A Feature-subspace-based Ensemble Method for Estimating Long-term Voltage Stability Margins

Ambreen Khurram
Dept. of Electronic and Electrical Eng.
University of Leeds, UK
elakh@leeds.ac.uk

Arief Gusnanto
School of Mathematics
University of Leeds, UK
A.Gusnanto@leeds.ac.uk

Petros Aristidou
Dept. of Elec. & Comp. Eng. & Informatics
Cyprus University of Technology, Cyprus
petros.aristidou@cut.ac.cy

***Abstract*—This study proposes a methodology for online voltage stability monitoring using a feature subspace based ensemble approach. The overall idea is to use the input from varied feature selectors for the ensemble and aggregate their outputs. This approach is superior to conventional feature selection methods because it can handle stability issues that are usually poor in existing feature selection methods and improve performance. The selected features are used as an input to three different regression algorithms to enable online voltage stability monitoring. A Bayesian optimization technique is used to tune machine learning (ML) models' hyper-parameters and determine the optimal number of features. The proposed approach is evaluated in experiments using simulated data from the Nordic test system. The simulation results have shown that the proposed method efficiently predicts the status of dynamic voltage stability in the test system.**

***Index Terms*—Voltage stability; feature selection; machine learning, Bayesian optimization; regression methods**

I. Introduction

Power systems are evolving with increasing penetration of renewable energy resources and decreasing number of conventional units performing stabilizing control. As a result of increased demand, power systems are operating closer to their stability limits, and dynamic problems are more and more frequent. One of the urgent concerns in modern power systems is the early detection and prevention of on-coming stability problems, such as long-term voltage stability (LTVS). Long-term voltage instability manifests due to the inability of the combined generation and transmission system to supply the demand [1] and the risk increases with the lack of voltage-regulating units (conventional synchronous generators) and high network loading conditions.

Detecting an upcoming voltage instability as early as possible is crucial to designing online control algorithms for stabilizing the system. In recent years, the advent of Phasor Measurement Units (PMUs), installed in Transmission Networks (TNs), and the high-speed real-time measurements collected from the power system have allowed for the design of online instability detection algorithms based on power-engineering methods [2], mathematical methods [3] and machine learning (ML) methods [4–7]. [4] proposed a deep neural network to monitor voltage stability by utilizing massive unlabelled PMU data. [5] proposed online monitoring of voltage stability using the Weighted Least Square Support Vector Machine. [6] presented an ensemble machine learning model to predict the long-term voltage stability margin. [7] proposed a machine learning-based approach for real-time inference of voltage stability margin by using transfer learning.

Due to the large volume of data that needs to be collected and processed for the online detection, ML methods have shown the most promising results and will be the focus of this paper. The basis of most online LTVS monitoring algorithms is the ability to estimate accurately, efficiently, and quickly the system's voltage stability margin (VSM). Monitoring the VSM using ML techniques is generally a two-step approach: 1) train the ML models using intensive off-line computations based on historical and simulated data, and 2) use the trained models and real-time measurements for online monitoring of VSM.

Many works have explored the performance of ML algorithms for voltage stability analysis of the power system. References [8–10] attempted to set up a direct mapping from the operating states to VSM using the supervised neural networks. These works use artificial neural networks (ANN) to explore the relationship between the voltage stability indicator and power system parameters affecting those indicators. References [11, 12], proposed using a classification and regression tree (CART) algorithm for voltage stability assessment. Reference [13] proposed a Random Forest model for monitoring the voltage stability based on drift detection and online bagging techniques. However, most of these techniques emphasize a large number of input variables, which poses a significant challenge to applying ML algorithms. High-dimensional data from PMUs may result in model over-fitting, irrelevant and redundant features, an increase in the model's search space size, and adverse processing time. Moreover, data from only a few nodes might be available due to economic constraints.

Recently, references [14–17] considered voltage stability assessment using machine learning algorithms with reduced inputs. A critical aspect of these methods is to remove the redundant features for estimating the VSM using principal component analysis (PCA). PCA is a feature extraction technique that aims to reduce the number of features in a dataset by creating new features from the existing ones [18]. However,

Submitted to the 22nd Power Systems Computation Conference (PSCC 2022).

PCA is most useful for linear models while the relationships between variables in the power system are often non-linear. Furthermore, the new features from PCA are sometimes less interpretative.

An alternative to feature extraction is a feature subset selection. The technique focuses on selecting a subset of variables to represent the entire dataset and provide good prediction performance efficiently. Nevertheless, to make the correct choice of features, a user needs to know the domain well and is also expected to understand the technical details of available algorithms. A more realistic approach is considered in the present work by using an ensemble feature selection (EFS) method. In the EFS, multiple feature subsets are combined to select an optimal subset of features by aggregation [19]. The proposed methodology discards the redundant variables while selecting the most relevant features that best describe the studied phenomena. Ensemble learning is based on the postulation that combining the output of multiple models is better than using a single model. Usually, it is employed for classification problems, but it can be also be extended to other disciplines such as feature selection (FS) [20].

Furthermore, a ML ensemble is tuned by a set of hyper-parameters. Selecting the best hyper-parameter configuration for ML models has a direct impact on the model's performance. Often, FS and hyper-parameter optimization are carried out in separate steps. In this study, we simultaneously optimize the model's hyper-parameters and fine-tune the base learner to automatically select optimal number of features. The objective is to explore the joint space concurrently as it is computationally less expensive.

The main contributions of this work are:

- Generation of training data by using load forecasting method based on density-based spatial clustering of applications with noise (DBSCAN).
- Propose an automated FS technique to identify the best features for improved model performance in LTVS monitoring algorithms.
- Use FS techniques as a diversity method to build a heterogeneous ensemble model, i.e., using different FS methods with the same training data.

