Delay Aware Power System Synchrophasor Recovery and Prediction Framework

James J.Q. Yu, *Member, IEEE*, Albert Y.S. Lam, *Senior Member, IEEE*, David J. Hill, *Life Fellow, IEEE*, Yunhe Hou, *Senior Member, IEEE*, and Victor O.K. Li, *Fellow, IEEE*

***Abstract*—This paper presents a novel delay aware synchrophasor recovery and prediction framework to address the problem of missing power system state variables due to the existence of communication latency. This capability is particularly essential for dynamic power system scenarios where fast remedial control actions are required due to system events or faults. While a wide area measurement system can sample high-frequency system states with phasor measurement units, the control center cannot obtain them in real-time due to latency and data loss. In this work, a synchrophasor recovery and prediction framework and its practical implementation are proposed to recover the current system state and predict the future states utilizing existing incomplete synchrophasor data. The framework establishes an iterative prediction scheme, and the proposed implementation adopts recent machine learning advances in data processing. Simulation results indicate the superior accuracy and speed of the proposed framework, and investigations are made to study its sensitivity to various communication delay patterns for pragmatic applications.**

I. Introduction

Modern power system operation and control require accurate and timely update of the system states. Conventionally these states were generated with the help of the Supervisory Control and Data Acquisition (SCADA) system, which samples the grid every 2-6 seconds. However, changes in modern systems, including the ever-increasing adoption of renewable energy sources and demand-side control, lead to new requirements for a faster system measurement technique [1]. As an alternative, modern Wide Area Measurement Systems (WAMS) equipped with measurement devices such as Phasor Measurement Units (PMU) are being gradually utilized [2]. These synchrophasor measurement devices can provide accurate, network-wide synchronized system state measurements of positive-sequence voltage and current phasors at a high resolution [2], [3].

Thanks to the sampled synchrophasors, grid operators can effectively improve the accuracy of both offline applications, such as post-disturbance analysis [4], [5], system identification [6], and online data-driven tasks, e.g., state estimation [7], [8]. Moreover, the power utilities are enabled to address conventional power system problems in a more advanced and timely manner [2], see [9], [10] for examples. For synchrophasor applications, the communication quality-of-service of PMU is among the essential factors that need to be considered to facilitate power system applications [11], [12]. According to North American Synchrophasor Initiative (NASPI), the data quality issue can be characterized by three properties, namely, data accuracy, data availability (data loss), and data timeliness (data latency) [13], [14]. Furthermore, power applications may require different levels of data quality, and various ones depend on timely synchrophasor data [15]. The stochastic packet drops and data transmission latency can significantly impact the system state data integrity at the control center, which in return influences the response time of these applications [11], [16].

However, as analyzed in [16]–[18], the current communication infrastructure of WAMS cannot guarantee a satisfactory QoS. In addition, with the expansion of power grids and introduction of new power electronics, the volume of data to be transmitted over the communication infrastructure is increasing drastically [12]. As a result, packet drops and notable data transmission latency can be expected, which leads to missing data in synchrophasor measurements. This problem is widely recognized in the previous literature, see [17], [19]–[21] for examples.

In common reliable communication networks, e.g., TCP/IP, data loss issues can be addressed by packet re-transmission. However, this will further increase the overall data transmission latency, rendering a larger response time for power system applications [17]. It becomes a critical issue to develop techniques that can recover missing synchrophasors with high accuracy without data re-transmission [19]. There exists previous work investigating solutions to handle missing PMU data, see [19], [22] for examples. The results mainly utilize the low-rank property of synchrophasors for data recovery [19]. However, as will be shown in Section VI, such low-rank-based methods cannot properly handle missing data caused by communication latencies since the available data may not satisfy the minimal obtained measurement requirement [19]. In addition, the employed synchrophasor completion methods involve data approximation, which can potentially undermine the data recovery accuracy.

