Online Transient Stability Assessment Under Concept Drift: An ARF Method Assisted Federated Learning for Data Streams

Mohamed Massaoudi[1,2], *Member, IEEE*, Maymouna Ez Eddin[1,2], Haitham Abu-Rub[3], *Fellow, IEEE*, Ali Gharayeb[3], *Fellow, IEEE*, And Katherine R. Davis[2], *Senior Member, IEEE*

***Abstract*—Transient instability poses a critical challenge to the reliable operation of modern power systems, often leading to large-scale blackouts. Despite the success of data-driven Transient Stability Assessment (TSA), its practical implementation remains limited by challenges in processing high-speed real-time data streams and preserving data privacy. To address these limitations, this article develops a novel Federated Adaptive Random Forest (FedARF) method that integrates federated learning with the Adaptive Random Forest (ARF) model. The proposed decentralized framework incorporates concept drift adaptation mechanisms to accommodate the stochastic and dynamic characteristics of modern power systems. FedARF facilitates distributed knowledge aggregation learned from various heterogeneous local data sensors (clients) to predict and evaluate the TSA status with minimal communication overhead. Comprehensive experiments on the New England 39-Bus system, the IEEE 68-Bus system, and the large-scale ACTIVIgs 25k-Bus system demonstrate the efficiency of the proposed method with an overall accuracy of 99.65%. Compared to traditional centralized forecasting methods, and state-of-the-art models, the proposed approach not only maintains high prediction accuracy but also enhances data privacy preservation while substantially reducing communication bandwidth requirements.**

***Index Terms*—Concept drift, data stream, federated learning, smart cyber-physical grids, transient stability assessment.**

NOMENCLATURE

Functions

L	Inference Latency
L_{avg}	Average Inference Latency
TSI	Transient Stability Index

Variables

$\delta_{\max}$	Maximum rotor-angle deviation
η	Learning rate
P	Active power
Q	Reactive power
V	Bus voltage vector
w	Model weight vector
x	State variable vector
N	Number of trees

Abbreviations

ADWIN	Adaptive Windowing
CCT	Critical Clearing Time
CD	Concept Drift
DAE	Differential Algebraic Equation
DDM	Drift Detection Method
EFDT	Extremely Fast Decision Tree
FedARF	Federated Adaptive Random Forest
FL	Federated Learning
HDDM_A	Hellinger Distance Drift Detection A-test
HDDM_W	Hellinger Distance Drift Detection W-test
IAda	Incremental AdaBoost
IL	Inference Latency
KSWIN	Kolmogorov–Smirnov Windowing
LSTM	Long-Short-Term Memory
NADINE	Neural Network with Dynamically Evolved Capacity
OANN	Online Artificial Neural Network
OTN	Online Transformer Network
OTSA	Online Transient Stability Assessment
PAC	Passive-Aggressive Classifier
PageHinkley	Page-Hinkley Test
PMU	Phasor Measurement Unit
PS	Power System
SEOA	Selective Ensemble-based Online Adaptive DNN
t-SNE	t-distributed Stochastic Neighbor Embedding

This publication was made possible by NPRP12C-33905-SP-220 from the Qatar National Research Fund (a member of Qatar Foundation), by the US Department of Energy under award DE-CR0000018, and by NSF EPCN Award 2220347. The open access funding is provided by Qatar National Library. The statements made herein are solely the responsibility of the authors.

Mohamed Massaoudi Maymouna Ez Eddin are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77840 USA, and also with the Department of Electrical Engineering, Texas A&M University, Doha, Qatar (e-mail: mohamed.massaoudi@qatar.tamu.edu).

Ali Ghrayeb and Haitham Abu-Rub are with the College of Science and Engineering, Hamad Bin Khalifa University, Doha 34110, Qatar (e-mail: aghrayeb@hbku.edu.qa; haburub@hbku.edu.qa).

Katherine R. Davis is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: katedavis@tamu.edu).

I. INTRODUCTION

THE increasing penetration of renewable energy sources has significantly altered the structure and dynamic behavior of Power Systems (PS). Typically, these resources are integrated through power electronic converters, which lack the inherent inertia of conventional synchronous generators. As a result, the PS's ability to maintain stability following disturbances has become harder to maintain. These fluctuations in output power and rapid changes in generator power angles can lead to transient instability and, in severe cases, system-wide blackouts. Therefore, an accurate and fast Transient Stability Assessment (TSA) has become essential to ensure

the secure and reliable operation of modern power grids [1]. Real-time TSA evaluates a PS ability to recover after a severe disturbance. TSA has received increasing attention to enable early detection of transient instability, allowing preventive control actions to be taken before loss of synchronism or widespread system failure [2].

Time-domain simulation methods, while accurate, often require hundreds of milliseconds to several seconds. As a result, they cannot meet the time constraints of real-time applications [3]. Direct methods, such as transient energy function approaches, are faster but typically still take tens of milliseconds. Even Phasor Measurement Unit (PMU)-based assessments, limited by measurement and communication delays, generally operate in the 20–100 ms range [4]. In contrast, recent advances in Machine Learning (ML) have demonstrated the ability to deliver highly accurate TSA results with inference times measured in milliseconds [5]. This ultra-fast response enables corrective actions to be initiated almost instantaneously following a disturbance, offering a critical advantage for power system security and reliability over conventional methods.

Online TSA (OTSA) is a real-time evaluation technique that continuously monitors the stability of the power grid during normal operation and in response to disturbances. This approach offers several advantages over offline and traditional methods, which primarily rely on static or pre-computed TSA [6]. While receiving limited attention in existing studies, OTSA features real-time monitoring, adaptability, faster response to contingencies, and enhanced situational awareness. By continuously monitoring the grid, operators can detect potential issues and take proactive measures before the system collapses [7]. Online methods can adapt to rapid changes in grid dynamics, offering an up-to-date assessment of grid stability.

However, OTSA faces several challenges, such as handling the high dimensionality of data, accurately modeling system dynamics, dealing with Concept Drift (CD), and ensuring computational efficiency [8]. CD occurs when TSA-based data analytics encounter regular shifts in data distribution. This is often prompted by the highly dynamic nature of traffic in edge cloud environments, necessitating constant adjustments to the ML model. There are four common types of CD: sudden drift, gradual drift, incremental drift, and recurring drift concepts [9]. CD adaptation is a significant challenge in OTSA, as the underlying relationships between input features and stability may change over time. Online assessment methods must be able to detect and adapt to these changes, often using adaptive or incremental learning techniques [10].

Data-driven approaches based on ML methods have gained substantial attention from the research community. For instance, Paper [11] proposed a transfer learning-based method for TSA. The method, tested on various PSs, demonstrated practical efficiency and scalability across different configurations. However, improvements in feature extraction techniques are required to handle thousands of variables. In [12], the authors proposed a data-driven Long-Short-Term Memory (LSTM) model for TSA in large-scale PSs. The model's architecture first employs leveraged time-delay neural networks. These networks identify primary buses for PMU placement and perform data-dimension reduction. It then uses bidirectional LSTM layers to produce precise TSA. An online self-check function is also added to ensure the validity of the OTSA. Paper [13] introduced an active transfer learning method for adaptive TSA using deep belief networks. The method combines active learning and transfer learning to reduce the time and cost of generating labeled samples and improve evaluation performance. The proposed method is effective in reducing computation costs and enhancing the adaptability of the model. Unfortunately, while these approaches provide insights into adaptive learning, none of these methods directly address CD in online data streams, particularly for TSA. The existing models do not adequately handle the dynamic environments in which the online data diverges from offline training data [14]. This results in tedious model updating and deployment tasks. To the best of the authors' knowledge, studies explicitly focusing on CD detection in TSA remain sparse in the literature, highlighting a significant research gap. The swift OTSA under-CD issue is accompanied by escalating privacy concerns related to electrical measurement data. These two problems have been decoupled due to the complexity of handling dynamic power flow behavior under the uncertainty of renewable energy sources.

In particular, to the author's best knowledge, this paper offers the first attempt to solve the aforementioned problem in a coupled approach. This approach computes OTSA and Federated Learning (FL) simultaneously. FL provides decentralized training, which preserves data privacy. In the meantime, Adaptive Random Forest (ARF) mitigates the issues arising from CD in real-time data streams. While prior works [11], [12], [13] have advanced TSA methods, they differ significantly from the proposed approach in CD handling. Existing approaches exhibit limitations: transfer learning lacks explicit drift detection (150-200 sample delays) [11], LSTM with self-check requires centralized data (120 sample delays) [12], and active transfer learning needs manual intervention (5-10 second adaptation times) [13]. In contrast, the proposed method features explicit drift detection with shorter delays, continuous adaptation without manual intervention, data privacy through FL, and superior accuracy (99.65% versus 96.2% for the work in [12]). In summary, the main contributions of this paper are listed as follows.

- A novel approach to TSA under CD is proposed. Unlike existing methods that assume static data distributions, this paper introduces an efficient data stream analytics framework capable of detecting and adapting to CD. This represents one of the first attempts to address CD in TSA explicitly.
- An effective method for TSA that uses a FL environment is proposed. The proposed model aims to preserve data privacy by training the model on local servers.
- An online prediction model based on data streams is proposed. The proposed method uses Adaptive Random Forest (ARF) to generate classification results based on the streams of data.

The rest of the paper is structured as follows. Section II

provides detailed discussions about the problem statement. The proposed architecture and the simulation results are outlined in Section III and Section IV, respectively. Finally, the paper is concluded in Section V.