The rest of the paper is organized as follows. In Section II, the problem formulation is discussed. In Section III, we provide the theoretical background and architectural structure of the proposed scheme. In Section IV and Section V, we validate our assumptions through simulations and show the benefits of our solution over existing ones. Finally, in Section VI, we draw the conclusions.

II. Problem formulation

The main objective of online voltage stability analysis (VSA) is to determine whether the current operating point of the power system is stable and meets various operational criteria [21]. Voltage stability is often assessed through Power-Voltage (P-V) analyses [1]. The P-V curve of the system (see Fig. 1), in combination with the current operating conditions, can be used to obtain the VSM. The P-V curve is usually calculated by simulating a continuous load growth, starting from a pre-defined operating point until the system simulation collapses. The point of collapse is the maximum loading point (represented as P_{max} in Fig. 1). The initial state is illustrated in the figure by P_0. The VSM [22] can then be represented by:

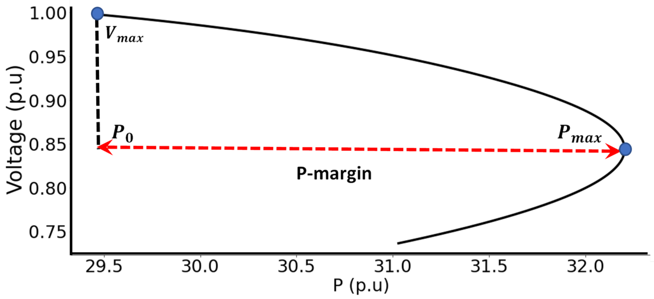

Fig. 1. A P-V curve showing VSM.

$$\lambda = \frac{P_{max} - P_0}{P_{max}} \geq 0 \tag{1}$$

A larger λ value indicates a stable system while a decreasing value suggests closeness towards voltage instability [23].

It should be noted that different system topologies and operating conditions produce different P-V curves. In addition, faults occurring in the system that can lead to topology changes (e.g., line tripping) or changing operating conditions (e.g., generator tripping) can significantly affect the P-V curve and the value of λ for the current operating condition.

Calculating the VSM using conventional methods, like the continuation power flow (CPF), can be time-consuming (especially in large-scale systems) and thus ineffective for *online* monitoring solutions. On the contrary, ML-based techniques can be used to estimate the VSM almost instantaneously but require heavy *offline* computations for model training, FS, and parameter tuning.

This paper employs a promising ML technique that uses a feature subspace method for ensemble learning to fit a model on different groups of randomly selected features in the training dataset. The aim of the EFS is to capture the knowledge mapping between the input-output pairs generated in offline simulations. In our case, the inputs are the system measurements (provided by the PMUs), such as the bus voltage magnitude and phase angle, and the output is the VSM indicator λ. Then, the method is used during the online operation to estimate the VSM indicator λ.

III. Proposed Methodology

The proposed method consists of 3 stages: A) data generation, B) training, and C) online-update and assessment. Fig. 2 provides an overview while the details are given below.

A. Data Generation Stage

At this stage, the training dataset of P-V curves is generated based on different operating points and fault conditions (contingencies). The consumption patterns from the retailer set

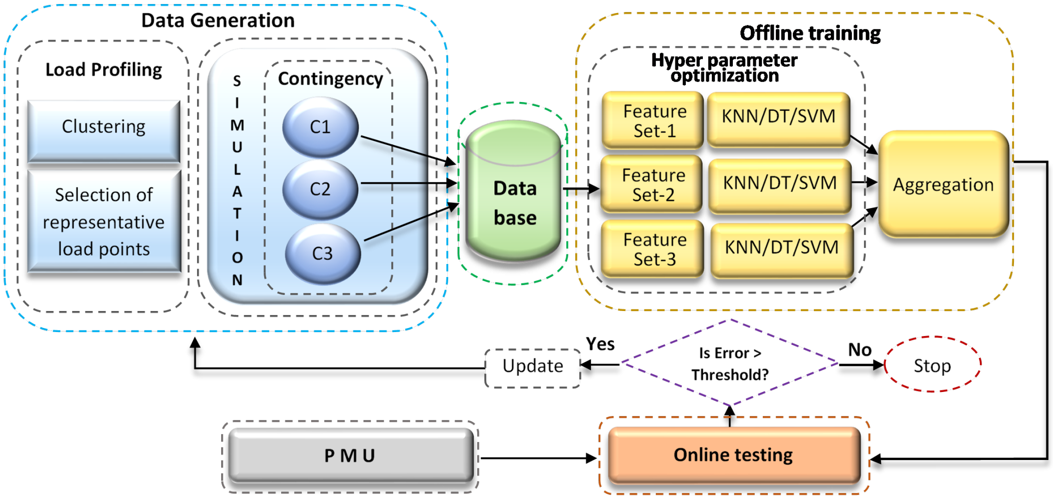

Fig. 2. An overview of the proposed framework with three stages: 1)Data generation 2) Offline Training and 2) Online Prediction and Update

of hourly customer load readings over twelve months period is used to extract load operating points. The yearlong load profile W is first separated in a continuous sequence of daily load profiles L_d built from hourly load data $H_{d,x}$.

$$\begin{aligned} W &= [L_1, L_2, \ldots, L_d] \\ L_d &= [H_{d,1}, H_{d,2}, \ldots . H_{d,24}] \end{aligned} \tag{2}$$

The feature vectors L_d can then be fed to Density-based spatial clustering of applications with noise (DBSCAN) algorithm [24] to group the load profiles based on similarity.