In response to the research gap, in this work we propose a delay aware Synchrophasor Recovery and Prediction Framework (SRPF) to recover system states using existing incomplete synchrophasor data. The proposed methodology can recover missing synchrophasors subject to system faults, and does not require data re-transmission or approximation. In addition, it can be further employed to predict future system states. SRPF extracts the temporal and spatial relationships of the power system dynamics from historical data, and employs the result for real-time synchrophasor data processing. The generated system states can then be adopted for high-level system control and decision making. Moreover, the proposed framework is modular, with its constituent sub-systems able to have various implementations to meet the requirements of different applications. Existing related research can also be easily adopted into the framework subject to requirements.

The main contributions of this paper are:

- We propose a modularized framework to generate the past, present, and future system states utilizing existing incomplete synchrophasor data due to communication latency.
- We devise a practical implementation of the proposed framework using machine learning techniques by applying new machine learning and big data methods to synchrophasor data processing.
- We conducted case studies to illustrate the efficacy of the proposed framework and implementation. The robustness and sensitivity to latency are also investigated.

The rest of the paper is organized as follows. In Section II we introduce synchrophasor communication delay and the main problem to be addressed. Section III presents the system model employed. We elaborate on the proposed SRPF in Section VI, and devise a practical implementation of SRPF in Section V. Section VI records the numerical experiments, and Section VII concludes this paper.

II. Synchrophasors Delay

As expected by power utilities, future WAMS can directly provide dynamic views of power systems for supervision and control [2]. The system can lead to significantly improved state estimation performance. Moreover, thanks to the synchronized nature of synchrophasors, time skew errors can be prevented when the data is employed in power system applications. This time-synchronized characteristic is widely acknowledged by the literature, and it is common to assume that the control center can receive synchrophasors with zero latency caused by communication, see [7], [23]–[25] for recent examples.

While this assumption can significantly reduce the difficulty in system modeling and problem formulation, assuming zero delay will mean inappropriate control gets applied. Meanwhile, assuming all synchrophasors arrive together means that the delay is determined by the slowest one. According to IEEE Standard for Synchrophasor Data Transfer for Power Systems [3], a typical communication latency value from PMUs[1] to local Phasor Data Concentrators (PDC) ranges from 20 to 50 milliseconds, and extra delay can be expected for more complex WAMS structures. Moreover, due to the high sampling rate of PMUs, WAMS will generate more system data than the conventional SCADA. The large data traffic will introduce more data congestion in the supporting communication infrastructure, which may lead to stochastic latency spikes for synchrophasors. According to the analysis in [26], which investigated the GPS-synchronized wide-area FNET/GridEye records [27], "the communication delay may vary dramatically in short terms" and "the realtime communication delay presents strong dynamic characteristics". On average, the studied records experienced approximately 90ms of communication latency with random latency spikes and data losses. This level of delay may be insignificant for steady-state operation of power systems or "slow" stability issues such as voltage stability, but it can notably influence the system response for other time-sensitive issues, e.g., transient stability. What makes things worse is that existing literature on power system applications requires the control centers to have complete knowledge of the system states [28]. They need to wait for the slowest synchrophasor data packet in the system to arrive, which can be both time-consuming and unnecessary.

To address this problem, new methodologies should be developed to make use of incomplete synchrophasor data for delay-sensitive power system applications. Such techniques must 1) be capable to accept a stream of data as input and recover system states in real time, so as to adapt to the data stream of synchrophasors, 2) be computationally efficient to fulfill the fast response speed requirement of power applications, and 3) be "model-free", i.e., independent to the model of specific power system components. Such model-free methodologies can be applied to a wider range of systems as they do not rely on the model of specific components, rendering a better adaptability.

III. System Model

We model the power network with an undirected graph $G(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of buses and branches, respectively. For bus $i \in \mathcal{V}$, we denote the set of its connecting branches by $\mathcal{E}_i \subset \mathcal{E}$. In addition, neighboring buses of bus $i \in \mathcal{V}$ are defined by $\mathcal{N}_i = \{k \in \mathcal{V} | (i,k) \in \mathcal{E}_i\}$.

We use symbol t to denote the current time. In addition, we use $t_\tau, \tau \in \mathbb{Z}$ to represent the discrete time instances when PMUs generate system variable samples. For ease of further definitions, we use t_0 and t_1 to denote the previous and next "sampling" time instances. Therefore, $t_0 \leq t < t_1$ holds.

