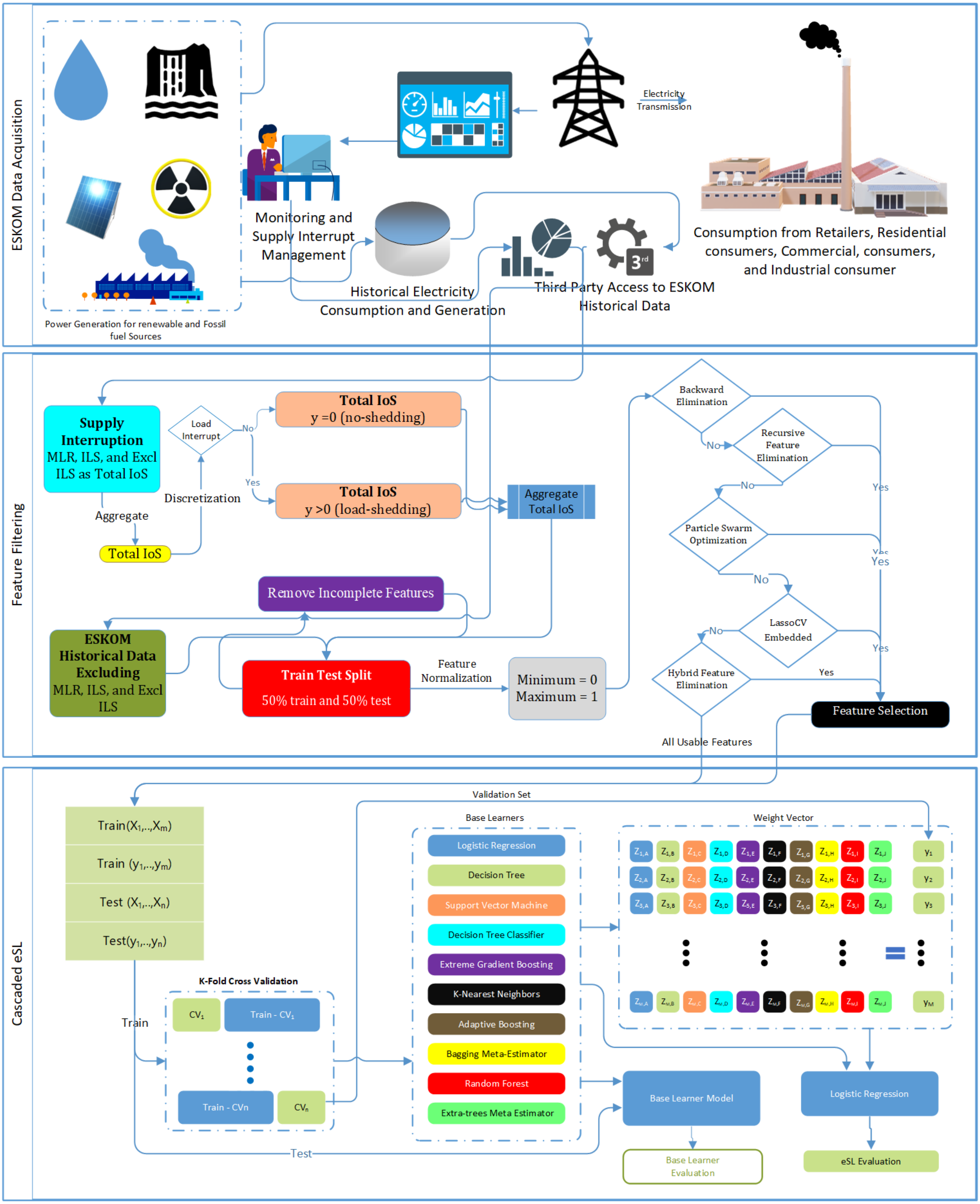

FIGURE 1. The eSL workflow three subsystems.

To control the source of randomness, a uniform random state was set across all models in the base learners and meta-learner models.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. ESKOM DATA DESCRIPTION

We collected hourly electricity data from the South African ESKOM domestic utility company. ESKOM hourly electricity dataset is available on a shareable website: https://www.eskom.co.za/dataportal. Hourly dataset logs from ESKOM operations for electrical power generation, demand, and supply interruptions comprise the features and labels for experimentation. The aggregate sum from supply interruptions makes up the class label, and the features are the residual electricity demand and generation from fossil fuel and renewable sources. The hourly data for the experiment span from April 1, 2019, 12:00:00 AM to May 19, 2023, 11:00:00 PM. The total logged period was 36,040 hours, equivalent to 4 years, 1 month, and 19 days. An equal ratio of train-test split determined in-sample and out-of-sample sets.

The ESKOM data description for the features excluded label ILS, MLR, and Excl ILS discretized as Total interruption of supply (IOS). All remaining variables were considered as features for filtering. ILS Usage is the predetermined load interrupt without notice from ESKOM National Control Center, MLR are isolated restrictions on electric load usage, and Excl ILS are other categories of load restriction separate from MLR and ILS. Another feature removed was ''Original Res Forecast before COVID-19 Lockdown''. The latter is the change from the national lockdown.

The ESKOM data portal provides electricity statistics. In Table 2, Std is the standard deviation, while min and max are the minimum and highest values, respectively. The lower, median, and higher quartiles are respectively 25%, 50%, and 75%.

B. EXPLORATORY DATA ANALYSIS

An exploratory data analysis (EDA) was required for elucidation. The probe reveals features distribution and relationships with the ESKOM total supply interruption. Further analyses provided more precise intuition for the model's predictions. In Fig. 2, the ESKOM load-shedding binary category shows no interruption of supply (no shedding) and interruption of supply (load-shedding). The count of ESKOM data indicates an imbalance in the distribution of class labels (82. 5% no load shedding and 17. 5% load shedding).

C. FEATURE CONFIGURATION

The ESKOM dataset had 42 features and 36240 observations. After removing load interruption features (MLR, ILS, and Excl ILS summed as Total IOS and discretized as a categorical variable), incomplete observations features (Date Time Hour Beginning, and Original Res Forecast before Lockdown. The ESKOM unlabeled feature had 47 not-a-number (NAN) replaced with zeros for consistency. The

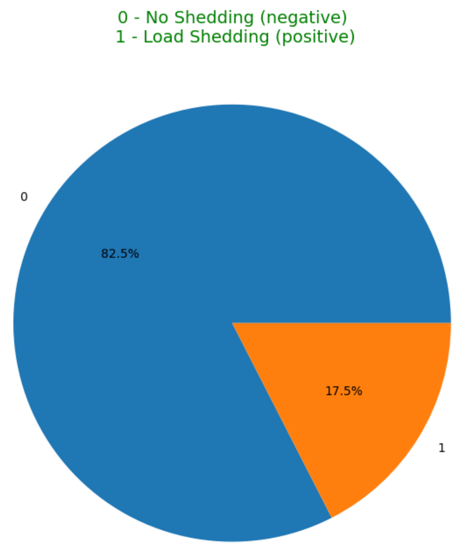

FIGURE 2. ESKOM load-shedding Binary Category. The frequency of no-shedding observations is approximately four times the frequency of load-shedding categorization.

train-test configuration for the train (50%) and test (50%) remains the same in all models, with variations in the number of feature selections. In Fig. 3, feature selections for various configurations are given as All (38), RE_LCV (9), RFE_base (19), RFE_Opt (5), VT_KBEST (15), PSO (22), and BE_OLS (15).