1) DBSCAN: creates a circle of epsilon (ϵ) radius around every data point and the neighborhood $N(p)$ of the data-point within the circle has to contain at least minimum number of points (*MinPts*) to qualify as a dense region. The shape of the neighborhood is determined by the choice of distance function for two points p and q denoted by $dist(p,q)$ in a dataset D. In distinguishing regions of data by density, the following terms are used [24]:

- **ϵ-neighborhood**: The ϵ-neighborhood of a point p, is defined as $N(p) = \{q \in \mathbf{D} \mid dist(p,q) \leq \epsilon\}$. The neighborhood represent the collection of points whose distance from p is less than or equal to ϵ. The cardinality of ϵ-neighborhood defines the threshold density of p.
- **Core point**: A point is a core point if its ϵ-neighborhood contains at least *MinPts* points.
- **Border point**: A point is a border point if it has less than *MinPts* in its ϵ-neighborhood, but it lies in the neighborhood of another core point.
- **Directly density reachable**: A point p is directly density reachable from a point q w.r.t ϵ, *MinPts* if:
  1) $p \in N(q)$ and
  2) $N(q) \geq MinPts$
- **Density connected**: A point p is density connected to a point q w.r.t ϵ and *MinPts* if there is a point o such that both p and q are density reachable from o w.r.t ϵ and *MinPts*.
- **Density reachable**: A point p is density reachable from a point q w.r.t ϵ and *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density reachable from p_i.
- **Cluster**: A cluster C w.r.t ϵ and *MinPts* satisfy the following conditions:
  1) $\forall\ p, q$ if $p \in C$ and q is density reachable from p w.r.t ϵ and *MinPts*, then $q \in C$.
  2) $\forall\ p, q \in C$: p is density connected to q w.r.t ϵ and *MinPts*.
- **Noise**: A set of points in a database D that do not belong to any cluster C.

DBSCAN begins with an arbitrary starting point that has not been observed and discover its ϵ-neighbourhood. If the ϵ-neighbourhood of a point is greater than or equal to *MinPts*, it is considered as a dense point and a density connected cluster is formulated. Otherwise, the point is labelled as noise.

2) Cluster Validity Index: A cluster validity index (CVI) provides information about the quality of grouping. In this study, we have used the silhouette index [25], which is calculated by taking into account the mean intra-cluster distance $\mathbf{u}$ and the mean nearest cluster distance $\mathbf{v}$ for each data point. The silhouette coefficient for a given sample is defined as

$$S_C = \frac{v - u}{max(u, v)} \tag{3}$$

where the score is in the range of $[-1, +1]$. A score with a value near +1 means the data point is in the correct cluster, a score near 0 means the data point might belong in some other cluster, and a score with a value near -1 means the data point is in the wrong cluster.

3) Contingencies: Two major contingencies are considered a) a three-phase short circuit in a transmission line and b) generator outage. Details are provided in Section IV.

B. Ensemble Training

The objective of the EFS is to implement the input/output mapping function during the offline training. There are two important steps in formulating a feature subspace ensemble. The first step involves creating a set of feature selectors, each providing a target VSM indicator. The second step aggregates the model's results by calculating the average of the member predictions.

In this work, we aim at directly achieving the data diversity by formulating a heterogeneous ensemble with three FS methods, namely ANOVA (F-Test), Variance (Threshold) and Lasso regression. The first two FS algorithms can be classified as filter methods, while the latter is classified as an embedded methods. Filter methods rank features by calculating a score for each feature using various statistical metrics. The objective is to determine the strength of the relationship between a feature and the target variables [26]. Embedded methods perform the FS during the learning process, then derive feature importance from this model, which measures the extent of a feature when making a prediction [27].

The method is validated using three different regressor at a time namely, 1) Classification and Regression Tree (CART) 2) K Nearest Neighbor (KNN) and 3) Support Vector Regressor (SVR) on network dataset. A brief mathematical background of FS techniques are provided below

1) ANOVA: Suppose that the number of generated datasets $\mathbf{G}$ from the extracted operating points is a. Each dataset consists of m variables, here $m = 154$ and include voltage magnitude and phase angle. The raw data containing n samples is stacked into a matrix $\mathbf{X}^r \in \mathbb{R}^{n_a \times p}$, $n_a = n \times a$. It is first normalized to a matrix $\mathbf{X}$ with elements x_{ij}, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$ and correspond to voltage observation and voltage variables respectively. Each data matrix $\mathbf{X}$ is associated with the target output y_i, the vector of labels is $\mathbf{y} = (y_1, y_2, \ldots, y_n)$.

To apply ANOVA as a FS method, the column values in the data matrix $\mathbf{X}$ would be treated as a group. ANOVA is a statistical method for comparing means to determine if there is a statistically significant difference between the corresponding groups. The objective of ANOVA is to find F-ratio, which can be defined as between-group variance over within-group variance. The between-group variance is calculated as

$$\sigma^2_{between} = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (\bar{x}_j - \bar{x})^2 = n_j \sum_{j=1}^{p} (\bar{x}_j - \bar{x})^2 \qquad (4)$$

Where n is the sample size of group j, $\bar{x}_j$ is the mean of the group j and $\bar{x}$ is the overall mean. The within group variance is calculated as:

$$\sigma^2_{within} = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (x_{ij} - \bar{x_j})^2 \qquad (5)$$

Where x_{ij} is the i-th measurement of the j-th group. An F-ratio is then calculated as the ratio between the two variances:

$$F = \frac{\sigma^2_{between}}{\sigma^2_{within}} \qquad (6)$$

Then, the p-value based on F-statistic is calculated as p-value $= Prob\{F(j-1, n-j) > F\}$, where $j-1$ and $n-j$ are the degrees of freedom. Features are ranked by sorting p-value in ascending order. The magnitude of the F-ratio shows the group separation. Features with an F-ratio greater than a specified threshold are retained while those below the threshold are removed.