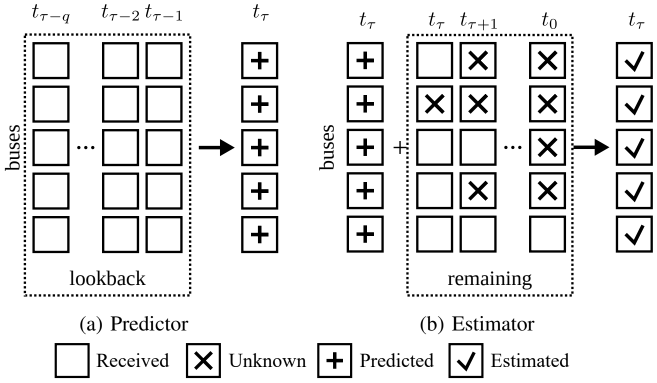
In the power system defined by $G(\mathcal{V},\mathcal{E})$, M PMUs are installed on system buses, which are denoted by set $\mathcal{V}^{\mathrm{M}} \subseteq \mathcal{V}$. We use $\mathcal{V}^{\mathrm{N}} = \{k \in \mathcal{N}_i | i \in \mathcal{V}^{\mathrm{M}}\}$ to denote the neighboring buses connected with these PMU-equipped buses. In this work, we employ a widely adopted PMU model [2], [29], i.e., each PMU measures the complex voltage phasor $V_{i,\tau}$ of its installation bus i at t_τ, and all the complex current phasors $I_{ik,\tau}$ of branch $(i,k) \in \mathcal{E}_i$. Considering the lumped-circuit model for transmission lines, voltage phasors of buses $k \in \mathcal{N}_i$ can be calculated by employing transmission line parameters: $V_{k,\tau} = V_{i,\tau} - (I_{ik,\tau} - y_{ik}^{sh} V_{i,\tau})/y_{ik}$, where y_{ik} and y_{ik}^{sh} are the series and shunt admittance of branch (i,k), respectively. Therefore, voltage phasors of $\mathcal{V}^{\mathrm{M}}$ can be sampled directly, and those of $\mathcal{V}^{\mathrm{N}}$ can be calculated with the aid of measured current phasors. We consider the investigated power system is fully observable, i.e., $\mathcal{V}^{\mathrm{M}} \cup \mathcal{V}^{\mathrm{N}} = \mathcal{V}$.

In this work, we consider the *system state* of the power grid at t_τ is composed of the voltage phasors of all buses in the network, denoted by $S_\tau = \{V_{i,\tau} | i \in \mathcal{V}\}$. The collection of voltage and current phasors directly measured by the measurement devices is denoted as $M_\tau = \{V_{i,\tau}, I_{ik,\tau} | i \in \mathcal{V}^{\mathrm{M}}, (i,k) \in \mathcal{E}_i\}$, called *full measurements*. At current time t, all M_τ for $\tau \leq 0$ have been sampled by PMUs. However, due to non-deterministic communication latencies, the control center only

[1] We use "PMU" to summarize all possible synchrophasor measurement units in current and future WAMS, such as branch PMU and line PMU, etc.

TABLE I
COMPARISON OF DIFFERENT SYSTEM STATE AND MEASUREMENT SETS

	$V_{i,\tau}, i \in \mathcal{V}^{\mathrm{M}}$	$I_{ik,\tau}, i \in \mathcal{V}^{\mathrm{M}}$	$V_{i,\tau}, i \in \mathcal{V}^{\mathrm{N}}$
System State	✓	×	✓
Full Measurement	✓	✓	×
Received State	✓(Incomplete)	×	✓(Incomplete)
Received Measurement	✓(Incomplete)	✓(Incomplete)	×



(a) Predictor
(b) Estimator

Fig. 1. Schema of SRPF sub-systems.

receives a subset of these measurements $M^-_{\tau,t} \subseteq M_\tau, \forall \tau \leq 0$, named *received measurements*.