All (38) feature variables were not scaled down using a feature filtering algorithm; all usable variables were incorporated for the classification model task. The RE_LCV (9) feature filtering model eliminated 29 variables. The RE_LCV (9) tuning parameter resulted in an alpha score of 0.034. The best score was 0.236, and 5-fold cross-validation. The RFE_Opt (5) selected five features with a score of 0.889128 for the optimum model from the RFE base selection of 19 features. In BE_OLS (15) with 15 features selected from the list. The tunable p-value parameter discards p-values greater than 0.05. The PSO (22) combines the SVC and particle swarm optimization (PSO) [66] to achieve a subset accuracy of 0.777 compared to all features' accuracy of 0.735. VT_KBEST (15) combines variance threshold [51], SelectKBest [37] and [50] for feature filtering in a pipeline implementation for optimal feature selection.

For comparison of the ESKOM electricity load interruption dataset, all seven models were implemented using feature filtering techniques in combination with stacked eSL using Google Collaboratory [67]. Google Collaboratory is a virtual configuration running a Linux operating system with Python 3 programming language and a suite of supported packages. Hardware acceleration was not required.

V. RESULTS AND ANALYSIS

After optimization of the feature filtering techniques, the prediction from the stacked eSL meta-model and the competencies of the base models and meta-model were evaluated in the holdout sets. The classification accuracy was analyzed

TABLE 1. Hyperparameters of the eSL base learners and meta-learner models.

Model	Hyperparameter	Objective function
Logistic regression	Inverse regularization strength	100
	Solver for model optimal fit	liblinear
	Maximum Iteration for convergence	100
	Tolerance for stopping criteria	0.0001
Decision tree classifier	Criterion	Gini
	Minimum number of samples required to split an internal node	2
	Minimum number of samples required to be in a leaf node	1
Support vector classifier	Inverse regularization strength	100
	Decision Function	one-vs-rest
	kernel	radial basis function (RBF)
	Degree of the polynomial kernel function	3
	Tolerance for stopping criteria	0.0001
Extreme gradient boosting classifier	Learning task	binary:logistic
K-nearest neighbors classifier	Number of neighbors	5
	Power parameter	Minkowski
	Maximum number of leaf node	30
AdaBoost classifier	Weight applied to each classifier	1
	The maximum number of estimators at which boosting is terminated	50
Bagging classifier	The number of features to draw from	1
	The number of samples to draw from	1
	The number of features to draw from	10
Random forest classifier	Criterion	Gini
	The number of features to consider when looking for the best split	Square root
	The number of trees in the forest	10
	The minimum number of samples required to be at a leaf node	1
	The minimum number of samples required to split an internal node	2
Extremely randomized trees classifiers	Criterion	Gini
	The number of features to consider when looking for the best split	Square root
	The number of trees in the forest	10
	The minimum number of samples required to be at a leaf node	1
	The minimum number of samples required to split an internal node	2

Optimized hyperparameter settings for the base learners and meta-learner models.

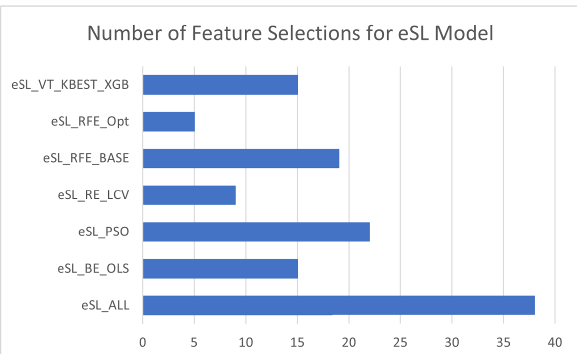

FIGURE 3. Number of selected features from South Africa ESKOM data portal variables.

based on balanced accuracy, confusion matrix, PR_Curve, Brier skill score, class likelihood ratio, and critical difference factor.

A. RESULTS

1) BALANCED ACCURACY VS. ACCURACY

In Table 3, Fig. 3, and Fig. 4, balanced accuracy for the base learner and the meta-learner in six models shows improved performances. The strengths of individual base learners complemented feature selections for meta-learners. In Table 3, the base learner for eSL_RFE_Opt SVC (90.694%) was the highest score with feature filtering. Likewise, in eSL_VT_KBEST_XGB, where ADC scored highest, eSL_RFE_Opt had the highest score compared to other base learner models. Across base learners, the ADC (71.301%) for VT_KBEST_XGB, eSL_RFE_Opt Bagging classifier (84.368%), DTC (83.084%), ERTC (79.758%), kNNC (79.166%), LR (88.828%), RF classifier (81.578%), SVC (90.694%), and XGB Classifier (86.264%) models were most significant results. Of all nine base learners, the SVC and XGB classifier models had the two highest balanced accuracy scores.

In Table 3 and Fig. 5, the accuracy and balanced accuracy scores for the meta-learners show the scores for eSL_RFE_Opt and eSL_VT_KBEST above the 91% mark compared to the other meta-learners. The eSL_RFE_Opt (91.319% and 91.421%) meta-learner result was highest for accuracy and balanced accuracy, closely followed by the eSL_VT_KBEST_XGB (89.936% and 89.817%) and RFE_BASE (89.953% and 89.834%). The eSL_RE_LCV (54.726% and 54.210%) meta-learner model had the lowest score for accuracy and balanced accuracy.

2) CONFUSION MATRIX

The classification results of stacked eSL models were assessed with confusion metrics to illustrate error types in 4 defined categories. In Fig. 6(a) to Fig. 6(g), the confusion metrics have two rows and two columns for the no shedding and shedding electricity load using six filtering stacked eSL models. The correct classification

TABLE 2. ESKOM data from 2019 to 2022.