2) Variance Threshold: A simple and effective method for FS is motivated by the idea that low-variance features contain less relevant information and less value in predicting the response variable. It calculates the variance of each feature and removes those with a variance less than a given threshold. The variance threshold is an unsupervised method that looks only at the feature values ($\mathbf{X}$) and not the desired output ($\mathbf{y}$). In the case of variance thresholding, the only hyper-parameter to be tuned is the threshold value of the variance.

3) Lasso: The least absolute shrinkage and selection operator (Lasso) [28] performs two main tasks: regularization and FS. Regularization helps in reducing errors and overfitting, while FS eliminates unimportant variables that are not associated with the response variables. For multiple linear regression, the response variable y_i is often influenced by two or more explanatory variables X. This relationship can be expressed as:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots, + \beta_p x_p + \epsilon_i \qquad (7)$$

Where the parameters $\beta_0, \beta_1, \ldots, \beta_p$ are regression. Moreover, ϵ_i is the error term providing random variation in y_i not explained by X variables. The objective is to optimize β and ϵ to minimize the cost function. In Lasso regression, the cost function is altered by adding a penalty:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} \beta_j - x_{ij} \right)^2 + \delta \sum_{j=0}^{p} |\beta_j| \quad (8)$$

Where $\hat{y}_i$ is the predicted value and δ denotes the amount of shrinkage. All the features are considered when $\delta = 0$, and equation (8) becomes equivalent to the linear regression. On the other hand, as δ approaches ∞, more and more features are eliminated. The bias increases with increases in δ, and variance increases as δ decreases. The regularization process updates the coefficient values of the regression variables by reducing few to zero, meaning that it can nullify the impact of irrelevant features in the data [29].

C. Bayesian Optimization

Bayesian hyperparameter optimization (BHO) searches for the best hyperparameter on the domain space Φ by using Bayesian optimization [30]. For a given PMU data of training set and validation set $G = \{G_{train}, G_{val}\}$ we train a FS model involving hyperparameter vector ϕ. The best hyperparameter vector is determined by minimizing the validation error $E(\phi, G_{train}, G_{val})$. Generally there are three inputs to

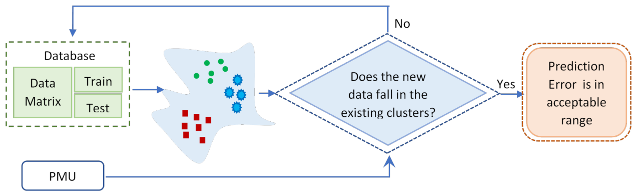
Database
Data Matrix
Train
Test
Does the new data fall in the existing clusters?
No
Yes
Prediction Error is in acceptable range
PMU

Fig. 3. Diagram of database update scheme.

BHO [31] , (i) a target function $E(\phi, G_{train}, G_{val})$ which determines validation error or classification accuracy based on the hyperparameter vector and training/validation datasets, (ii) h different hyperparameter vectors $\phi_s = (\phi_1^* \dots, \phi_h^*)$ and (iii) a limit L which specifies the number of candidates of hyperparameter vectors to search the best configuration. The BHO searches a minimum, gradually accumulating $(\phi_s^*, E_1(\phi_s^*))$ with s increasing and returns the best configuration of hyperparameters ϕ^b. Using a predictive distribution, the BHO guides the search to only focus on the areas of the input space that are expected to provide the most useful information about the solution to the optimization problem. Starting with a set of initial hyperparameter vectors $\{(\phi_1^*, E_1), \dots, (\phi_h^*, E_h)\}$, a surrogate function model $F_{surrogate}$ is fitted to the data with the accumulated set of hyperparameter vector and its corresponding validation error. In this paper, the Gaussian process (GP) regression model F_G serves as a surrogate function that approximates the landscape of E over the space Φ. BHO utilizes all the information in the history (reflected by the built surrogate model) to determine what will be sampled next. Thus, next hyperparameter is sampled at the place optimizing an acquisition function $A(\phi|F_G)$ at which the validation error E is evaluated. More information about the BO can be found in [32].

D. Database update condition

The methodology used to generate the data for training cannot include all the possible operating conditions and grid topologies. It is possible that in real-time operation, the system will drift to operating conditions far from the ones considered in the training stage. Therefore the initial database requires updating to adapt to the dynamic operating environment. This study implements the database update scheme using a K-means clustering approach. The entire process is shown in Fig. 3. K-means clustering is applied to the database, the distances between each point to the cluster centers are calculated, and a threshold ratio is specified. The distance is measured again for each new data point that arrives from the PMUs during the online monitoring. If the distance is within the specified threshold, the latest data belongs to one of the existing clusters, and the prediction accuracy is not much affected. On the other hand, if the distance of new data is above the threshold, the operating point is added back to the dataset, and the training process is restarted offline.