Then, utilizing $M^-_{\tau,t}$, we can construct *received (measured) states* $S^-_{\tau,t} = \{V_{i,\tau} | i \in \mathcal{V}^{\mathrm{MP}}_{\tau,t} \cup \mathcal{V}^{\mathrm{NP}}_{\tau,t}\}$ for all $\tau \leq 0$. In this definition, $\mathcal{V}^{\mathrm{MP}}_{\tau,t} \subseteq \mathcal{V}^{\mathrm{M}}$ represents the buses whose system variables are sampled by the measurement unit, and received at the control center. The neighbor buses of $\mathcal{V}^{\mathrm{MP}}_{\tau,t}$ are defined as $\mathcal{V}^{\mathrm{NP}}_{\tau,t} = \{k \in \mathcal{N}_i | i \in \mathcal{V}^{\mathrm{MP}}_{\tau,t}\}$. Table III summarizes these terminologies, where check-marks denote inclusion and cross-marks denote exclusion.

Using the above defined notations, the main objective of this work can be interpreted as recovering and predicting $S_\tau, \tau \in \mathbb{Z}$ utilizing available $M^-_{\tau,t}$ at t.

IV. SYNCHROPHASOR RECOVERY AND PREDICTION FRAMEWORK

We propose a delay-aware SRPF to achieve the objectives summarized in Section III. According to the discussion in Section II, the methodology should have three characteristics, namely, 1) streaming input, 2) computationally fast, and 3) model-free. In this section, we first develop a modularized framework that can schematically recover and predict synchrophasors with streaming input. Then in the following section we propose an implementation of the framework to fulfill the second and third characteristics stated above.

The framework is designed to be an on-line process, i.e., real-time system states can be generated with known received measurements. For optimal recovery and prediction performance, the calculation process of the framework is executed whenever new synchrophasors are received by the control center.

A. Modular Sub-systems

The proposed SRPF is composed of two sub-systems, called *Predictor* and *Estimator*. Each system addresses a sub-problem of the stated key objective. The former is designed to utilize complete system states to predict a future system state, while the latter is defined to combine a predicted system state with the received synchrophasors for the same time.

Specifically, the predictor is designed to accept a sequence of *complete* system states $S_{\tau-1}, S_{\tau-2}, \cdots, S_{\tau-q}$ as input, and predict the system state for the next sampling time instance:

$$S^+_\tau = \mathrm{Predictor}(S_{\tau-1}, S_{\tau-2}, \cdots, S_{\tau-q}), \qquad (1)$$

where S^+_τ denotes a prediction of system state at time τ, i.e., S_τ. In this process, q is a user-controlled *lookback length* parameter. It instructs the predictor to "look" back by q sampling time instances from t_τ, and take all system states in the lookback window as input. Fig. 1a is an illustration of this system. The predictor should fully utilize the temporal and spatial relationships among synchrophasors to predict future system state, and its accuracy has a great influence on the performance of the complete system.

Meanwhile, the estimator combines S^+_τ with the remaining received states $S^-_{\tau,t}, S^-_{\tau+1,t}, \cdots, S^-_{0,t}$ to construct an estimation of S_τ at t, denoted by $\hat{S}_{\tau,t}$ (*estimated state*):

$$\hat{S}_{\tau,t} = \mathrm{Estimator}(S^+_\tau, S^-_{\tau,t}, S^-_{\tau+1,t}, \cdots, S^-_{0,t}). \qquad (2)$$

In case that there are no synchrophasors known in these received states, the estimated state $\hat{S}_{\tau,t}$ is set by S^+_τ. This will happen when predicting future system states, where no power system variables are sampled for future sampling time instances. Fig. 1b depicts the input and output of the estimator.

B. Framework Workflow

In Section IV-A, we define the input and output of the predictor and estimator, which enable SRPF to recover and predict unknown synchrophasors in an iterative manner. At t, new synchrophasors are received by the control center, and SRPF starts its recovery and prediction tasks. The pseudo-code for SRPF is presented in Algorithm 1.