II. Problem Formulation

This section formulates the TSA problem for PSs under uncertainty, considering the challenges of data streams and CD in modern power networks. PSs are complex networks where electrical stability is a key concern. This is especially true under dynamic conditions and in the presence of uncertainty. Transient stability, a critical aspect of PS analysis, evaluates the system's ability to maintain synchronism when subjected to significant disturbances. This ability is crucial for ensuring operational continuity and reliability. The dynamics of PSs can be encapsulated by stochastic Differential Algebraic Equations (DAEs), which forms the basis for transient stability analysis:

$$\dot{x} = f(x, V, t), \tag{1}$$

$$0 = g(x, V, t), \tag{2}$$

$$x = \{x_i | i = 1, 2, \ldots, n\}, \quad x(t_0) = x_0, \tag{3}$$

$$V = \{V_b | b = 1, 2, \ldots, m\}, \quad V_b = [|V_b|, \angle\theta_b]^T. \tag{4}$$

The state variables, represented by $x \in \mathbb{R}^n$, evolve according to a set of differential equations, and their initial conditions are denoted as x_0. The time variable t spans an interval $[t_0, T]$. The algebraic variables, symbolized by $V \in \mathbb{R}^m$, include the nodal voltages of the system. The functions $f(x)$ and $g(x)$ encapsulate the system's nonlinear DAEs, respectively, with n and m denoting the count of generators and buses. The assessment of transient stability is determined by the maximum phase angle deviation, $\delta_{\max}$, which is derived from the state vector x. The interaction between the state vector x and the algebraic variable vector V is described by a nonlinear system of DAEs.

$$x(t_0 + \Delta t) = x_0 + \int_{t_0}^{t_0+\Delta t} f(x, V, t)\, dt, \tag{5}$$

$$0 = g(x(t_0 + \Delta t), V(t_0 + \Delta t), t). \tag{6}$$

The goal of TSA is to evaluate whether the system can maintain stability, which is primarily determined by monitoring the phase angle difference between generators. The largest phase angle difference $|\delta_{\max}|$ is determined as

$$|\delta_{\max}| = \max\left\{|\delta_i(t) - \delta_j(t)|, \forall i, j \in \{1, ..., n\}, t \in [t_0, T]\right\}, \tag{7}$$

where $|\delta_{\max}|$ symbolizes the utmost phase angle variation between any two generators during the transient period. This stability index provides a binary indication of the system's state, which is crucial for quick decision-making. If this difference exceeds a certain threshold, the system is considered unstable, as it cannot maintain synchronism. $\delta_{\max}$ serves as the primary stability criterion because of its direct correlation with synchronizing power between generators (proportional to $\sin(\delta_i - \delta_j)$). When this angle exceeds $90°$, synchronizing power decreases and instability becomes likely. While alternative criteria exist, such as energy functions, equal area criterion, and frequency metrics, $\delta_{\max}$ offers superior computational efficiency for multi-generator systems. Energy functions require complex modeling unsuitable for real-time applications, equal area criterion becomes impractical beyond two-machine systems, and frequency metrics are better for post-event analysis than prediction. $\delta_{\max}$ thus provides both physical relevance and practical implementation aligned with established Lyapunov stability theory for PSs.

The Transient Stability Index (TSI) is a widely recognized metric for gauging the transient stability of PSs. The TSI is calculated as follows.

$$TSI = 100 \times \frac{360 - \delta_{\max}}{360 + \delta_{\max}}, \tag{8}$$

where $\delta_{\max}$ is the peak rotor angle difference between any two generators throughout dynamic simulations. A TSI value exceeding zero signifies system stability and is denoted by a label of 1, while a negative TSI indicates potential instability, which is marked with a label of -1. The TSI is formulated as

$$y = \begin{cases} 1 & \text{(stable), TSI} > 0 \\ -1 & \text{(unstable), TSI} \leq 0. \end{cases} \tag{9}$$

This stability index provides a binary indication of the system's state, which is crucial for quick decision-making. The transient stability evaluation function $M(\cdot)$, which correlates TSI with the state vector $\zeta = [P_{\mathrm{w}}^u, P_{\mathrm{L}}, Q_{\mathrm{L}}, P_{\mathrm{G}}]^T$, can be articulated as $TSI = M(\zeta)$, where P_{w}^u, P_{L}, Q_{L}, and P_{G} correspond to the stochastic active power of wind generation, the active power of loads, the reactive power of loads, and the active power of generators, respectively. To investigate uncertainty's effects on system stability, Monte Carlo simulation is utilized to sample a multitude of potential outcomes $Y = \{TSI_1, TSI_2, \ldots, TSI_N\}$ based on a probability distribution function associated with the uncertain factors from the sample space $X = \{\xi_1, \xi_2, \ldots, \xi_N\}$. This probabilistic approach allows assessing the robustness of the PS's stability under diverse operating conditions, addressing the inherent uncertainty in renewable generation and fluctuating loads. Building on this probabilistic assessment of system stability, the next section introduces the components of the proposed framework. It details the core learning architecture and its integration with federated processing and concept drift adaptation mechanisms.

III. Proposed architecture

This section explores the ARF model, followed by an examination of FL implementation. The intricacies of CD detection are then discussed. The proposed FL-based ARF (FedARF) model integrates the preceding elements to form a robust and adaptable OTSA system.

A. Adaptive Random Forest

The ARF method is an ML algorithm that combines the principles of both Random Forest (RF) and online learning [15]. For any given instance i in a data stream, the ARF model constructs an ensemble of decision trees $\{T_j\}_{j=1}^N$ [16].

Each tree T_j is trained on a bootstrap sample D_j drawn from the data stream. The bootstrapping process ensures that the trees are exposed to different subsets of the data, promoting diversity within the ensemble. The ARF algorithm evaluates the predictive performance of each tree T_j through a metric $\text{Acc}(T_j)$, which represents the accuracy of tree T_j on a holdout set H_j that is not used during the tree's training. When new data arrives, the ARF algorithm updates the performance metric and determines if a tree should be replaced. If $\text{Acc}(T_j)$ falls below a threshold θ, tree T_j is pruned from the ensemble and replaced with a new tree $T_{j'}$ trained on recent data.

When replacing underperforming trees in the ARF ensemble, diversity is explicitly maintained through several mechanisms. First, each new replacement tree is trained on a bootstrap sample with randomly selected features, following the standard RF approach. The bootstrap sampling ensures different training data distributions, while the random feature selection (typically using $\sqrt{d}$ features, where d is the total feature dimension) prevents new trees from converging to similar decision boundaries. Additionally, a stratified feature sampling approach is employed, where the feature space is partitioned and replacement trees are assigned different feature subsets based on their position in the ensemble. This strategy ensures that even after multiple tree replacements due to concept drift, the ensemble maintains heterogeneity in the data samples and feature spaces used by individual trees. This diversity is crucial for preserving the collective predictive power of the ensemble.

Adaptation to the evolving data is achieved by updating the tree ensemble dynamically. When an instance i is misclassified by a tree T_j, a drift detection method $\text{DriftDet}(i, T_j)$ is invoked, which may trigger the replacement of T_j if the CD is detected. The output of the ARF for a new instance x is given by the majority vote or the average prediction across the ensemble, defined as [15]

$$\text{Decision Tree:} \quad T_j(x;\Theta_j) = \sum_{i=1}^{I} \gamma_{ij}(x;\Theta_j), \tag{10}$$

$$\text{Random Forest:} \quad \text{RF}(x) = \frac{1}{N}\sum_{j=1}^{N} T_j(x;\Theta_j), \tag{11}$$

$$\text{ARF:} \quad \text{ARF}(x) = \frac{1}{N}\sum_{j=1}^{N} T_j(x;\Theta_j), \tag{12}$$

where $T_j(x;\Theta_j)$ is a decision tree indexed by j with parameters Θ_j, and $\gamma_{ij}(x;\Theta_j)$ is the prediction of the i-th leaf node in tree j. The ensemble prediction $\text{ARF}(x)$ is the average prediction of the ARF with N trees at a given time. The parameters Θ_j are optimized to minimize a loss function L across the data stream.

B. Federated Learning

At the onset of the comprehensive training regimen, all or a subset of clients are selected and furnished with the most recent global model parameters [17]. Each client C_k carries out multiple optimization epochs (for instance, using the Adaptive Moment Estimation or Adam) utilizing the amassed local data D_k. It is important to note that in the proposed FL implementation, local client variability can impact global model convergence. The Adam optimizer's adaptive learning rates help mitigate the effects of non-IID (Independent and Identically Distributed) data distributions across clients, which is common in PSs where different operators may experience distinct operational patterns. To further address convergence challenges, a momentum-based aggregation approach is implemented, weighting client updates based on both their data volume and historical contribution patterns. This approach helps prevent model bias toward clients with larger datasets or more frequent updates. Additionally, a synchronization coefficient β is employed to control the influence of local updates on the global model, allowing a balance between fast convergence and stability: $w_{t+1} \leftarrow w_t - \beta \sum_{k=1}^{k} \frac{n_k}{n} w_{t+1}^k$. Through empirical testing on the PS datasets, it was found that $\beta = 0.8$ provides the optimal trade-off between convergence speed and model stability, especially when handling the CD scenarios discussed in Section III.C.

The local model parameters are subsequently adjusted as follows [18].

$$w_{t+1}^k \leftarrow w_t - \eta \bigtriangledown \ell(w_t). \tag{13}$$

Here, η symbolizes the learning rate and $\eta \bigtriangledown \ell(w_t)$ represents the batch gradient. These updates are dispatched to the server where the secure aggregation is conducted. The underlying computation process is formulated as [18]

$$w_{t+1} \leftarrow w_t - \sum_{k=1}^{k} \frac{n_k}{n} w_{t+1}^k, \tag{14}$$

where $n_k = |D_k|$ and $n = |D_1 \cup \ldots \cup D_k|$. The whole process is then initiated once more, repeating the previous steps.