1) K-means clustering: K-means partitions the n-dimensional data points $\mathbf{X} = [x_1, x_2, x_3, \dots, x_p]$ into set of clusters $\{C_1, C_2, C_3, \dots, C_k\}$ where $k \leq p$. The algorithm finds a partition such that the squared error between empirical mean of a cluster and the points in the cluster is minimized:

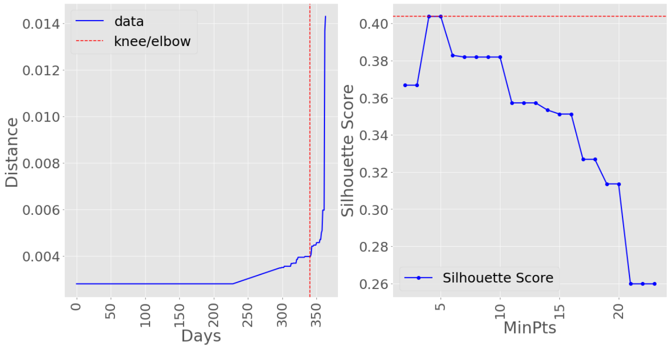

Fig. 4. The k-nearest neighbor distances of optimal ϵ Silhouette Index of the corresponding *MinPts*.

$$\underset{C}{\operatorname{argmin}} \frac{1}{p} \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \tag{9}$$

Where μ_i is the mean of points in C_i. The K-means can deal with clustered and not clustered data appropriately, and the selected load objects pronounce well the data distribution.

E. Performance Evaluation

The evaluation of the feature ensemble model was undertaken with the aid of two statistical measures. The first measure is R^2, defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

The second measure is the mean squared error, given as $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Where y and $\hat{y}$ represent the actual and the expected outputs respectively and n denotes the number of observations.

IV. Case Study

To generate the training and testing data for the proposed algorithm, dynamic simulations were performed using the IEEE Nordic test system [33] depicted in Fig. 5. The 74-bus test network consists of 20 generator buses (with nominal voltages of 15 kV), 32 transmission buses (with nominal voltages of 400 kV, 220 kV, and 130 kV) and 22 buses at the distribution level (with a voltage of 20 kV). The transmission lines from the North to Central Area form the power transfer corridor, with sending buses 4021, 4031, and 4032 and boundary receiving buses 4041, 4042, and 4044. Simulation is performed with PyRAMSES [34] software.

A. Data generation

1) Extraction of load operating points: DBSCAN algorithm is used to extract the load operating points from the historical electricity consumption load profile data [35]. The load profile dataset contains 8760 operating points (365 days times 24 hourly consumption data) for 2018. Regarding the

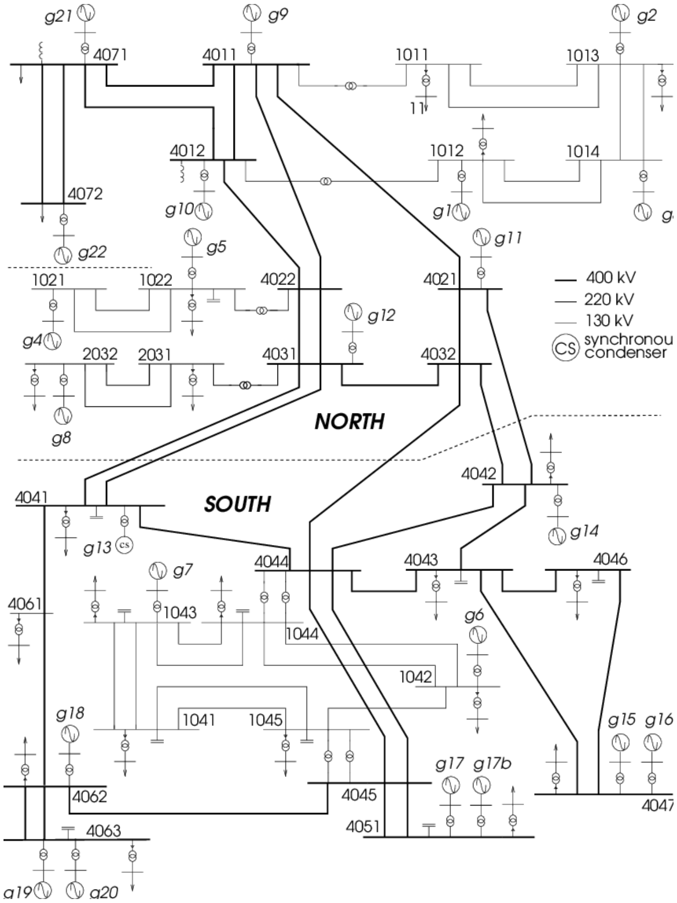

Fig. 5. One-line diagram of Nordic test system

DBSCAN algorithm, the difficulty lies in choosing proper values for parameters ϵ and *MinPts*.

We have used the nearest neighbors method to reach a fair estimation for ϵ. The technique calculates the average distance between each point and its k nearest neighbor. Fig. 4-(a) depicts the result of a distance plot of the load profile data points sorted in ascending order to the 20-th nearest neighbor. The angle that is bent the most is selected as the ϵ value and is found to be 0.005. Once the ϵ value is found, the CVI was applied to find the *MinPts*. Fig. 4-(b) shows the results of the application of CVI iterated for different sample sizes. The red dotted line shows the maximized scores for the Silhouette index. Based on the score, we can choose a minimum sample size of 4 or 5. With ϵ=0.005 and $MinPts$=5, ten clusters were obtained. The clustering results of the method in this paper are shown in Fig. 6. The representative members from the clusters were obtained using mean, minimum, and maximum. After clustering 8760 profiles with DBSCAN, 720 representative load operating points were extracted.