The framework first constructs a matrix of system states, denoted by $\mathbb{S}$ (Line 1 in Algorithm 1). Each column of the matrix represents the received state of a sampling time instance. The unknown synchrophasors are left empty in the matrix. SRPF then looks back in $\mathbb{S}$ from the current time, and finds the last incomplete column, which represents the incomplete system state of the corresponding time instance (Line 2). The time is denoted by t_p, and the synchrophasors in all previous states are known, i.e., $S^-_{r,t} = S_r, \forall r < p$. Based on the value of p, we create a lookback window from t_{p-q} to t_{p-1}, in which all system states are complete (Line 3), and we focus on estimating the system state for time instance p, denoted by τ, the currently estimating time index (Line 4). This ends the initialization of SRPF

Next, the complete system states in the lookback window are input into the predictor to generate S^+_p using (1) (Line 6). The predicted system state is then combined with the remaining states $S^-_{p,t}, S^-_{p+1,t}, \cdots, S^-_{0,t}$ to develop an estimated state $\hat{S}_{p,t}$ using (2) (Line 7). The resulting estimated state is

Algorithm 1 SRPF

```
1: Construct a system state matrix $\mathbb{S}$.
2: Find the time instance $t_p$ that all previous states are complete.
3: Create a lookback window from $t_{p-q}$ to $t_{p-1}$ in $\mathbb{S}$.
4: Determine currently estimating time index $\tau$.
5: do
6:     Use the predictor to predict $S_\tau$ with lookback window, denoted by $S_\tau^+$.
7:     Use the estimator to estimate $S_\tau$ with $S_\tau^+$ and remaining states, denoted by $\hat{S}_\tau^+$.
8:     Insert $\hat{S}_\tau^+$ into $\mathbb{S}$.
9:     Move lookback window forward by one time instance.
10:    $\tau \leftarrow \tau + 1$.
11: while $\tau \leq \tau^{\text{end}}$
12: Output $\mathbb{S}$.
```

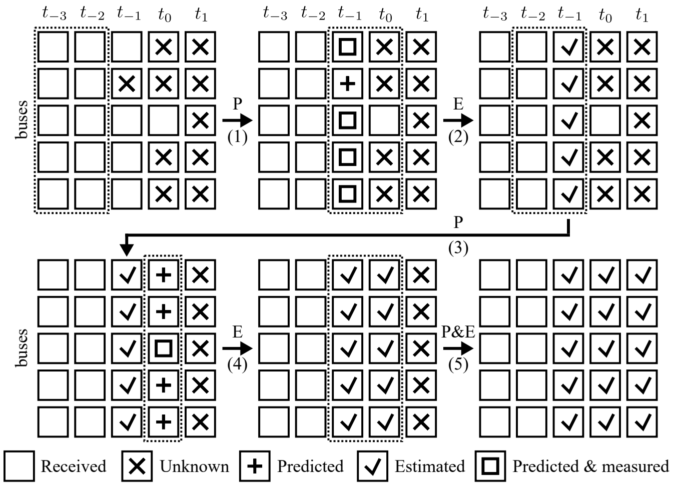


Fig. 2. Workflow of SRPF recovering and predicting synchrophasors. Numbers in brackets are the indices for steps. “P” stands for Predictor, and “E” stands for Estimator.

then inserted back into the corresponding column of system state matrix $\mathbb{S}$ (Line 8). This estimated state can be considered as a best estimation of the power system state at the investigated time instance. Consequently, the lookback window can move forward by one sampling time instance, and the included system states remain complete (Line 9). The current estimating time index τ is incremented by one (Line 10), and this ends the first iteration of SRPF. In the following iterations, the same synchrophasor recovery and prediction operations are repeated. This process iterates until $\tau > \tau^{\text{end}}$ (Line 11), where τ^{end} is the sampling time instance to which SRPF is designed to recover and predict.

Fig. 2 demonstrates the workflow of SRPF. At the current time, system states for t_{-2} and t_{-3} are complete, and those for t_0 and t_{-1} are not. t_1 refers to a time instance in the future, so no synchrophasor can be measured now. Assume that the lookback length $q = 2$. In this case, SRPF uses three iterations to recover and predict all unknown synchrophasors. In the first iteration, the predictor employs S_{-2} and S_{-3} to develop S_{-1}^+ (Step 1), which is combined with $S_{-1,t}^-$ by the estimator to generate $\hat{S}_{-1,t}$ (Step 2). This result is considered as S_{-1} in the following iterations. Then in the next iteration, the lookback window moves forward and uses S_{-1} and S_{-2} to develop S_0^+ (Step 3). The estimator subsequently generates $\hat{S}_{0,t}$ (Step 4). Finally, the predictor employs previous results to predict the synchrophasors at t_1, and the estimator simply uses the predicted state as the final system state S_1 (Step 5).