Feature	Mean	Std	Min	25%	50%	75%	Max
Residual Forecast	24229.95	2925.757	14319.14	21876.69	24533.93	26274.56	34134.04
RSA Contracted Forecast	25849.55	3240.577	15172.65	22879.23	26581.77	28326.22	35034.35
Dispatchable Generation	23633.41	3019.001	13797.94	21312.51	23661.73	25834.63	33065.91
Residual Demand	24234	2963.839	13797.94	21893.78	24475.66	26291.33	34029.03
RSA Contracted Demand	25865.13	3250.721	14929.99	22942.84	26585.15	28324.57	35004.75
International Exports	1531.001	235.5965	0	1363.387	1514.232	1693.292	2375.939
International Imports	1169.106	233.6048	0	1088	1180	1329	1765
Thermal Generation	20838.29	2201.569	13774	19349	20949	22423.61	27807
Nuclear Generation	1279.211	442.6647	-36	911	926	1834	1854
ESKOM Gas Generation	0.287362	7.297938	0	0	0	0	323
ESKOM OCGT Generation	226.77	453.4027	0	0	0	231	2120
Hydro Water Generation	202.9025	242.1064	0	0	80	423	610
Pumped Water Generation	546.2026	634.6249	0	0	273	1006	2746
Dispatchable IPP OCGT	101.3225	242.6568	0	0	0	0	1021.874
ESKOM Gas SCO	-1.77773	0.428777	-4	-2	-2	-2	0
ESKOM OCGT SCO	-2.92638	1.827023	-16.53	-4.93	-3	-1.6	0
Hydro Water SCO	-6.04E-06	0.000118	-0.004	0	0	0	0
Pumped Water SCO Pumping	-725.987	963.0068	-2848	-1696	-29	-14	0
Wind	925.9493	498.7223	19.803	543.488	859.5205	1229.163	3028.065
PV	509.3761	639.6039	0	0	21.783	1086.383	2099.486
CSP	179.4642	172.3296	0	0	139.607	337.6583	506.249
Other RE	16.34347	10.08129	0.849	9.308	13.076	18.462	46.997
Total RE	1631.133	903.6248	48.747	892.4368	1491.342	2275.854	5126.079
Wind Installed Capacity	2649.308	539.5236	2079.76	2079.76	2495.02	3163.37	3442.57
PV Installed Capacity	1979.511	313.3653	1474.19	1774.19	2211.09	2212.09	2287.09
CSP Installed Capacity	500	0	500	500	500	500	500
Other RE Installed Capacity	31.01391	12.52072	21.78	21.78	25.58	50.58	50.58
Total RE Installed Capacity	5159.834	831.6381	4075.73	4375.73	5231.69	5926.04	6280.24
Installed ESKOM Capacity	45889.32	1137.501	43691	44926	46329	46800	47520
Total PCLF	4873.936	1672.498	695.777	3639.242	4836	6018	11289.42
Total UCLF	11618.74	2862.557	4670.626	9181.047	11524.04	13921.87	19421.49
Total OCLF	1002.563	593.0825	78.025	547.258	844.875	1371.314	5219.432
Total UCLF+OCLF	12620.81	2757.101	5658	10394	12608	14737.25	21535
Non Comm Sentout	452.7755	306.2853	0	163	443	719	1922
Drakensberg Gen Unit Hours	493.4905	537.5243	0	0	445.5	768.25	2506
Palmiet Gen Unit Hours	82.31225	10.5589	21.4	75.8	84.6	90.3	102
Ingula Gen Unit Hours	42.20647	9.22606	9.1	35.7	43.7	49.6	58.7
New(Undefined)	36.73689	9.550602	0	30	37.05	43.71	63.6

Details of terminology abbreviations available on ESKOM data portal glossary page.

TABLE 3. Base learner and meta-learner results from holdout sets.

Learner	Models	ALL	BE_OLS	PSO	RE_LCV	RFE_BASE	RFE_Opt	VT_KBEST_XGB
Base-Learner	**ADC**	64.9	68.128	62.213	52.997	66.488	67.716	**71.301**
	Bagging classifier	55.05	65.221	66.278	54.277	72.542	**84.368**	79.829
	DTC	56.48	69.853	69.058	55.7	61.773	**83.084**	76.988
	ERTC	55.05	57.765	53.881	53.368	63.304	**79.758**	63.94
	kNNC	58.64	55.214	60.352	52.807	60.638	**79.166**	61.512
	LR	77.53	*88.856*	*87.236*	53.561	*88.621*	**88.828**	88.705
	RF classifier	57.88	59.153	59.937	53.206	59.758	**81.578**	67.93
	SVC	76.62	86.717	81.796	52.765	88.495	***90.694***	*89.169*
	XGB classifier	*78.368*	81.312	72.752	*55.847*	82.153	**86.264**	84.837
Meta-Learner	**Brier score loss**	21.38	10.329	12.714	45.274	10.048	**8.579**	10.064
	class_likelihood_ratios LR+	nan	nan	6806.48	5.945	7300.918	nan	nan
	PR-AUC Score	89.06	94.713	93.488	69.923	94.853	**95.609**	94.849
	ROC-AUC Score	78.37	89.549	87.136	54.21	89.834	**91.319**	89.817
	Accuracy	78.62	89.671	87.286	54.726	89.953	**91.421**	89.936
	Balanced_Accuracy	78.37	89.549	87.136	54.21	89.834	**91.319**	89.817

counts were identified in the *TN* and *TP* columns. Similarly, type I and type II error counts were identified in the *FN* and *FP* columns for mistaken classes. The eSL_RFE_Opt (Fig. 6(f)) model had the highest correct classes with counts for *TN* (50.59%) and *TP* (40.84%) categories. This was followed by the eSL_VT_KBEST_XGB (Fig. 6(g)) model with counts for TN (50.59%) and *TP* (39.35%). The class of interest was within the eSL_RFE_Opt model 0% was misclassified as load-shedding, and 8.58% was misclassified as no shedding in the results. Similarly, in the eSL_RFE_BASE model, 0% was misclassified as load-shedding, and 10.06% was misclassified as no shedding. The least performing result is the eSL_RE_LCV (Fig. 6(d)) model *TN* (49.72%) and *TP* (5.00%) with lowest misclassified FN (0.86%) and highest misclassified FP (44.41%).