C. Drift Detection Methods

Consider a data stream represented as $S = \{(x_t, y_t)\}_{t=1}^{T}$, where each $x_t \in \mathbb{R}^d$ is a feature vector from a d-dimensional space and y_t is the corresponding label from a set of categories $\{c_1, c_2, \ldots, c_n\}$, with $n > 1$ signifying the count of distinct categories. At each time step, a new data instance (X, Y) is observed and a predictive model is tasked to infer the label of x_t utilizing the prior instances $\{(x_1, y_2), \ldots, (x_{t-1}, y_{t-1})\}$. Subsequently, the true label of x_t is disclosed by the environment. A scenario where $\exists t_d : P_{t_d}(X, Y) \neq P_{t_{d+1}}(X, Y)$ indicates the occurrence of CD within the stream S [19]. Ref. [20] introduced an enhancement to the prevailing definition of CD by stipulating that $P_t(X, Y) = P_{t+1}(X, Y)$ for $t \in \{t_d + 1, t_d + \tau\}$, with $\tau > 2$, signifying that the data distribution sustains consistency across at least two time points and thus, can be identified as CD, which is distinguishable from mere stochastic fluctuations.

Several methods have been proposed to tackle this issue, each with its unique approach and mathematical foundations. Particularly, this paper discusses Adaptive Windowing (ADWIN), Drift Detection Method (DDM), Hellinger Distance Drift detection Method - A-test (HDDM_A), Hellinger Distance Drift detection Method - W-test (HDDM_W),

Kolmogorov-Smirnov Windowing (KSWIN), and Page-Hinkley test (PageHinkley). The following subsections provide a brief description of each data drift method.

1) Adaptive Windowing (ADWIN): ADWIN, which was designed to identify shifts in the distribution's mean of x_t, operates on the presumption that each data point x_t (where $1 \leq t \leq n$) is confined to the range $[0, 1]$, where n is the length of the window. The process begins with ADWIN initializing an empty window ω. Each new x_t is then added to the end of the window in the form $\omega \leftarrow \omega \cdot x_t$. Subsequently, the algorithm calculates the test statistics as [21]

$$U(\omega) \underset{\omega_1 \cdot \omega_2 = \omega}{\text{Max}} \{|\hat{\mu}(\omega_1) - \hat{\mu}(\omega_2)| - \epsilon_{\text{cut}}\}, \tag{15}$$

here, the maximum allowed error is calculated as [21]

$$\epsilon_{\text{cut}} = \sqrt{\frac{\frac{1}{|\omega_1|} + \frac{1}{|\omega_2|}}{2} \log\left(\frac{4n}{\delta}\right)}, \tag{16}$$

with $\hat{\mu}(\omega)$ signifies the mean value of the data within window ω. The algorithm proceeds to discard the earliest element in the window, persisting in this manner until $U(\omega) \leq 0$ is satisfied, at which point it continues with the addition of new data. The ADWIN method uses the Hoeffding bound to decide the length of the window. Let the Φ be the difference between the empirical average and the true average. The Hoeffding bound is computed as [21]

$$P[|\Phi| > \epsilon] \leq 2exp(-2n\epsilon_{\text{cut}}). \tag{17}$$

For prompt detection, it is crucial that the actual value y^t of the instance x^t be readily accessible right after the prediction output before moving to the instance x^{t+1}. Nevertheless, if immediate access to the ground truth is not possible, the effectiveness of the ADWIN in detecting drifts could be significantly diminished.

2) Drift Detection Method (DDM): To detect CD and changes in data distribution, DDM controls the number of errors produced by the learning model during prediction. It compares the statistics of two windows: one with all the data, and a smaller one with the recent data. When the error rate in the small window is higher, a warning level is triggered. If the error increases further, a drift is detected. The DDM method employs the Binomial distribution and standard deviation to calculate error rates and detect drifts. If p_i is the error rate, n is the number of samples, and s_i is the standard deviation calculated as [22]

$$s_i = \sqrt{\frac{p_i(1-p_i)}{i}}. \tag{18}$$

3) Hellinger Distance Drift Detection Methods (HDDM_A and HDDM_W): These are unsupervised CD detection methods that measure the distance between two probability distributions of the current and the reference time windows. The A-test is for nominal attributes, and the W-test is for numeric attributes. The Hellinger distance between two probability distributions P and Q is defined as [23]

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i}^{n} (\sqrt{p_i} - \sqrt{q_i})^2}, \tag{19}$$

where p_i and q_i are the discrete distributions of P and Q. The choice between HDDM_A and HDDM_W is determined by the nature of the input data. HDDM_A is specifically designed for categorical (nominal) attributes, making it suitable for discrete features such as bus statuses or circuit breaker positions. It uses a frequency-based estimation approach to calculate probability distributions. In contrast, HDDM_W employs a windowing scheme with kernel density estimation, optimized for continuous numerical data such as power flows, voltage magnitudes, and phase angles. In the implementation of the proposed methodology, HDDM_A is applied to discrete topology-related features and HDDM_W to continuous measurement variables from PMUs. When both detectors are used in combination, a more robust detection capability is enabled that can capture drift in both the categorical system configuration parameters and the continuous state variables.

4) Kolmogorov-Smirnov Windowing (KSWIN): KSWIN is a CD detection method based on the Kolmogorov-Smirnov test. KSWIN does not maintain a window, rather a statistic based on the maximum distance between the empirical cumulative data distribution function of the reference window and the test window. The Kolmogorov-Smirnov test computes the absolute distance D_i between two empirical cumulative distributions $\{F_1, F_2\}$ as [24]

$$D_i = Max|F_1(x_i) - F_2(x_i)|, \tag{20}$$

where $F_1(x)$ and $F_2(x)$ are the empirical distribution functions of the two samples.

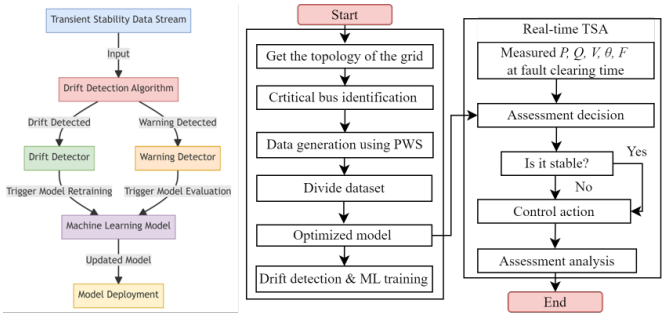
5) Page-Hinkley Test (PageHinkley): This is a sequential analysis technique typically used for monitoring change detection. It allows the detection of changes more quickly by producing an alarm following a change. It has been widely used in medical and industrial applications. The Page-Hinkley test is computed as [25]

$$PH_i = \sum_{i=t}(X_i - \bar{X}) - min \sum_{j=t}(X_j - \bar{X}), \tag{21}$$

where X_i is the i^{th} observation, $\bar{X}$ is the average of all observations, and t is the current time step. A change is detected if $PH_i > \lambda$, where λ is a threshold. In this study, the process of continuous monitoring with CD detection is conducted through drift detectors and warning detectors as shown in Fig. 1 (a). When the algorithm detects a drift or a warning, it triggers either the drift detector or the warning detector, respectively. The drift detector then triggers the retraining of the ML model, while the warning detector triggers the model's evaluation. Following the update, the newly refined model is then deployed, ensuring that the predictions it makes are always based on the most recent learnings.

D. Federated Adaptive Random Forest

The FedARF method is a ML algorithm that combines the principles of both FL and ARF [26]. It is used for classification tasks in TSA analysis, where data is distributed across multiple PS operators. The FedARF algorithm works by creating a local model for each PS operator using the ARF algorithm. The local models are trained on the local data of each operator,



(a) (b)

Fig. 1: (a) Drift detection paradigm, and (b) proposed framework.

and then the models are aggregated to form a global model using FL techniques, as shown in Fig. 1 (b). The global model is then used for the prediction of transient stability across the entire PS. The FedARF algorithm is based on the following equations.

$$\text{Local Model:} f_i(x) = \sum_{j=1}^{N_i} w_{ij} h_j(x;\theta_{ij}) \ , \tag{22}$$

$$\text{Global Model:} F(x) = \sum_{i=1}^{K} w_i f_i(x) \ , \tag{23}$$

$$\text{FL:} w_i^{(t+1)} = w_i^{(t)} - \eta \nabla L_i(w_i^{(t)}, F^{(t)}) \ , \tag{24}$$

where $f_i(x)$ is the local model at operator i, and w_{ij} is the weight of tree j in the ensemble for operator i. The function $h_j(x;\theta_{ij})$ denotes the j^{th} decision tree in the ensemble with parameters θ_{ij}. $F(x)$ represents the global model, and K is the number of PS operators. The term w_i denotes the weight of operator i in the global model. Finally, $L_i(w_i, F)$ is the loss function for operator i with weights w_i and global model F. The learning rate η plays a critical role in the proposed FedARF method. Rather than using a fixed value, an adaptive learning rate strategy is employed that responds to detected concept changes. Initially, η is set to 0.01 based on empirical testing across the generated PS datasets. When CD is detected by any client, the server adjusts η according to $\eta_{t+1} = \eta_t \cdot (1 + \alpha \cdot \text{drift\_magnitude})$. Here, the drift_magnitude represents the severity of detected drift (measured as the statistical distance between old and new distributions), and α is a sensitivity parameter set to 0.2. This approach allows faster adaptation during significant distribution shifts while maintaining stability during minor fluctuations. The proposed algorithm is shown in Algorithm 1.