2) Dynamic Simulations: Starting from 720 load operating points, dynamic simulations were carried out to generate the corresponding P-V curves. At the beginning of the simulation, the loads were set to the initial operating point P_0. The loads in the system were uniformly scaled up while the power factor

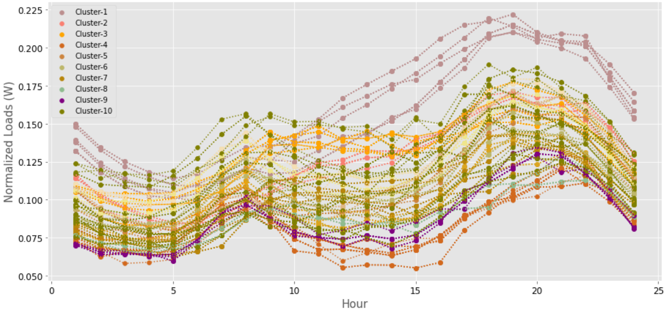

Fig. 6. Load profile clusters from DBSCAN

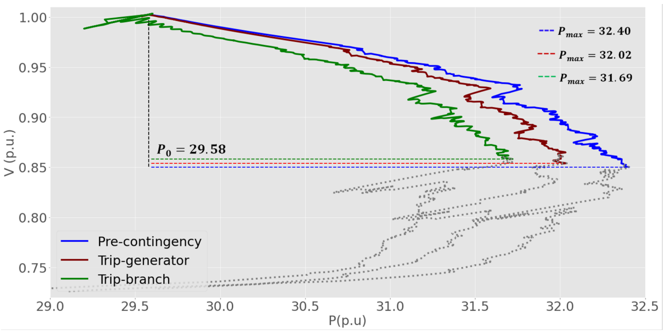

Fig. 7. P-V curves for load increase in the Central area. The blue curve shows when the system operates without a contingency, while the red and green curves show the system with contingencies.

was kept constant.

3) Contingencies: Two contingencies, namely Fault-1 and Fault-2, are considered to investigate the impact on the P-V curve. Fault-1 is a short-circuit of 100 ms on transmission lines 4031-4041. This transmission line pass between the 'North' and the 'Central' area and is close to bus 4022. The fault is cleared by tripping the line. Fault-2 takes place near bus 4012 in the "North" region by tripping generator g10.

Given the 720 operating states, the two contingencies are simulated, yielding two more datasets. The generated datasets are stored in a database with input vectors comprising voltage magnitude and phase angle of all the buses. The output vector/target is the P_{Margin}.

Fig. 7 show three P-V curves before and after the contingencies. The blue P-V curve represents the base case, while the red and green P-V curves represent fault-2 and fault-1. In Fig. 7 the blue curve corresponds to the load bus '1041' and its VSM indicator is $\lambda = 0.0870$. At the initial point P_0, the baseload active power is 29.58 pu. At the critical point ($\lambda = 0.0870$), the maximum load active power is 32.40 pu. Therefore for this initial operating point, the load $P_{\text{Margin}} = 2.82$ pu.

The load margin for the post-contingency P-V curves is smaller than the pre-contingency curve (blue) because the topology and characteristics of the system changed after the disturbance (contingency) in the network at $t = 1s$.

V. Offline training

All the datasets are combined into one big data matrix for training and testing purposes. After pre-processing, there are

TABLE I
OPTIMAL VALUE OF HYPERPARAMETERS FOR FS ALGORITHMS

	ANOVA	**Variance Threshold**	**Lasso**
Hyperparameter	k	T	delta
Optimal value	22	0.6	0.004
No of features selected	22	24	30

154 voltage variables and 208646 observations in the dataset. The pre-processing steps consist of checking and removing any missing values, eliminating duplicates, converting the phase angles from degrees to radians, and feature scaling. Additionally, the dataset is randomly divided into two independent sets, where 70% of the data is allocated to the training set, and the remaining 30% is assigned to the test set. Out of 720 operating points, 40 operating points were kept aside for online testing.

A. Feature selection

In the proposed method, offline computation includes model training and aggregation of the ensemble outputs. ML models KNN, CART, and SVR are applied one at a time to measure the algorithm's performance. The number of features selected for ensemble learning is based on the optimal value of the hyperparameters. The optimal hyperparameter was obtained using the BO technique. The parameter settings utilized in the proposed algorithm are also shown in Table I. For ANOVA, the tuned hyperparameter is k, representing the number of optimal features based on the F-score. For the Variance threshold, the optimal feature is determined with threshold τ. Similarly, δ is the regularisation parameter for Lasso. Based on hyperparameter optimization, the optimal number of features selected are 22,24 and 30 for ANOVA, variance threshold, and Lasso, respectively. Table III shows the selected features from the offline analysis, the features with (a) represent phase angle.

Experimental results are reported in Table II and shown in Fig. 8. Each value presented in the table is the average over the three runs of three-fold cross-validation outcomes. We also show the RMSE and R-squared for each model. We refer to the base regressors KNN, SVR, and CART in the EFS as EKNN, ESVR, and ECART. The results demonstrate that the FS ensembles with base regressor CART and KNN outperformed 4 out of 6 cases. The MSE of the standalone models KNN, SVR, and CART are higher than the top-performing ensemble techniques.

B. Online Evaluation and Database update

The FS ensemble is applied online to monitor and inspect new situations during the online evaluation stage. The database update phase runs parallel with the online assessment to include the latest system conditions. In this phase, the k-means clustering method is carried out on the training data set resulting from the previous stage. Four time-domain simulations of online VSM estimations are presented in Fig. 9. The plot shows the result of three FS-ensemble models by substituting the base learners with KNN, CART, and SVR. The simulations were carried out without increasing the load factor.