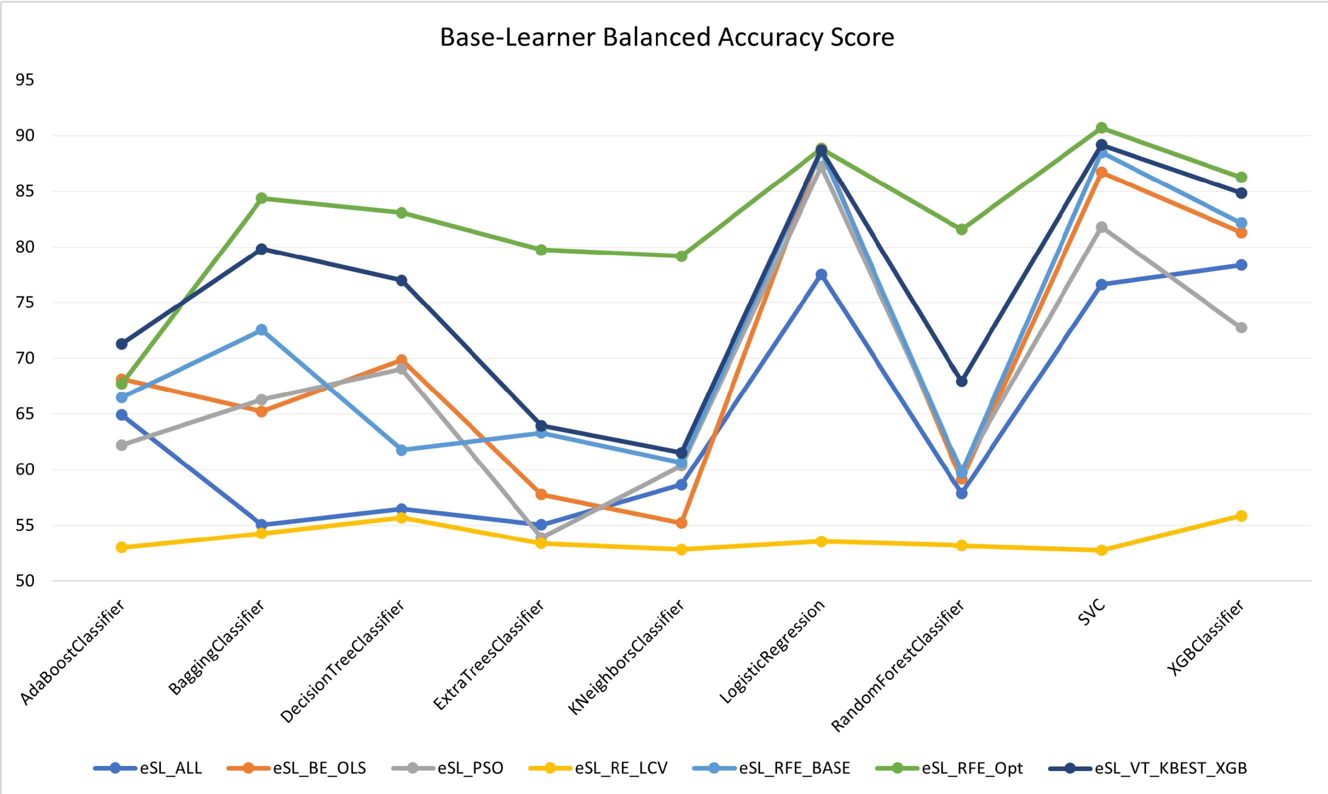

FIGURE 4. **Model Comparison for the Base Learner given the number of features selections.**

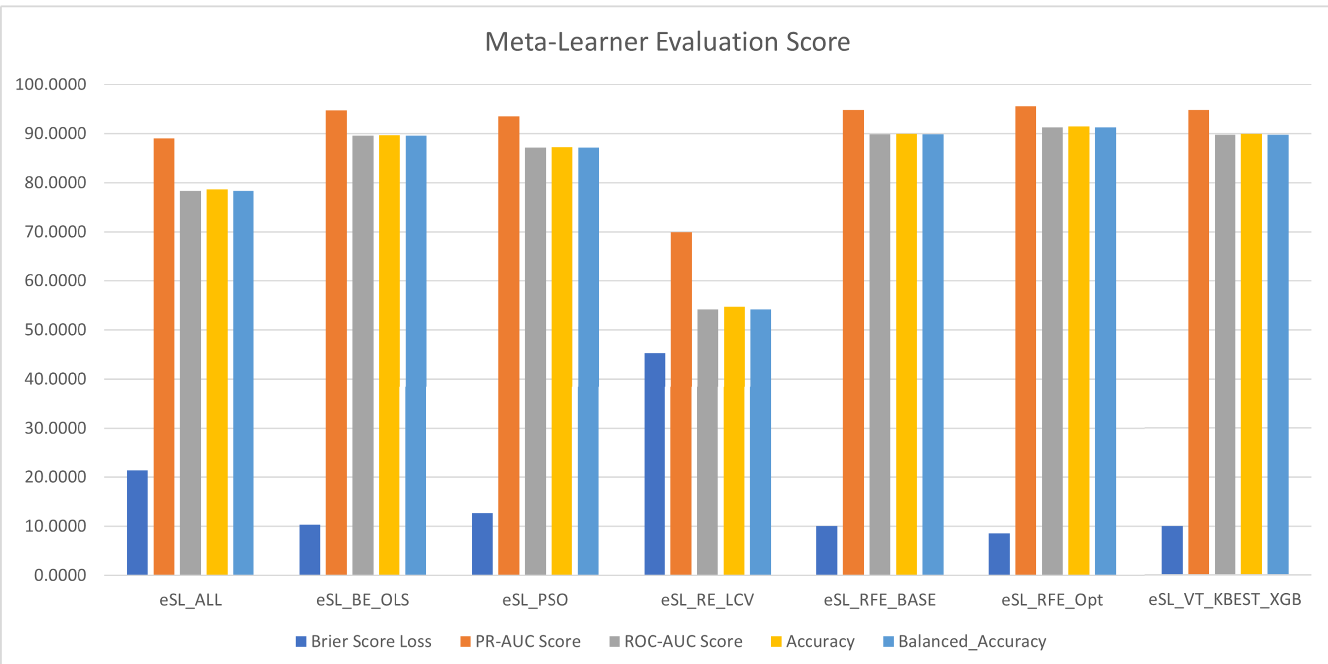

FIGURE 5. **Meta-Learner comparison bar plots.**

3) PRECISION RECALL CURVE

Fig. 7 (a) to Fig. 7(g) shows the precision-recall curve for the stacked eSL model capabilities. Plots show that the area between the precision and recall curves illustrates the model's predictive power on the holdout set. As shown in graphs scaled from 0 to 1, recall scores are comparatively high in all models, with higher variance in precision margin for all models except eSL_RE_LCV. The RFE (Fig. 7(f)) model produced the most significant area under the curve, followed by the hybrid eSL_VT_KBEST_XGB (Fig. 7(g)), predicting