A systematic block diagram of the proposed FedARF-based TSA is presented in Fig. 2. The framework follows a structured process beginning with local client initialization and data preprocessing of PMU measurements into time series buffers. The core innovation lies in the enhanced drift detection ensemble that triggers adaptive tree replacement logic when concept drift is detected, maintaining model diversity through stratified feature sampling. Local ARF model updates

Algorithm 1 Proposed FedARF Method-based TSA

Input: A distributed data stream S across multiple clients $\{C_k\}$, each with its dataset D_k
Initialize global model parameters Θ
Set the number of trees N for each local ARF model
for each client C_k in parallel **do**
    Initialize local ARF model with N trees, parameters θ_{ij}
end for
repeat
    for each client C_k in parallel **do**
        Bootstrap local dataset D_{kj} for each tree T_j in ARF
        Train each tree T_j on D_{kj}
        Evaluate local ARF on holdout set H_k
        Detect concept drift using DriftDet on H_k
        if drift is detected **then**
            Prune trees with accuracy below threshold θ
            Create new replacement trees with bootstrap samples from recent data
            Ensure diversity by randomizing feature subsets ($\sqrt{d}$ features)
            Apply stratified feature sampling for replacement trees
        end if
        Calculate local updates Δw_k using gradients $\nabla L_k(\theta_k)$
    end for
    Aggregate local updates Δw_k on the server
    Update global model parameters Θ using weighted aggregation
    Broadcast updated Θ to all clients C_k
until convergence or maximum iterations reached
Output: Global ARF model optimized for the distributed data stream S

incorporate drift-aware learning followed by secure parameter extraction and transmission to the central aggregator. The federated aggregation process combines FedAvg with adaptive learning rate adjustment ($\eta = \eta_0 \times (1 + \alpha \times \text{drift magnitude})$) and model diversity strategies. Update synchronization ensures all clients receive the global model while maintaining privacy through secure protocols. The iterative process continues until convergence, ultimately providing real-time stable/unstable classifications that can be used by system operators for preventive control actions. To evaluate the practical performance of the proposed framework, the next section presents simulation experiments conducted on multiple standard PS test cases.

IV. Simulation Results

This section illustrates the effectiveness of the proposed method through a variety of simulation results. Three bus systems with different scales are discussed: IEEE 39-Bus test system, 68-Bus system, and ACTIVSg25K-Bus test system. All the simulations have been implemented via Google Colab Pro Plus High-RAM and Background Execution options Enabled. Pre-installed packages were used to reduce potential

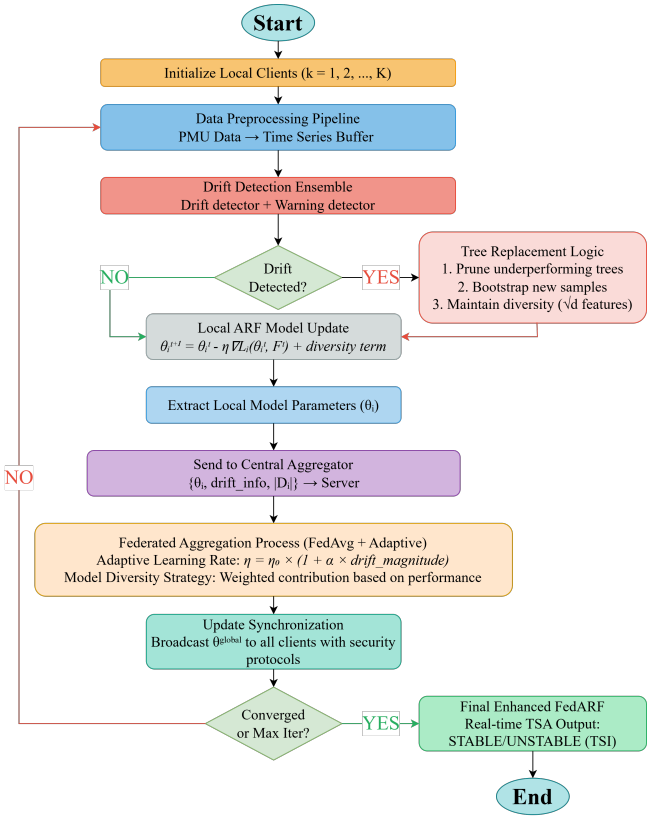


Fig. 2: Systematic FedARF diagram for OTSA.

errors from version incompatibilities. The River library is installed for coding the online learning models [27]. All the simulations were performed using PowerWorld Simulator (Version 23). The computational simulations were conducted on a Lenovo laptop with 16 GB RAM. The proposed method and comparative models are implemented using Python with Scikit-Learn framework. All simulations are conducted using 10 repeated trials for each IEEE bus system, which are used to calculate the average resulting metrics.

A. Data Pre-Processing

To generate the training data, a three-phase short-circuit fault is induced on a transmission line and subsequently cleared by removing the faulted line. The fault location is randomly selected at either end of the line. The fault clearing time is also chosen randomly within the range of 0.1 to 10 seconds, using an integration step of 0.01 seconds. The stability of the system is assessed upon the completion of the simulation. When the fault is cleared, the values of P and Q are recorded for all the lines, as well as V and θ for the buses (with the angle at the generator of the critical bus serving as a reference). These measurements are the initial input features. Then, data points are created for each bus configuration. The distribution of these samples is presented in Table I. To ensure fairness during model training and address class imbalance, the dataset was generated to maintain an approximately equal number of stable and unstable scenarios for each system configuration.

TABLE I: Description of the knowledge base.

Bus system	39-Bus	68-Bus	ACTIVSg25K
Stable	11227	34411	12021
Unstable	11993	34389	11983
Total	23220	68800	24004

The t-distributed Stochastic Neighbor Embedding (t-SNE) maps the high-dimensional space into a 2-dimensional space. A two-dimensional projection using the t-SNE algorithm for different IEEE bus systems is plotted in Fig. 3. According to

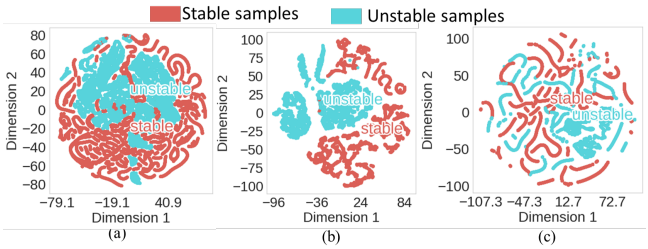


Fig. 3: The 2-D visualization of sampling strategy behaviors with t-SNE. a) 39-Bus system, b) 68-Bus system, and c) ACTIVSg25K-Bus system.

Fig. 3, all samples are interspersed in the initial feature space. However, in Fig. 3 (a), it can be seen that the samples gradually separate into two distinct clusters. This separation enhances the intuitive identification of unstable scenarios within the representation space, offering an advantage over the initial feature space. As illustrated in Fig. 3 (b), the feature space exhibits significant overlap between different regions, indicating challenges in distinguishing between sample categories. Fig. 3 (c) demonstrates that a large portion of the data points are widely dispersed and do not closely approach the dividing lines between stable and unstable classifications. This distribution pattern illustrates that the model's learning process is challenging to avoid any overfitting issues with topological changes. Fig. 4 (a,b) demonstrates the rotor angle of generators for less severe and more severe stable scenarios when the New England 39-Bus system has suffered a short-circuit fault, respectively. While Fig. 4 (c), illustrates an unstable scenario when the New England 39-Bus system has suffered a short-circuit fault on the critical bus.

For comparative study, the online artificial neural network (OANN), Extremely Fast Decision Tree classifier (EFDT), Passive-Aggressive classifier model (PAC), incremental adaptive boosting (IAda) classifier are introduced as competitive models to the ARF algorithm [28], [29]. In addition, three state-of-the-art online techniques were included in the comparative study, including Neural Network with Dynamically Evolved Capacity (NADINE) [30], Selective Ensemble-based Online Adaptive deep neural networks (SEOA) [31], and Online Transformer Network (OTN). All the deployed models are trained in an FL environment. Furthermore, the hyperparameters of the implemented models are tuned using Optuna [32]. Considering the Optuna results, the ARF model employs the DDM for both drift and warning detection. The OANN from the linear model module operates with a hinge

TABLE II: Key hyperparameters for each online model.

Model	Key Hyperparameters with Optuna Optimization
ARF	$n_{\text{models}} = 10$, max_features = SQRT, $\lambda\_value = 6$
OANN	hidden_layers = (50,), solver = SGD(0.01), learning_rate_init = 0.01
EFDT	grace_period = 200, split_confidence = 1×10^{-7}, tie_threshold = 0.05
PAC	C = 1.0, mode = 1, learn_intercept = True
IAda	base_estimator = Hoeffding Tree Classifier, $n_{\text{models}} = 10$, seed = 42
OTN	hidden_dim = 64; num_heads = 4; window_size = 10
SEOA	hidden_dims = [32, 64, 128]; learning_rate = 0.01; fluctuation_window = 10
NADINE	initial_hidden = 16; learning_rate = 0.01; memory_buffer_size = 50

loss function. The EFDT is set with a grace period of 40, uses information gain as the split criterion, and predicts using the naive bayes adaptive approach and reevaluates at a minimum of 40 samples. The PAC model is parameterized with a regularization parameter C set to 0.001. The IAda classifier uses a base model of Hoeffding tree classifier with a Gini impurity split criterion, a change detection threshold (delta=1e-5), and a grace period of 20. The optimization results for the TSA algorithm are found in Table II.