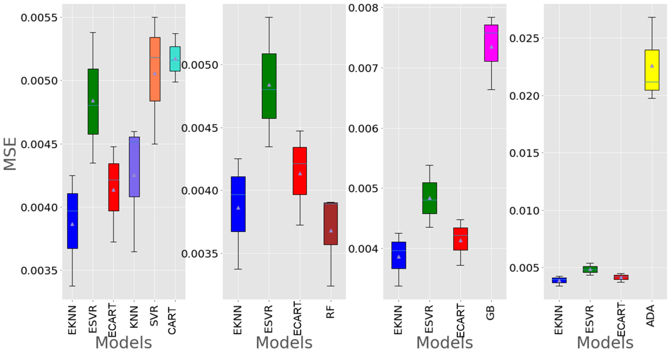

Fig. 8. A box plots of Mean Square Error (MSE) of the feature ensemble using three regression algorithms (EKNN,ECART and ESVR).

TABLE II
PERFORMANCE EVALUATION OF THE FS-ENSEMBLE

Models	MSE	RMSE	R-squared	time
EKNN	0.0039	0.062	0.997	1min 05s
ESVR	0.0048	0.070	0.996	2h 42min
ECART	0.0041	0.064	0.997	1min 21s
KNN	0.0043	0.065	0.997	1min 39s
SVR	0.0051	0.071	0.995	4h 37min
CART	0.0052	0.072	0.997	2min 15s
RF	0.0037	0.061	0.997	32min 9s
GB	0.0074	0.086	0.993	17min 40s
ADA Boost	0.0216	0.229	0.943	4min 36s

In the first simulation, the system initially operated under normal conditions. The estimated VSM remains constant, around 2.8 pu to 2.825 pu for all three ensemble models. Next, fault-1 is simulated. For this case, all three FS-ensemble models show a declining trend for the estimated VSM. A warning will be given when the VSM crosses a threshold set by the system operator. In the subsequent simulation, fault-2 is applied. We can notice a significant dip in the estimated VSM that recovers after 40 seconds. In the last case, a new contingency is introduced by tripping the branch 4012-4022 close to bus 4031 at $t = 1s$. The FS-ensemble model was not trained previously on this fault condition. The newly measured data point distance is measured from the existing cluster centroids. The average distance and the confidence interval of each cluster member from the cluster centroid are used as a benchmark. If the distance of new data falls between the two confidence intervals, the point is considered similar to the existing clusters. Otherwise, the operating point is put back in the database for retraining. The existing dataset is updated, and the newest incremental samples are injected into the initial dataset. Subsequently, the updated datasets would, in turn, serve as the initial database for the next update period.

VI. CONCLUSION

This paper proposes a methodology to monitor large power systems for LTVS by selecting relevant features from the PMU data. The proposed scheme is implemented in three stages; data generation, offline training, and online monitoring. In the data generation stage, DBSCAN clustering is applied to

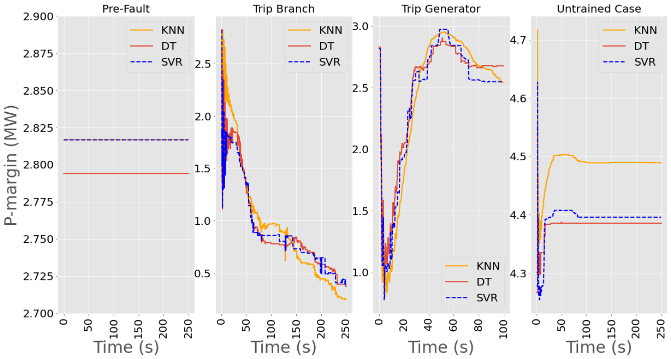

Fig. 9. Evolution of the predicted VSM during the online testing. The first simulation represents a case without contingency. In the second simulation, a critical transmission line 4021-4032 is tripped. For the third case generator 'g10' is tripped and the last case represents an unknown contingency not included during the training phase.

TABLE III
SELECTED FEATURES FROM OFFLINE ANALYSIS

Variance	1041(a)	1042(a)	1043(a)	1044(a)	1045(a)
	2(a)	3(a)	4045(a)	4046(a)	5(a)
	g16b(a)	g6(a)	g7(a)	4051(a)	43(a)
	1(a)	g16(a)	46(a)		
Anova	1041	1043	1044	2031	22
	4022	4031	4032	4041	4045
	4072(a)	71(a)	72(a)	g19(a)	g20(a)
	4021	4071(a)			
Lasso	1041(a)	1(a)	g2	g4	g6
	g7	g7(a)	g6(a)		

smart meter electricity demand data to identify the patterns in electricity consumption for load forecasting. The extracted load operating points were used to simulate the data and generate the P-V curve. The loadability limit from the P-V curve is considered as the stability margin. An ensemble filter-embedded feature ranking scheme is proposed that selects the optimal features from the training dataset. The performance of the FS ensemble was evaluated and compared by conducting experiments with three different regression algorithms, namely KNN, CART and SVR. The results of the VSA showed that the CART and KNN had the best performance as they achieved a small mean squared error. Furthermore, to cater to the changing system topology, we developed an adaptive database update model that measures the distance of the new data arrivals with the existing cluster centroids. The results of case studies using the Nordic test system illustrate the effectiveness of the proposed scheme.

REFERENCES

[1] T. Van Cutsem and C. Vournas, *Voltage stability of electric power systems*. Boston, MA: Springer US, 2008.

[2] P. Mandoulidis and C. Vournas, "A PMU-based real-time estimation of voltage stability and margin," *Electric Power Systems Research*, vol. 178, pp. 1–12, jan 2020.