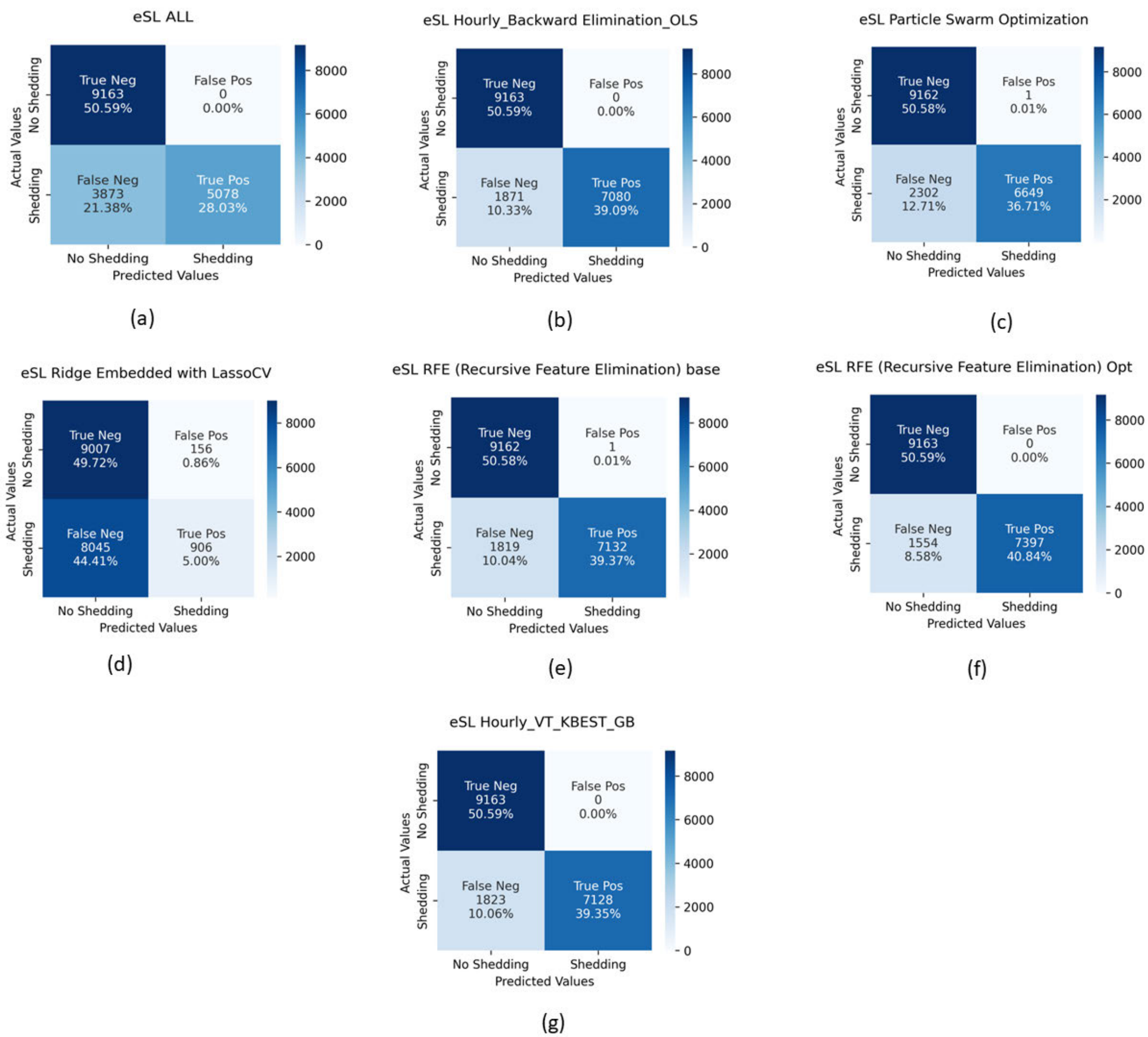

FIGURE 6. (a) to (g) shows the confusion metrics plot.

TABLE 4. Cross-examination of the stacked eSL brier skill score.

Classifier	ALL	BE_OLS	PSO	RE_LCV	RFE_BASE	RFE_OPT	VT_KBEST_XGB
ALL	0	0.517	0.405	-1.117	0.53	0.599	0.529
BE_OLS	-1.07	0	-3.383	0.027	0.027	0.169	0.026
PSO	-0.682	0.188	0	-2.561	0.21	0.325	0.208
RE_LCV	0.528	0.772	0.719	0	0.778	0.811	0.778
RFE_BASE	-1.128	-0.028	-0.265	-3.506	0	0.146	-0.002
RFE_OPT	-1.492	-0.204	-0.482	-4.277	-0.171	0	-0.173
VT_KBEST_XGB	-1.125	-0.026	-0.263	-3.499	0.002	0.148	0

Stacked eSL Brier skill score, lower is better.

an improved agreement between scaled features and stacked eSL techniques. The eSL_RE_LCV technique is nearly a flat recall and high precision.

4) CLASS LIKELIHOOD RATIO

In the case of meta-learner, for class_likelihood_ratios (LR+) the eSL_ALL, eSL_BE_OLS, eSL_RFE_Opt, and eSL_VT_KBEST_XGB had *FP* ratio for specificity score as zero and indicting higher LR+, but for eSL_PSO, eSL_RE_LCV, and eSL_RFE_BASE, LR+ scores were 6806.478, 5.945, and 7300.918. The eSL_RE_LCV class likelihood is the lowest. The result indicates that the odds of a holdout set of true positive increases with respect to the pre-test odds.

5) BRIER SKILL SCORES

A further examination of the meta-learner with the Brier score loss in Table 3 and Fig. 5 for eSL_RFE_Opt (8.579%),