B. Evaluation measures

The efficacy of the proposed method is assessed using Accuracy (ACC), Precision (Prec), Recall (R), and F1-score (F1). Their mathematical definitions are given below.

$$ACC = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}, R = \frac{T_P}{T_P + F_N}, \tag{25}$$

$$F1 = 2 \times \frac{Prec \times R}{Prec + R}, Prec = \frac{T_P}{T_P + F_P}, \tag{26}$$

where T_P, T_N, F_N, and F_P denote True Positive, True Negative, False Negative, and False Positive, respectively.

C. IEEE 39-Bus System

In this subsection, the experimental results are derived from simulations conducted on the widely recognized New England 39-Bus system, frequently showcased in TSA studies [1]. The IEEE 39-Bus system comprises 10 power generators, 19 load points, 12 transformers, and 34 power transmission lines. The IEEE 39-Bus system is subjected to testing to corroborate the efficacy of the proposed methodology. Three-phase faults are applied at multiple line locations. The fault durations of 5 and 9 cycles are selected to represent distinct severity levels commonly encountered in practical PSs. The 5-cycle duration (83.3 ms at 60 Hz) simulates typical fault-clearing times with modern protection systems. The 9-cycle duration (150 ms) represents delayed clearing scenarios due to backup protection operation. This range covers the Critical Clearing Time (CCT) threshold for many practical system configurations. The analysis revealed that drift detection performance is directly influenced by fault duration, with longer durations producing more pronounced shifts in the data distribution. This observation aligns with PS theory, where longer fault durations push the system closer to its stability limits, creating more defined separation between stable and unstable cases, as visualized in the t-SNE projections in Fig. 3. Several effective data stream analytics methods, including CD and warning detectors are implemented and verified. Fig. 5 (a) illustrates an exhaustive pairwise comparison of several drift detection and warning detection algorithms on the IEEE 39-Bus system.

According to Fig. 5 (a), the values range from about 96.35% to 98.95%, indicating that all combinations perform quite well. The highest value in the heatmap figure (98.95%) is achieved when KSWIN is used as both the drift and warning detector. This suggests that KSWIN's underlying statistical approach might be particularly well-suited to the IEEE 39-Bus system among the algorithms tested. This advantage comes from leveraging the Kolmogorov-Smirnov test to sensitively detect distributional changes in streaming data. The lowest value in the heatmap figure (96.35%) comes from pairing HDDM_W as the drift detector and KSWIN as the warning detector. This may indicate some degree of incompatibility or suboptimal performance between these two specific detectors. From an operational perspective, the difference between the best and worst combinations (a spread of approximately 2.6%) could have meaningful implications for real-time power system stability assessment, especially under high-frequency PMU data streams. Selecting suboptimal detector combinations could

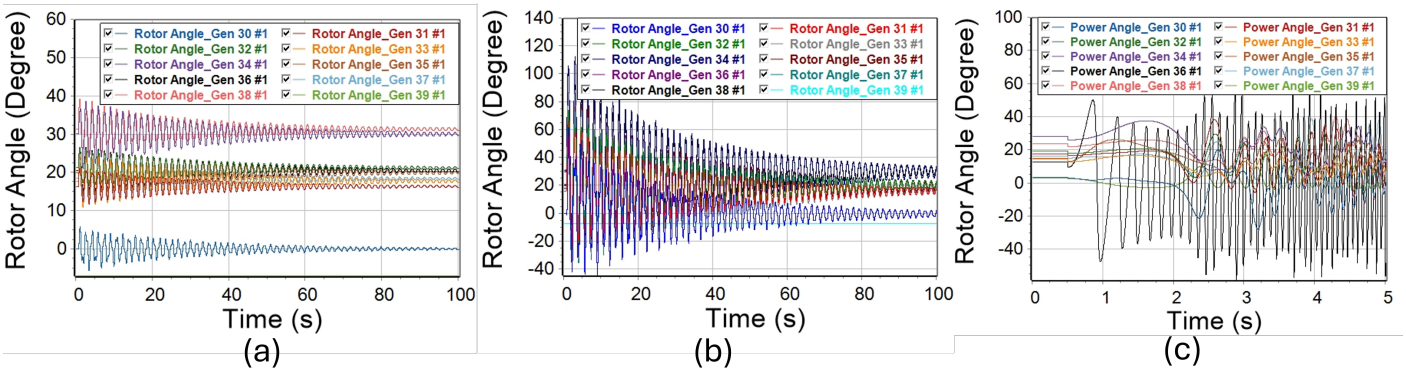
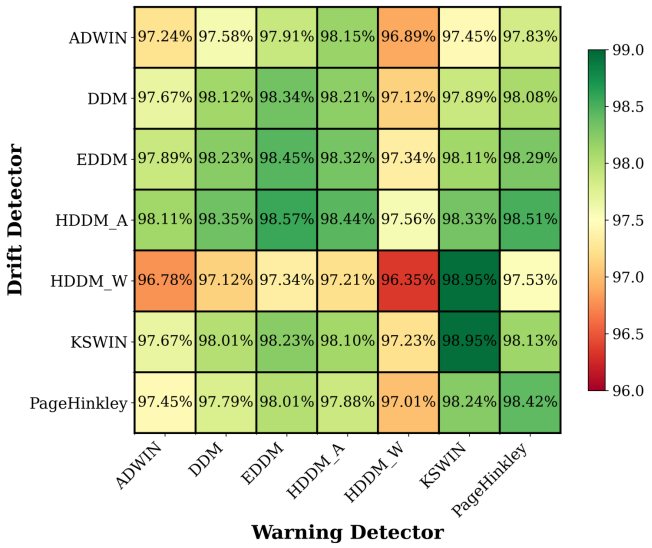
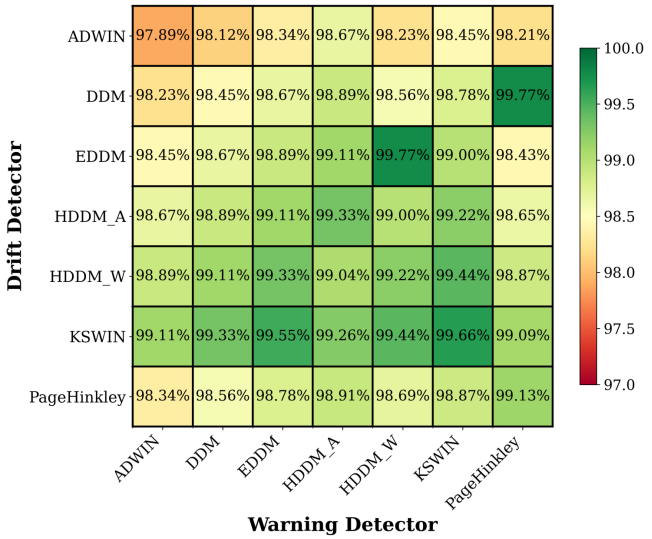


Fig. 4: a) Transient stability observed under a minor fault, b) Transient stability maintained despite a severe fault, and c) Transient instability scenario.

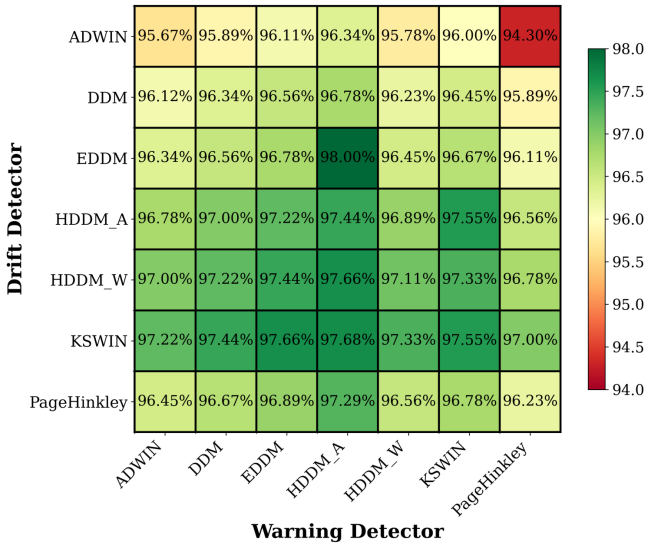
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3583978



(a) Accuracy with the IEEE 39-bus system.



(b) Accuracy with the IEEE 68-bus system.

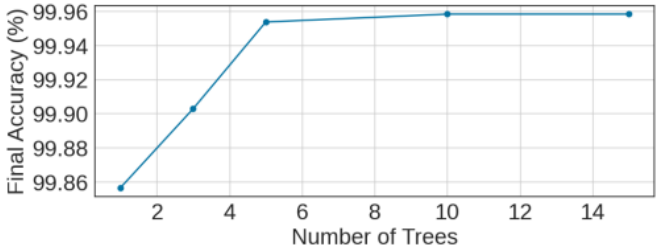


(c) Accuracy with the IEEE 25k-bus system.

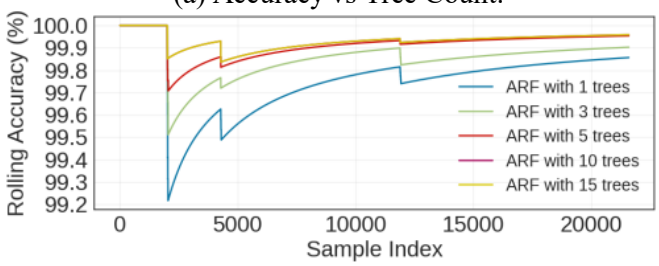
Fig. 5: Data stream classification accuracy with combinations of drift and warning detectors for the (a) IEEE 39-Bus (b) IEEE 68-Bus (c) IEEE 25k-Bus systems.

result in hundreds of additional misclassifications per day in large-scale deployments, underscoring the practical importance of these findings.