[3] G. K. Morison, B. Gao, and P. Kundur, "Voltage stability analysis using static and dynamic approaches," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1159–1171, 1993.

[4] T. Wu, Y. J. A. Zhang, and H. Wen, "Voltage Stability Monitoring Based on Disagreement-Based Deep Learning in a Time-Varying Environment," *IEEE Transactions on Power Systems*, vol. 36, pp. 28–38, jan 2021.

[5] A. H. Poursaeed and F. Namdari, "Real-time voltage stability monitoring using weighted least square support vector machine considering overcurrent protection," *International Journal of Electrical Power & Energy Systems*, vol. 136, pp. 1–18, mar 2022.

[6] K. D. Dharmapala, A. Rajapakse, K. Narendra, and Y. Zhang, "Machine Learning Based Real-Time Monitoring of Long-Term Voltage Stability Using Voltage Stability Indices," *IEEE Access*, vol. 8, pp. 222544–222555, 2020.

[7] J. Li, "Effect of Excitation Regulation on Power System Stability," in *Design and Application of Modern Synchronous Generator Excitation Systems*, pp. 67–75, Singapore: John Wiley &Sons Singapore Pte. Ltd, mar 2019.

[8] A. A. El-Keib and X. Ma, "Application of artificial neural networks in voltage stability assessment," *IEEE Transactions on Power Systems*, vol. 10, pp. 1890–1896, nov 1995.

[9] V. R. Dinavahi and S. C. Srivastava, "ANN based voltage stability margin prediction," *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, vol. 2, pp. 1275–1279, 2001.

[10] C. Bulac, I. Triştiu, A. Mandiş, and L. Toma, "On-line power systems voltage stability monitoring using artificial neural networks," *2015 9th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2015*, pp. 622–625, jun 2015.

[11] X. Meng, P. Zhang, Y. Xu, and H. Xie, "Construction of decision tree based on C4.5 algorithm for online voltage stability assessment," *International Journal of Electrical Power & Energy Systems*, vol. 118, pp. 1–8, jun 2020.

[12] L. Vanfretti and V. S. Arava, "Decision tree based classification of multiple operating conditions for power system voltage stability assessment," *International Journal of Electrical Power and Energy Systems*, vol. 123, pp. 1–10, dec 2020.

[13] H. Su and T. Liu, "Enhanced-online-random-forest model for static voltage stability assessment using wide area measurements," *IEEE Transactions on Power Systems*, vol. 33, pp. 6696–6704, nov 2018.

[14] H. Shah and K. Verma, "Voltage stability monitoring by different ANN architectures using PCA based feature selection," *2016 IEEE 7th Power India International Conference,*, pp. 1–6, oct 2017.

[15] P. Duraipandy and D. Devaraj, "On-line voltage stability assessment using least squares support vector machine with reduced input features," *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, pp. 1070–1074, dec 2014.

[16] A. Jiménez C. and C. A. Castro, "Voltage stability security margin assessment via artificial neural networks," *2005 IEEE Russia Power Tech, PowerTech*, 2005.

[17] B. Jeyasurya, "Artificial neural networks for on-line voltage stability assessment," in *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, 2000.

[18] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proceedings of 2014 Science and Information Conference, SAI 2014*, 2014.

[19] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: Homogeneous and heterogeneous approaches," *Knowledge-Based Systems*, vol. 118, pp. 124–139, feb 2017.

[20] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Information Fusion*, 2019.

[21] K. S. Sajan, B. Tyagi, and V. Kumar, "Genetic algorithm based artificial neural network model for voltage stability monitoring," *2014 18th National Power Systems Conference, NPSC 2014*, may 2015.

[22] D. Q. Zhou, U. D. Annakkage, and A. D. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 25, pp. 1566–1574, aug 2010.

[23] A. C. Zambroni De Souza, F. W. Mohn, I. F. Borges, and T. R. Ocariz, "Using PV and QV curves with the meaning of static contingency screening and planning," *Electric Power Systems Research*, 2011.

[24] D. Rangaprakash, T. Odemuyiwa, D. N. Dutt, G. Deshpande, and A. D. N. Initiative, "Density-based clustering of static and dynamic functional MRI connectivity features obtained from subjects with cognitive impairment," *Brain Informatics*, vol. 7, dec 2020.

[25] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," *Proceedings - 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA 2020*, pp. 747–748, oct 2020.

[26] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, 2015.

[27] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, 2014.

[28] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, pp. 267–288, jan 1996.

[29] J. Lv, M. Pawlak, and U. D. Annakkage, "Prediction of the transient stability boundary using the lasso," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 281–288, 2013.

[30] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[31] J. Kim, S. Kim, and S. Choi, "Learning to Warm-Start Bayesian Hyperparameter Optimization," ., 2018.

[32] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization," *Journal of Electronic Science and Technology*, vol. 17, pp. 26–40, mar 2019.

[33] T. Van Cutsem, M. Glavic, W. Rosehart, C. Canizares, M. Kanatas, L. Lima, F. Milano, L. Papangelis, R. A. Ramos, J. A. D. Santos, B. Tamimi, G. Taranto, and C. Vournas, "Test Systems for Voltage Stability Studies," *IEEE Transactions on Power Systems*, vol. 35, pp. 4078–4087, sep 2020.

[34] P. Aristidou, D. Fabozzi, and T. Van Cutsem, "Dynamic simulation of large-scale power systems using a parallel schur-complement-based decomposition method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2561–2570, 2014.

[35] Https://www.nationalgridus.com/Upstate-NY-Business/Supply-Costs/Load-Profiles, "Load Profiles National Grid," 2018.