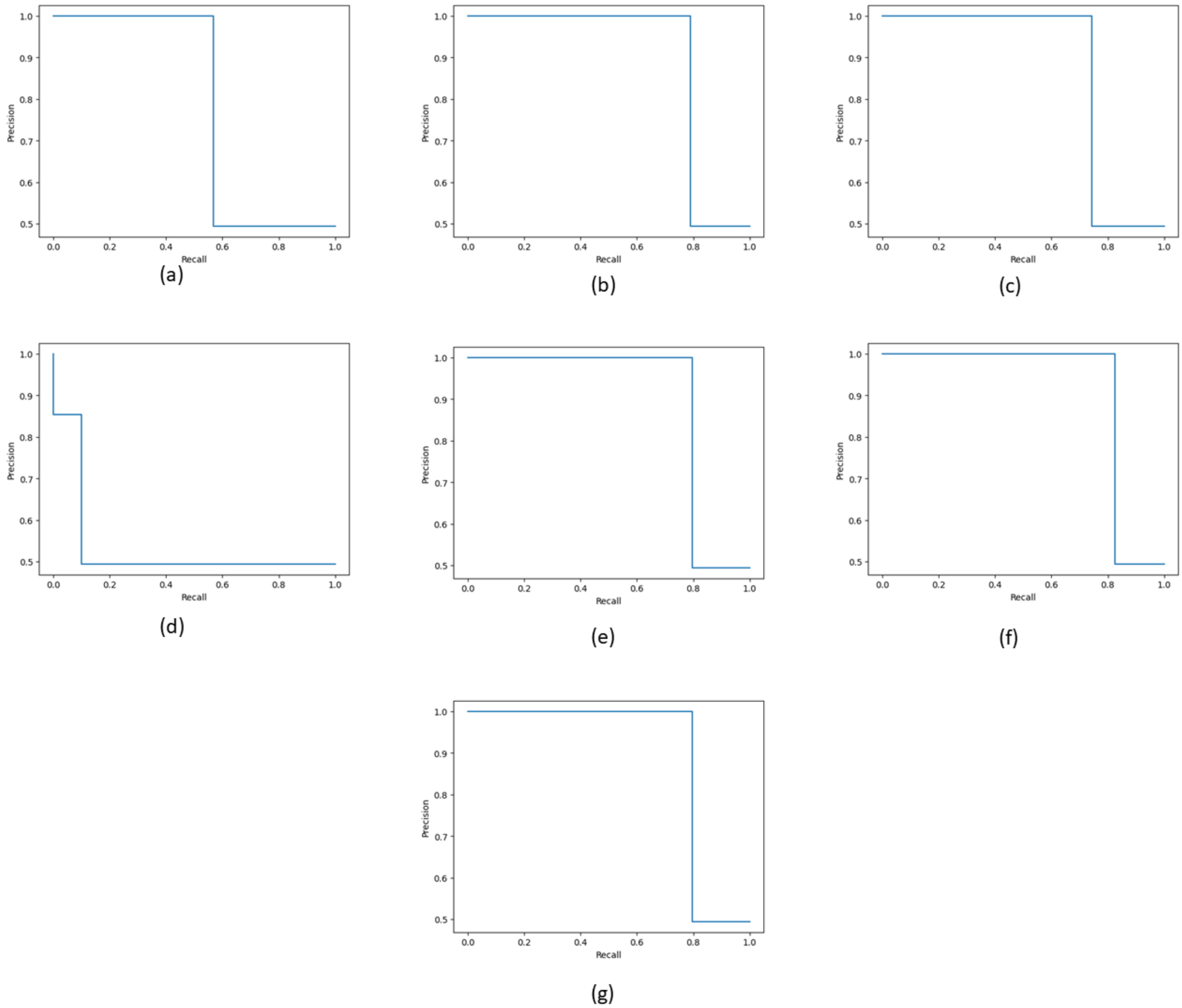

FIGURE 7. (a) to (g) shows the PR_curve plot.

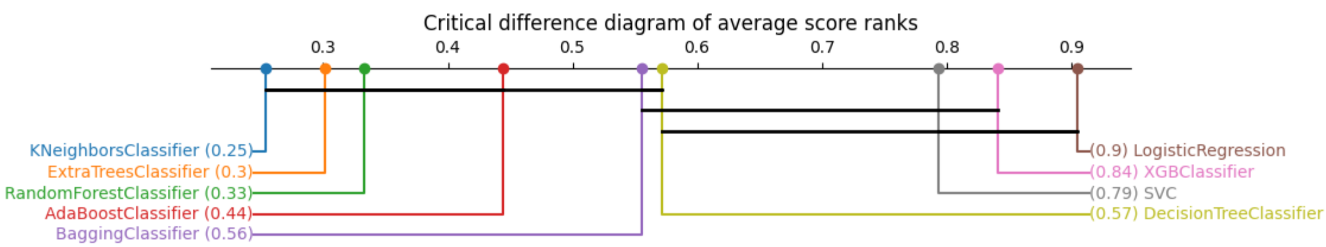

FIGURE 8. Illustrates the base learners' average result rank.

followed by the eSL_VT_KBEST_XGB (10.064%), and eSL_RFE_BASE (10.048%) were the lowest scores. Again, the eSL_RE_LCV had the worst metric loss, 45.274%, from all compared results. In Table 4, cross-examination of the Brier skill scores further confirms the gains from the eS_RFE_Opt proposed configuration.

6) CRITICAL DIFFERENCE FACTOR

The base plot in Fig. 8 shows that the XGB classifier model came second only to the LR model, which is superior to all other models. The least-ranked models were ETC and kNNC. These are followed in that order by the RF classifier, ADC, and Bagging classifier models.

However, there is not enough statistical justification for other comparisons.

B. DISCUSSION

We evaluated feature filtering for stacked eSL models in the ESKOM dataset for electricity load interruptions. The investigation determines the association of variables on super learner ensemble strategy while estimating performance metrics from implementation at base and meta-learners. The stacked eSL profits from the base learners and feature filtering. In Fig. 5, base learners constitute the input to the meta-learner but are not a guarantee for the meta-learner's optimum accuracy score. Similarly, feature filtering was crucial for optimal feature selection [68]. The view is well observed in the experimental results.

The balanced accuracy, PR_Curve, and Brier Score were essential metrics in the present experiment due to the imbalanced labels for ESKOM electricity load-shedding and no shedding. The proposed eSL_RFE and VT_KBEST_GB classification results show better performance. The improvement gained by the proposed eSL_RFE and eSL_VT_KBEST_GB models is demonstrated. Significant improvements in the PR-AUC score, balance accuracy, Brier skill score, and confusion metrics are observed after adjusting feature filtering with eSL_VT_KBEST_GB.

The study's findings will assist ESKOM in choosing features most critical to influencing load interruption and help it plan effective management and technical decisions on load interruption strategies in the near future.

VI. CONCLUSION

This study provides stacked eSL with feature filtering for the interruption of the ESKOM electricity load as TSC tasks. Initially, a suite of feature filtering techniques was constructed to reduce the number of features for optimal model performance. Feature filtering includes well-known filters, such as a wrapper, embedded, and hybrid techniques. ESKOM electricity load interrupt observations were discretized into a binary class for no shedding and load-shedding. We experimented with different feature filtering techniques and stacked nine base learners from the different feature filtering methods as input. The base learner's output was input for the meta-learner.

We termed the pipeline process stacked eSL. It was observed that higher accuracy in the base learners may lead to better performance in the meta-learner but is subject to optimal feature filtering. The established RFE and hybridizing models result in the selection of the features most relevant to predictive performance. Ensuring the efficacy of these tools is a function of a careful selection of features. Existing tools offer opportunities for improvement in results, but these measures show a clear justification of the model's configuration beyond accuracy. In addition, we experiment with 10-fold cross-validation and evaluate with proven techniques such as PR_Curve, Brier skill scores, and class likelihood ratio to further determine model performance.

The study highlights an intricate analysis of hourly generated data from ESKOM electricity load interruption from defined variables and accesses critical elements in electricity interruption from data. Implementing the proposed TSC model for load-shedding management can lead to economic benefits.