The relationship between the number of trees (N) and the generalization ability of ARF-ADWIN-based TSA is studied and illustrated in Fig. 6. Fig. 6 (a) shows that accuracy increases significantly from 1 to 5 trees (99.86% to 99.95%), with minimal gains beyond this point, demonstrating clear diminishing returns. This pattern aligns with the theoretical principle of ensemble learning, where the marginal contribution of each additional tree decreases as the ensemble size grows [33]. Fig. 6 (b) reveals how different tree counts respond to CD events (visible as accuracy drops near samples 2000 and 4000), with higher tree counts exhibiting smaller accuracy drops and faster recovery. The analysis confirms that 5 trees provide an optimal balance between accuracy and computational cost. Models with fewer trees (1-3) show greater vulnerability to drift, while larger ensembles (10-15 trees) offer negligible performance improvements despite significantly increased computational requirements.



(a) Accuracy vs Tree Count.



(b) ARF-ADWIN Results with Different Tree Counts.

Fig. 6: Impact of tree count on ARF-ADWIN performance for online TSA, with (a) representing the relationship between accuracy and tree count (b) illustrating how the ARF-ADWIN model with different tree counts responds to CD events.

A comprehensive performance evaluation for the proposed FedARF algorithm when facing different types of CD scenarios in the IEEE 39-Bus system is summarized in Table III. The results demonstrate the robustness of the proposed approach across varying drift conditions. For instance, the impressive performance was particularly achieved under recurring drift patterns (98.47% mean accuracy with minimal variance of 0.12). This indicates both high precision and stability of the FedARF model in scenarios where system states repeat over time. While gradual drift scenarios show satisfactory accuracy (97.87%), the FedARF requires slightly longer processing time (1.03±0.41s). This increase in runtime can be attributed to the

FedARF's need to continuously adapt to slowly evolving data distributions, which may involve more frequent model updates or increased communication overhead in the federated setting.

Notably, the F_P rates represent the percentage of stable operating conditions incorrectly classified as unstable. This metric is especially critical in PS operations where false alarms could lead to unnecessary control actions or load shedding. The proposed model maintains acceptable F_P rates across all drift types, with the lowest rate observed in recurring drift scenarios ($3.01\pm0.93\%$), indicating excellent discrimination ability even when system conditions return to previous states. The Delay Index (DI) is introduced as

TABLE III: Descriptive statistics for TSA-based FedARF under different types of CDs using IEEE 39-Bus system.

Statistical Metrics	**Drift type**			
	Sudden	Gradual	Incremental	Recurring
		Accuracy		
Mean	97.60	97.87	96.60	98.47
95% CI Lower	95.32	95.80	95.74	98.18
95% CI Upper	99.88	99.94	97.46	98.75
Std	0.92	0.83	0.35	0.12
Min	96.80	97.20	96.20	98.40
25%	97.10	97.40	96.50	98.40
50%	97.40	97.60	96.80	98.40
75%	98.00	98.20	96.80	98.50
Max	98.60	98.80	96.80	98.60
Runtime (s)	0.86 ± 0.21	1.03 ± 0.41	0.56 ± 0.42	0.64 ± 0.07
False Positive (%)	3.18 ± 3.04	3.45 ± 3.37	4.34 ± 1.81	3.01 ± 0.93
Delay Index	0.30	0.33	0.20	0.89

the ratio between mean detection delay and drift width (DI =mean_delay$_d$/drift_width$_d$ [34]) to provide a normalized measure of detection responsiveness. Lower values indicate faster adaptation to changing conditions relative to the drift transition period. Table III reveals a DI of 0.20 in incremental drift. This means that the FedARF algorithm detects changes within just 20% of the time it takes for the drift to fully manifest. The results show a slightly higher value for recurring drift of 0.89 due to the complexity of detecting multiple transition points. This higher DI in recurring drift may reflect the challenge of distinguishing between genuine new drifts and returns to previously seen states. The excellent responsiveness for sudden drift (0.30) and gradual drift (0.33) confirms that the proposed TSA-based FedARF framework effectively adapts to time-critical changing PS conditions.

D. IEEE 68-Bus System

To further verify the effectiveness of the proposed TSA method, the 16-machine, 68-Bus New England test system is utilized. This system consists of 16 machines, 86 transmission lines, and 5 areas. Fig. 5 (b) shows the performance of different combinations of drift detectors and warning detectors on the IEEE 68-Bus system. For instance, the combination of {EDDM and HDDM_W} and {DDM and PageHinkeley} as both the drift detector and warning detector achieves an accuracy of 99.77%. On the other hand, the combination of HDDM_W as the drift detector and HDDM_A as the warning detector achieves the highest accuracy in the heatmap figure, at 98.04%. The Inference Latency (IL) is the amount of time it takes for a model to process input and return an output. This metric is critical for real-time grid operations, where decisions must be made within milliseconds to prevent cascading failures. IL can be expressed mathematically in various ways. One common method is to measure the time taken for a single input to be processed (from the moment it is fed into the model until an output is produced). This can be written as

$$L = T_{\text{out}} - T_{\text{in}}, \tag{27}$$

where L, T_{in} and T_{out} represent the IL, the time at which an input data point enters the model and the time at which the output is produced by the model, respectively. For the TSA application, the average IL is measured over multiple data points to get a more accurate understanding of the model's performance. This can be defined as

$$L_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n}(T_{\text{out}_i} - T_{\text{in}_i}), \tag{28}$$

where L_{avg} and n represent the average IL as the total number of data points. T_{out_i} and T_{in_i} denote the times at which the output is produced and the input is received, respectively, for the i-th data point.

To evaluate the effectiveness of the proposed model, a comparison is made against five established FL approaches: Federated Averaging (FedAvg) [35], which randomly samples client UAVs each round; Federated optimization (FedProx) [36], which adds a proximal term to constrain local updates toward the global model; Federated Matching (FedMatch) [37], which balances inter-client consistency with disjoint learning to capture both shared and unique features; Polaris [38], which employs asynchronous updates prioritized by communication reliability to minimize latency; and Automatic Layer Freezing (ALF) [39], which freezes layers whose stability index falls below 0.13 to reduce communication overhead.

An end-to-end latency, pure computation time, and per-round communication volume comparison is conducted for six FL strategies under identical experimental conditions, as reported in Table IV. The proposed approach achieves the lowest latency of 3.61 ms and a low communication cost of 0.97 MB, demonstrating a substantial improvement over all existing baselines. This dramatic reduction in latency is particularly significant for time-sensitive power and industrial Internet of Things (IoT) systems. Among the published methods, FedAvg offers the next best trade-off with a low latency of 21.97 ms and a moderate bandwidth of 3.54 MB. FedProx incurs a latency of 8.44 ms and a lowest runtime of 506.7 s. In contrast, asynchronous methods such as FedMatch and Polaris exhibit over 20 ms latency, and ALF, despite freezing stable layers, requires nearly 28 ms and 2.05 MB per round. FedARF's superiority stems from its adaptive, resource-efficient design that enables real-time model updates without costly gradient computations. Thus, the proposed approach results in dramatically reduced latency and minimal communication overhead.

TABLE IV: Comparison of FL-based OTSA strategies.

FL Strategy	Latency (ms)	Runtime (s)	Com Cost (MB)
FedMatch [37]	22.78 ± 0.24	1366.69 ± 14.27	3.54
FedProx [36]	8.44 ± 0.02	506.17 ± 1.16	2.05
FedAvg [35]	21.97 ± 0.09	1318.26 ± 5.14	3.54
Polaris [38]	21.79 ± 0.16	$1\,307.12 \pm 9.48$	3.54
ALF [39]	28.38 ± 0.02	450.03 ± 3.45	2.05
Ours	3.61 ± 0.01	742.93 ± 6.51	0.97

TABLE V: Performance comparison for the IEEE 68-Bus system.

Model	Accuracy	Precision	Recall	F1-Score	Time (s)
ARF	99.65%	99.59%	99.89%	99.74%	3.87
OANN	94.98%	91.77%	98.84%	95.17%	0.88
EFDT	96.30%	93.77%	99.28%	96.45%	1.97
PAC	84.24%	76.18%	99.68%	86.36%	1.22
IAda	97.26%	95.61%	99.16%	97.35%	13.72
OTN	67.2%	67.2%	100%	80.38%	3.08
SEOA	85.4%	85.6%	94.03%	89.62%	3.05
NADINE	81.8%	81.01%	95.24%	87.55%	0.59

Table V illustrates the performance of the competitive models on the 68-Bus system in terms of score errors and computational testing time (s). Looking at Table V, the ARF model exhibits the best overall performance, achieving the highest accuracy of 99.65% and an F1-Score of 99.74% with a moderate runtime of 3.87 s. The IAda classifier follows with strong accuracy of 97.26% and an F1-Score of 97.35%, albeit at a higher computational cost of 13.72 s. Among the neural-based methods, OANN achieves 94.98% accuracy and a 95.17% F1-Score in only 0.88 s; SEOA records 85.4% accuracy, 94.03% recall, and an 89.62% F1-Score in 3.05 s; NADINE attains 81.8% accuracy, 95.24% recall, and an 87.55% F1-Score with the fastest testing time of 0.59 s; and OTN, while showing the lowest accuracy of 67.2%, reaches perfect recall of 100% as a result of a strong bias toward predicting the positive class.