VII. CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ACKNOWLEDGMENT

The authors would like to thank South Africa, ESKOM, for providing the hourly energy report dataset for conducting the research.

REFERENCES

[1] Y. Wang, R. Wang, K. Tanaka, P. Ciais, J. Penuelas, Y. Balkanski, J. Sardans, D. Hauglustaine, W. Liu, X. Xing, J. Li, S. Xu, Y. Xiong, R. Yang, J. Cao, J. Chen, L. Wang, X. Tang, and R. Zhang, "Accelerating the energy transition towards photovoltaic and wind in China," *Nature*, vol. 619, no. 7971, pp. 761–767, Jul. 2023.

[2] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi, and F. S. Oueslati, "An effective ensemble learning approach-based grid stability assessment and classification," in *Proc. IEEE Kansas Power Energy Conf. (KPEC)*, Apr. 2021, pp. 1–6.

[3] H. Wang, Q. Wang, and Q. Chen, "Transient stability assessment model with improved cost-sensitive method based on the fault severity," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 20, pp. 4605–4611, Oct. 2020.

[4] K. O. Akpeji, A. O. Olasoji, C. Gaunt, D. T. O. Oyedokun, K. O. Awodele, and K. A. Folly, "Economic impact of electricity supply interruptions in south Africa," *SAIEE Afr. Res. J.*, vol. 111, no. 2, pp. 73–87, Jun. 2020.

[5] D. Linaro, F. Bizzarri, D. del Giudice, C. Pisani, G. M. Giannuzzi, S. Grillo, and A. M. Brambilla, "Continuous estimation of power system inertia using convolutional neural networks," *Nature Commun.*, vol. 14, no. 1, p. 4440, Jul. 2023.

[6] R. V. Phillips, M. J. van der Laan, H. Lee, and S. Gruber, "Practical considerations for specifying a super learner," 2022, *arXiv:2204.06139*.

[7] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[8] I. M. El-Hasnony, S. I. Barakat, M. Elhoseny, and R. R. Mostafa, "Improved feature selection model for big data analytics," *IEEE Access*, vol. 8, pp. 66989–67004, 2020.

[9] L. Jiang, N. Haiminen, A. Carrieri, S. Huang, Y. Vázquez-Baeza, L. Parida, H. Kim, A. D. Swafford, R. Knight, and L. Natarajan, "Utilizing stability criteria in choosing feature selection methods yields reproducible results in microbiome data," *Biometrics*, vol. 78, no. 3, pp. 1155–1167, Sep. 2022.

[10] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, 2022.

[11] L. K. Kumari and B. N. Jagadesh, "An adaptive teaching learning based optimization technique for feature selection to classify mammogram medical images in breast cancer detection," *Int. J. Syst. Assurance Eng. Manage.*, vol. 15, no. 1, pp. 35–48, Jan. 2024.

[12] F. Shahsavari and Z. Shaghaghian, "Application of classification and feature selection in building energy simulations," 2021, *arXiv:2108.12363*.

[13] G. A. Susto, A. Cenedese, and M. Terzi, "Time-series classification methods: Review and applications to power systems data," in *Big Data Application in Power Systems*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 179–220.

[14] S. Lee, N. Nguyen, A. Karamanli, J. Lee, and T. P. Vo, "Super learner machine-learning algorithms for compressive strength prediction of high performance concrete," *Structural Concrete*, vol. 24, no. 2, pp. 2208–2228, Apr. 2023.

[15] M. J. Van der Laan, E. C. Polley, and A. E. Hubbard, "Super learner," *Stat. Appl. Genet. Mol. Biol.*, vol. 6, no. 1, Sep. 2007.

[16] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[17] B. Latha, S. C. Pravin, J. Saranya, and E. Manikandan, "Ensemble super learner based genotoxicity prediction of multi-walled carbon nanotubes," *Comput. Toxicol.*, vol. 24, Nov. 2022, Art. no. 100244.

[18] K. Mnich, A. Polewko-Klim, A. Kitlas Golinska, W. Lesinski, and W. R. Rudnicki, "Super learning with repeated cross validation," in *Proc. Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2020, pp. 629–635.

[19] S. Butte, A. R. Prashanth, and S. Patil, "Machine learning based predictive maintenance strategy: A super learning approach with deep neural networks," in *Proc. IEEE Workshop Microelectron. Electron Devices (WMED)*, Apr. 2018, pp. 1–5.

[20] P. Casas and J. Vanerio, "Super learning for anomaly detection in cellular networks," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.

[21] M. Zhang and M. Wu, "Efficient super greedy boosting for classification," in *Proc. 10th Inst. Electr. Electron. Engineers Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Oct. 2020, pp. 192–197.

[22] S. Zian, S. A. Kareem, and K. D. Varathan, "An empirical evaluation of stacked ensembles with different meta-learners in imbalanced classification," *IEEE Access*, vol. 9, pp. 87434–87452, 2021.

[23] M. A. Hedeya, A. H. Eid, and R. F. Abdel-Kader, "A super-learner ensemble of deep networks for vehicle-type classification," *IEEE Access*, vol. 8, pp. 98266–98280, 2020.

[24] G. Valdes, Y. Interian, E. Gennatas, and M. Van der Laan, "The conditional super learner," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10236–10243, Dec. 2022.

[25] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, Aug. 2007.

[26] M. Kumar and S. K. Rath, "Feature selection and classification of microarray data using machine learning techniques," in *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 213–242.

[27] K. Maresch, G. Marchesan, and G. Cardoso, "A logistic regression approach for improved safety of the under-frequency load shedding scheme owing to feeder machine inertia," *Electr. Power Syst. Res.*, vol. 218, May 2023, Art. no. 109189.

[28] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *J. Roy. Stat. Soc., Ser. C, Appl. Statist.*, vol. 29, no. 2, pp. 119–127, 1980.

[29] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[30] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[31] C. Reinders, H. Ackermann, M. Y. Yang, and B. Rosenhahn, "Learning convolutional neural networks for object detection with very little training data," in *Multimodal Scene Understanding*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 65–100.

[32] H. R. Arabnia and Q. N. Tran, *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology: Systems and Applications*. Amsterdam, The Netherlands: Elsevier, 2016.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[34] Y. Li, N. Lu, X. Wang, and B. Jiang, "Islanding fault detection based on data-driven approach with active developed reactive power variation," *Neurocomputing*, vol. 337, pp. 97–109, Apr. 2019.

[35] S. Liu, J. McGree, Z. Ge, and Y. Xie, "Computer vision in big data applications," in *Computational and Statistical Methods for Analysing Big Data With Applications*, S. Liu, J. McGree, Z. Ge, and Y. Xie, Eds. San Diego, CA, USA: Academic, 2016, pp. 57–85. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128037324000040

[36] N. Mohanty, A. L.-S. John, R. Manmatha, and T. Rath, "Shape-based image classification and retrieval," in *Handbook of Statistics*, vol. 31, C. Rao and V. Govindaraju, Eds. Amsterdam, The Netherlands: Elsevier, 2013, ch. 10, pp. 249–267. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444538598000102

[37] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[38] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020.

[39] J. I. Aizpurua, S. D. J. McArthur, B. G. Stewart, B. Lambert, J. G. Cross, and V. M. Catterson, "Adaptive power transformer lifetime predictions through machine learning and uncertainty modeling in nuclear power plants," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4726–4737, Jun. 2019.

[40] J. Tao, C. Qin, W. Li, and C. Liu, "Intelligent fault diagnosis of diesel engines via extreme gradient boosting and high-accuracy time–frequency information of vibration signals," *Sensors*, vol. 19, no. 15, p. 3280, Jul. 2019.

[41] S. Zhao, W. Wang, D. Zeng, X. Chen, Z. Zhang, F. Xu, X. Mao, and X. Liu, "A novel aggregated multipath extreme gradient boosting approach for radar emitter classification," *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 703–712, Jan. 2022.

[42] J. Goldberger, G. E. Hinton, S. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004, pp. 513–520.

[43] S. M. Qaisar, "Adaptive rate EEG processing and machine learning-based efficient recognition of epilepsy," in *Advanced Methods in Biomedical Signal Processing and Analysis*, K. Pal, S. Ari, A. Bit, and S. Bhattacharyya, Eds. New York, NY, USA: Academic, 2023, pp. 341–373. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323859554000132

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY, USA: Springer, 2001.

[45] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[46] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee turnover prediction based on random forests and survival analysis," in *Proc. 21st Int. Conf. Web Inf. Syst. Eng. (WISE)*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, 2020, pp. 503–515.

[47] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.

[48] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2015, pp. 1200–1205.

[49] K. Z. Mao, "Identifying critical variables of principal components for unsupervised feature selection," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 35, no. 2, pp. 339–344, Apr. 2005.

[50] M. Ayyanar, S. Jeganathan, S. Parthasarathy, V. Jayaraman, and A. R. Lakshminarayanan, "Predicting the cardiac diseases using SelectKBest method equipped light gradient boosting machine," in *Proc. 6th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2022, pp. 117–122.

[51] M. Al Fatih Abil Fida, T. Ahmad, and M. Ntahobari, "Variance threshold as early screening to Boruta feature selection for intrusion detection system," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, Oct. 2021, pp. 46–50.

[52] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 14, 1995, pp. 1137–1145.

[53] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[54] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, 2002.

[55] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.

[56] P. Flach and M. Kull, "Precision-recall-gain curves: PR analysis done right," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 838–846.

[57] S. Ekici, F. Ucar, B. Dandil, and R. Arghandeh, "Power quality event classification using optimized Bayesian convolutional neural networks," *Electr. Eng.*, vol. 103, no. 1, pp. 67–77, Feb. 2021.

[58] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432.

[59] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.

[60] J. Lv, M. Pawlak, U. D. Annakkage, and B. Bagen, "Statistical testing for load models using measured data," *Electr. Power Syst. Res.*, vol. 163, pp. 66–72, Oct. 2018.

[61] M. Middlehurst, W. Vickers, and A. Bagnall, "Scalable dictionary classifiers for time series classification," in *Proc. 20th Int. Conf. Intell. Data Eng. Automated Learn. (IDEAL)*, Manchester, U.K. Cham, Switzerland: Springer, Nov. 2019, pp. 11–19.

[62] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," 2017, *arXiv:1702.01460*.

[63] G. Vrbančič, L. Brezočnik, U. Mlakar, D. Fister, and I. Fister Jr., "NiaPy: Python microframework for building nature-inspired algorithms," *J. Open Source Softw.*, vol. 3, no. 23, p. 613, Mar. 2018.

[64] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, and R. Kern, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[65] P. L. Gilabert, M. E. Gadringer, G. Montoro, M. L. Mayer, D. D. Silveira, E. Bertran, and G. Magerl, "An efficient combination of digital predistortion and OFDM clipping for power amplifiers," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 19, no. 5, pp. 583–591, Sep. 2009.

[66] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[67] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Apress, 2019.

[68] S. Asefi, M. Mitrovic, D. Ćetenović, V. Levi, E. Gryazina, and V. Terzija, "Power system anomaly detection and classification utilizing WLS-EKF state estimation and machine learning," 2022, *arXiv:2209.12629*.

SOLOMON OLUWOLE AKINOLA received the Bachelor of Computer Engineering degree from the Ladoke Akintola University of Technology and the master's degree from the University of Ibadan. He is currently pursuing the Ph.D. degree with the University of Johannesburg. His research interests include the number of disciplines, including machine (deep) learning and mathematical modeling.

PETER OLUKANMI received the B.Sc. degree in systems engineering from the University of Lagos, the M.Sc. degree in computer science from the University of KwaZulu-Natal (UKZN), and the Ph.D. degree from the University of Johannesburg. His research interests include fundamental and applied data science and mathematical modeling. He won two IEEE conference awards in soft computing and machine intelligence.

QING-GUO WANG received the B.Eng. degree in chemical engineering and the M.Eng. and Ph.D. degrees in industrial automation from Zhejiang University, China, in 1982, 1984, and 1987, respectively. He held the Alexander von Hummoldt Research Fellowship in Germany, from 1990 to 1992. From 1992 to 2015, he was with the Department of Electrical and Computer Engineering, National University of Singapore, where he became a full-time Professor, in 2004. He is currently a Distinguished Professor with the Institute for Intelligent Systems, University of Johannesburg, South Africa. He holds a rating from the National Research Foundation of South Africa (NRF). He has published nearly 300 international journal articles and seven research monographs. He received approximately 15000 citations with an H-index of 65. His current research interests include modeling, estimating, predicting, controlling, optimizing, and automating complex systems, including but not limited to industrial and environmental processes, new energy devices, defense systems, medical engineering, and financial markets. He is a member of the South African Academy of Sciences. He is currently the Deputy Editor-in-Chief of the *ISA Transactions* (USA).

TSHILIDZI MARWALA is currently a South African Mechanical Engineer and a Computer Scientist. He became a Professor with the University of the Witwatersrand, in 2003, and the Chairperson of System and Control Engineering in South Africa. Previously, he was with CSIR and South African Breweries. His research interests include the theory and application of artificial intelligence in engineering, computer science, finance, economics, social science, and medicine. He has made fundamental contributions to engineering science, including developing the concept of pseudo-modal energies and proposing the theory of rational counterfactual thinking, reasonable opportunity cost, and flexibly bounded rationality.