E. ACTIVSgs 25K-Bus system: Scalability Study

This section aims to address the challenges discussed in the study and assess the scalability and performance of the proposed approach. The system under consideration is a large-scale synthetic power grid, consisting of 25,000 buses and featuring significant photovoltaic (PV) penetration. The grid is designed to represent the geographical area of the Northeast and Mid-Atlantic regions in the United States. It is a highly detailed model created using geographic and statistical data for the purpose of planning and stability assessments [40]. The system includes 227 generators (116 modeled as PV systems) [41]. The PV energy contributes around 10 percent to the overall energy mix. This is achieved by regularly updating the models with new data to ensure their accuracy and relevance. The performance of different combinations of drift detectors and warning detectors on the IEEE 25k-Bus system is illustrated in Fig. 5 (c). As seen in the heatmap figure, the highest accuracy of 98.00% is achieved by the combination of EDDM as the drift detector and HDDM_A as the warning detector. Other strong combinations include HDDM_A and KSWIN (97.55%), PageHinkley and HDDM_A (97.29%), and KSWIN and HDDM_A (97.68%). On the other hand, the lowest performance (94.30%) is observed when ADWIN is paired with PageHinkley, demonstrating a potential incompatibility between these two detectors for large-scale PSs.

The rolling accuracy of ARF when paired with seven different drift detection algorithms across 20,000 test samples is presented in Fig. 7. The results demonstrate that ARF+HDDM_W achieves superior performance with the highest final accuracy (99.98%), closely followed by ARF+HDDM_A (99.76%). All configurations show a characteristic pattern of rapid early convergence followed by a temporary drop in accuracy around sample 4,000, indicating the presence of CD at this point. After this disturbance, the models exhibit varying recovery rates, with HDDM_W demonstrating the most robust recovery.

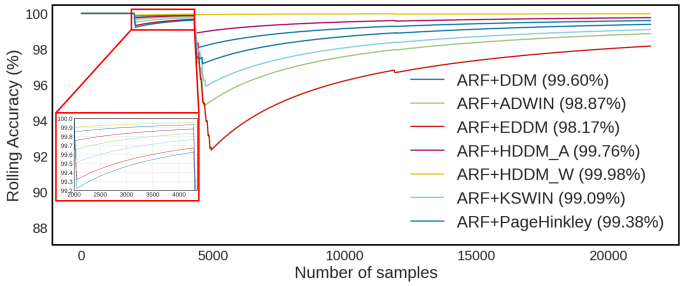
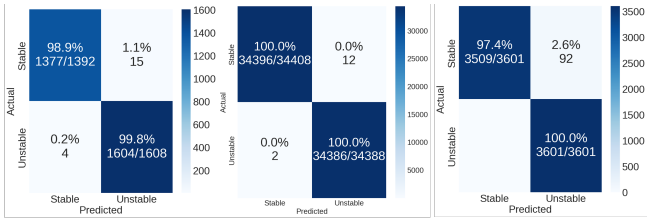


Fig. 7: Performance comparison of ARF method with various concept drift detection methods on the IEEE 25k-Bus system.

The study computes three confusion matrices for the FedARF model applied to three different IEEE test systems as shown in Fig. 8. These results demonstrate that the model performs extremely well across all three systems, with particularly impressive performance on the 68-Bus system. Specifically, the IEEE 68-Bus system achieves near-perfect classification with only 12 misclassifications out of nearly 69,000 samples. The 39-Bus system shows excellent performance with 98.9% accuracy for stable cases and 99.8% for unstable cases, while the significantly larger 25k-Bus system maintains 97.4% accuracy for stable cases.

Notably, the model maintains this high performance even as system complexity increases by several orders of magnitude (from 39 to 25,000 buses). For instance, the 25k-Bus system shows the highest F_P rate at 2.6%. This indicates that as system complexity increases, the FedARF model becomes slightly more prone to false alarms. Nevertheless, the F_P rates, though slightly higher than F_N rates, remain well within acceptable limits across all systems. The consistently low F_N rates across all test systems (significantly below the industry-acceptable threshold of 1%) confirm the model's reliability and robust generalization capabilities while handling different system topologies, operational conditions, and stability ratios.

A comprehensive comparison of IL performance is presented in Fig. 9 for various online learning models across three PS scales. ARF-based models (ARF-DDM, ARF-ADWIN, ARF-HDDM_W) maintain consistently low latency (0.15-0.25 ms for small systems, 6-8 ms for 25k-Bus) with predictable scaling across all PS sizes. The consistent performance of



(a) IEEE 39-Bus (b) IEEE 68-Bus (c) IEEE 25k-Bus

Fig. 8: Confusion matrix for the proposed ARF model for the IEEE (a) 39-Bus, (b) 68-Bus, and (c) 25k-Bus systems.

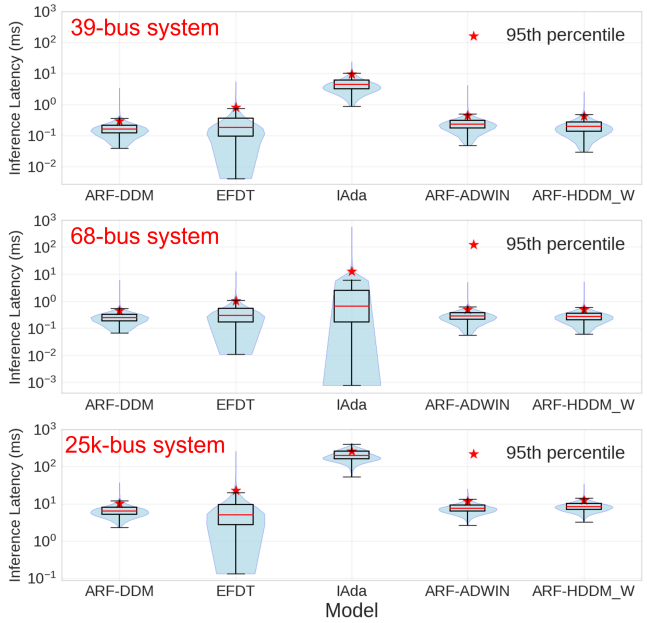


Fig. 9: Inference latency comparison of online learning models across power system scales.

ARF variants is attributed to their ensemble-based architecture, which maintains fixed-size tree structures and leverages efficient incremental learning algorithms. In contrast, IAda exhibits poor scaling properties, with latency increasing dramatically from 4 ms to 200 ms as system size grows. The exponential growth in IAda's latency can be attributed to its instance-based learning approach, which requires maintaining and processing increasingly larger sets of training instances as system complexity grows. While EFDT achieves median latencies of 0.35 ms, 0.42 ms, and 7.5 ms across the three system scales (comparable to ARF-DDM), its 95th percentile latencies are significantly higher-reaching 0.85 ms, 1.2 ms, and 15.8 ms, respectively. This inconsistency is particularly problematic for time-critical applications where worst-case performance guarantees are essential. The ARF-DDM model emerges as the optimal choice, with median latencies of 0.18 ms, 0.25 ms, and 6.2 ms for the 39-Bus, 68-Bus, and 25k-Bus systems, respectively. This ultra-fast inference enables near-instant corrective actions after a disturbance, offering a scalable and significant advantage for real PS security and reliability over all other methods.

V. Conclusion

The growing prevalence of renewable energy sources introduces new complexities to the planning and control algorithms of the Power System (PS). This paper proposed a pioneering approach for Transient Stability Assessment (TSA) in distributed PSs, while effectively tackling Concept Drift (CD). Furthermore, a novel Federated Adaptive Random Forest (FedARF) method was proposed for handling data streams, ensuring data privacy, and maintaining the operational stability of the PS. The proposed model was rigorously tested on the New England 39-Bus system, IEEE 68-Bus system, and ACTIVIgs 25k-Bus system, with its performance compared to that of other cutting-edge and state-of-the-art learning algorithms. Notably, the proposed model demonstrated an excellent performance with 99% accuracy, highlighting its potential for practical applications in dynamic PS environments. This FedARF model enhanced the real-time adaptability and scalability of the prediction framework, which is crucial for dynamic environments, without centralizing sensitive data. In future work, the use of big data platforms is intended to enhance the model's effectiveness for TSA from increased integration of power electronics-dominated grids.

References

[1] N. Hatziargyriou, J. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal *et al.*, "Definition and classification of power system stability revisited & extended," *IEEE Trans. Power Syst.*, 2020.

[2] M. S. Massaoudi, H. Abu-Rub, and A. Ghrayeb, "Navigating the landscape of deep reinforcement learning for power system stability control: A review," *IEEE access*, vol. 11, pp. 134 298–134 317, 2023.

[3] Z. Liu, Z. Ding, X. Huang, and P. Zhang, "An online power system transient stability assessment method based on graph neural network and central moment discrepancy," *Frontiers in Energy Research*, vol. 11, p. 1082534, 2023.

[4] S. M. Blair, M. H. Syed, A. J. Roscoe, G. M. Burt, and J.-P. Braun, "Measurement and analysis of pmu reporting latency for smart grid protection and control applications," *IEEE Access*, vol. 7, pp. 48 689–48 698, 2019.

[5] M. Massaoudi, T. Zamzam, M. E. Eddin, A. Gharayeb, H. Abu-Rub, and S. S. Refaat, "Fast transient stability assessment of power systems using optimized temporal convolutional networks," *IEEE Open Journal of Industry Applications*, 2024.

[6] L. Zhu, D. J. Hill, and C. Lu, "Hierarchical deep learning machine for power system online transient stability prediction," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2399–2411, 2019.

[7] C. Ren, H. Yu, R. Yan, Q. Li, Y. Xu, D. Niyato, and Z. Y. Dong, "Secfedsa: A secure differential privacy-based federated learning approach for smart cyber-physical grid stability assessment," *IEEE Internet of Things Journal*, 2023.

[8] Y. W. Liyanage, D.-S. Zois, and C. Chelmis, "Dynamic instance-wise joint feature selection and classification," *IEEE Trans. Artificial Intelligence*, vol. 2, no. 2, pp. 169–184, 2021.

[9] J. Lu, A. Liu, Y. Song, and G. Zhang, "Data-driven decision support under concept drift in streamed big data," *Complex & intelligent systems*, vol. 6, no. 1, pp. 157–163, 2020.

[10] A. Khan, M. Hosseinzadehtaher, M. B. Shadmand, S. Bayhan, and H. Abu-Rub, "On the stability of the power electronics-dominated grid: A new energy paradigm," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 65–78, 2020.

[11] H. Cui, Q. Wang, Y. Ye, Y. Tang, and Z. Lin, "A combinational transfer learning framework for online transient stability prediction," *Sustainable Energy, Grids and Networks*, vol. 30, p. 100674, 2022.

[12] G. Wang, J. Guo, S. Ma, X. Zhang, Q. Guo, S. Fan, and H. Xu, "Data-driven transient stability assessment with sparse pmu sampling and online self-check function," *CSEE Journal of Power and Energy Systems*, 2022.

[13] B. Li and J. Wu, "Adaptive assessment of power system transient stability based on active transfer learning with deep belief network," *IEEE Trans. Automation Science and Engineering*, 2022.

[14] C. Ren and Y. Xu, "A universal defense strategy for data-driven power system stability assessment models under adversarial examples," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7568–7576, 2022.

[15] Q. Wu, H. Wang, X. Yan, and X. Liu, "Mapreduce-based adaptive random forest algorithm for multi-label classification," *Neural Computing and Applications*, vol. 31, pp. 8239–8252, 2019.

[16] H. Ebrahimy, H. Aghighi, M. Azadbakht, M. Amani, S. Mahdavi, and A. A. Matkan, "Downscaling modis land surface temperature product using an adaptive random forest regression method and google earth engine for a 19-years spatiotemporal trend analysis over iran," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2103–2112, 2021.

[17] X. Zhang, F. Fang, and J. Wang, "Probabilistic solar irradiation forecasting based on variational bayesian inference with secure federated learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7849–7859, 2020.

[18] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[19] P. M. Gonçalves Jr, S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira, "A comparative study on concept drift detectors," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8144–8156, 2014.

[20] Y. Song, J. Lu, A. Liu, H. Lu, and G. Zhang, "A segment-based drift adaptation method for data streams," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 9, pp. 4876–4889, 2021.

[21] R. Kaneko, K. Miyaguchi, and K. Yamanishi, "Detecting changes in streaming data with information-theoretic windowing," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 646–655.

[22] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in *Fourth international workshop on knowledge discovery from data streams*, vol. 6. Citeseer, 2006, pp. 77–86.

[23] L. Piano, F. Garcea, V. Gatteschi, F. Lamberti, and L. Morra, "Detecting drift in deep learning: A methodology primer," *IT Professional*, vol. 24, no. 5, pp. 53–60, 2022.

[24] K. Jafseer, S. Shailesh, and A. Sreekumar, "Modeling concept drift detection as machine learning model using overlapping window and kolmogorov–smirnov test," in *Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021*. Springer, 2023, pp. 113–129.

[25] F. H. Nakagawa, S. B. Junior, and B. B. Zarpelão, "Attack detection in smart home iot networks using clustream and page-hinkley test," in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2021, pp. 1–6.

[26] Y. Ouyang and H. Wang, "Adaptive denoising combined model with sdae for transient stability assessment," *Electric Power Systems Research*, vol. 214, p. 108948, 2023.

[27] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: machine learning for streaming data in python," 2021.

[28] H. M. Gomes, J. Read, and A. Bifet, "Streaming random patches for evolving data stream classification," in *2019 IEEE international conference on data mining (ICDM)*. IEEE, 2019, pp. 240–249.

[29] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1953–1962.

[30] M. Pratama, C. Za'in, A. Ashfahani, Y. S. Ong, and W. Ding, "Automatic construction of multi-layer perceptron network from streaming examples," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1171–1180.

[31] Q. Xiang, L. Zi, X. Cong, and Y. Wang, "Concept drift adaptation methods under the deep learning framework: A literature review," *Applied Sciences*, vol. 13, no. 11, p. 6515, 2023.

[32] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[33] P. Latinne, O. Debeir, and C. Decaestecker, "Limiting the number of trees in random forests," in *Multiple Classifier Systems: Second International Workshop, MCS 2001 Cambridge, UK, July 2–4, 2001 Proceedings 2*. Springer, 2001, pp. 178–187.

[34] A. Liu, J. Lu, Y. Song, J. Xuan, and G. Zhang, "Concept drift detection delay index," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4585–4597, 2022.

[35] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[36] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[37] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," *arXiv preprint arXiv:2006.12097*, 2020.

[38] Y. Kang and B. Li, "Polaris: Accelerating asynchronous federated learning with client selection," *IEEE Transactions on Cloud Computing*, 2024.

[39] E. Malan, V. Peluso, A. Calimera, E. Macii, and P. Montuschi, "Automatic layer freezing for communication efficiency in cross-device federated learning," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6072–6083, 2023.

[40] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, 2016.

[41] T. Xu, A. B. Birchfield, and T. J. Overbye, "Modeling, tuning, and validating system dynamics in synthetic electric grids," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6501–6509, 2018.

Mohamed Massaoudi (Member, IEEE) received the M.Eng. degree in Energy Engineering from the National Engineering School of Monastir (ENIM), University of Monastir, Tunisia, in 2018. He earned the Ph.D. degree in Electronics Engineering from the National Institute of Applied Sciences and Technology (INSAT), University of Carthage, Tunisia, in 2021, and the second Ph.D. degree in Electrical and Computer Engineering from Texas A&M University (TAMU), College Station, TX, USA, in 2024.

He has eight years of hands-on experience in applying deep learning and machine learning strategies to tackle real-world problems. During his work at Texas A&M University at Qatar, he is the lead author of more than 50 peer-reviewed journal and conference publications and three book chapters. His research interests include AI applications for demand flexibility, cybersecurity in smart grids, and innovative prediction models. Dr Massaoudi was the recipient of the Outstanding Student Research Excellence Award in 2021, the Thomas W. Powell'62 and Powell Industries Inc., Fellowship award in 2024, the best Paper Presentation Recognition in the IECON24, and the Richard E. Ewing Award for Excellence in 2024 for his research contributions. His h-index is 16, and his work has been cited more than 1200 times.

Maymouna Ez Eddin received the B.Sc. degree in electrical engineering from Qatar University, Doha, Qatar, in 2020, and the M.Sc. degree in data science and engineering from Hamad Bin Khalifa University, Doha, in 2022. She received the Ph.D. degree in electrical and computer engineering at Texas A&M University, College Station, TX, USA.

She worked as a Research Assistant with Qatar University from 2020 to 2022, with the Qatar University Machine Learning Group. She is currently working as a Teaching Assistant with Texas A&M University at Qatar, Doha. Her research interests include the application of deep learning and machine learning in smart grid security, and healthcare.

Haitham Abu-Rub (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Technical University of Gdansk, Gdansk, Poland, in 1995, and the Ph.D. degree in humanities from Gdansk University, Gdansk, Poland, in 2004.

He has worked at many universities in many countries including Poland, Palestine, USA, Germany, and Qatar. Since 2006, Dr. Abu-Rub has been with Texas A&M University at Qatar. For five years, he has served as the Chair of the Electrical and Computer Engineering Program at Texas A&M University at Qatar and currently serving as the managing director of Smart

Grid Center. He has published more than 600 journal and conference papers, five books, and six book chapters. He has supervised many research projects on smart grid, power electronics converters, and renewable energy systems. His main research interests include electric drives, power electronic converters, renewable energy and smart grid.

Dr. Abu-Rub was the recipient of many prestigious national and international awards and recognitions, such as the American Fulbright Scholarship and the German Alexander von Humboldt Fellowship. He is the Coeditor-in-Chief for IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.

ALI GHRAYEB (Fellow, IEEE) received the Ph.D. degree in electrical engineering from The University of Arizona, Tucson, AZ, USA, in 2000.

He is currently a Professor with the Department of Electrical and Computer Engineering, Texas A&M University at Qatar. Prior to his current position, he was a tenured professor in the Electrical and Computer Engineering Department at with Concordia University, Montreal, QC, Canada. He has co-authored two books and published over 250 journal and conference papers. His research interests include wireless and mobile communications, physical layer security, massive MIMO, visible light communications, smart grid, artificial intelligence and machine learning. He served as an Instructor or co-Instructor in many technical tutorials at several major IEEE conferences. He served as the Executive Chair of the 2016 IEEE WCNC Conference. He served as a member of the IEEE ComSoc Conferences Council, a member of the IEEE GITC Committee, and a member of the IEEE WCNC Steering Committee. He served in different editorial capacities on a number of IEEE transactions journals. He currently serves on the IEEE ComSoc Awards Committee. He is a Fellow of the IEEE.

Katherine Davis (Senior Member, IEEE) received a B.S. in electrical engineering from The University of Texas at Austin, Austin, TX, USA, in 2007, and an M.S. and Ph.D. in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2009 and 2011, respectively. She is currently an Associate Professor of electrical and computer engineering with Texas A&M University, College Station, TX, USA. Her research interests include the operation and control of power systems, interactions between computer networks and power networks, security-oriented cyber–physical analysis techniques, and data-driven and model-based coupled infrastructure analysis and simulation.