V. System Implementation

In Section IV, we proposed SRPF to recover and predict delayed synchrophasors in modern WAMS. The methodology recovers and predicts synchrophasors with streaming inputs. Meanwhile, two other required characteristics in Section II, namely fast-computing and model-free, remain open. These properties can be achieved by appropriate design of the subsystems in SRPF.

In this section, we devise an implementation for SRPF that can provide both characteristics. We employ the Gated Recurrent Unit (GRU) as the basic component to construct the predictor. As a modern variant of Artificial Neural Network (ANN), GRU naturally fits the needs, which will be elaborated later. Moreover, we develop an intuitive rule-based estimator for SRPF to estimate synchrophasors, which is effective and fast.

A. Artificial Neural Network and Gated Recurrent Unit

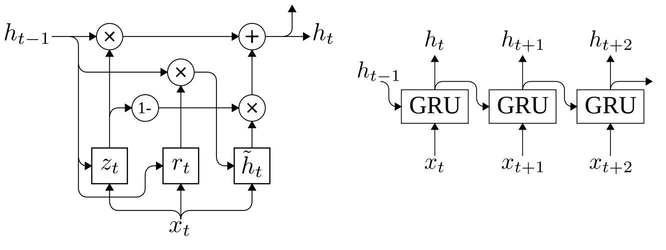
ANN is among the most commonly used machine learning techniques, and has been employed in a vast number of disciplines since its invention [30]. ANN tries to simulate investigated systems by learning from the mathematical relationship between the input and output of the system. Two major advantages make it suitable for being adopted into the predictor:

- Most computationally extensive process in ANN can be performed off-line, rendering a very fast response speed [30]. This makes it possible to conduct synchrophasor recovery and prediction in an online manner.
- Training ANN does not depend on the knowledge of analytical models of power system components such as thermal generators and wind turbines. As far as the training data describes the input-output relationship, ANN can emulate the system with arithmetic computations [31].

GRU is a modern variant of ANN [32]. It differs from the conventional ANN and its many variants in that besides spatial data correlation, it also extracts temporal data correlation in the training data to simulate the original system. Fig. 3 depicts the structure of a GRU block and its unrolled form. GRU separates the input data by timestamps, and accepts data of one timestamp, denoted by x_t, for each time. As presented in Fig. 3a, GRU calculates an output h_t using input x_t and its previous result h_{t-1} with the following equations:

$$z_t = \text{sigm}(W_{xz}x_t + W_{hz}h_{t-1} + b_z), \tag{3a}$$

$$r_t = \text{sigm}(W_{xr}x_t + W_{hr}h_{t-1} + b_r), \tag{3b}$$



(a) Computing graph

(b) Unrolled form

Fig. 3. Structure of GRU. Adopted from [33] with re-arrangements.

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h), \tag{3c}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \tag{3d}$$

where the W and b matrices are the GRU parameters [33], and $\odot$ denotes element-wise multiplication. Given a sequence of time-stamped data $\{x_1, x_2, \cdots, x_\tau, \cdots\}$, GRU first develops h_1 using x_1 and a randomly generated h_0 (usually initialized by zeros). Then it takes x_2 and h_1 to calculate h_2 and the process can be iterated until terminated by the user. This "chained" process is presented schematically in Fig. 3b. Note that all blocks labeled "GRU" in the figure share identical training parameter values.

B. GRU Ensemble-based Predictor

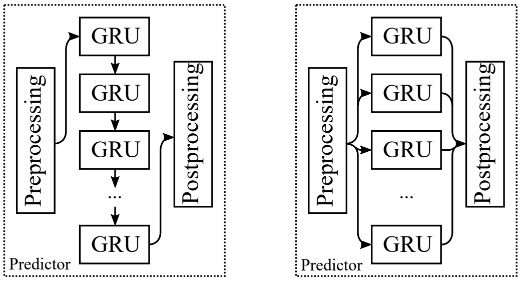
An ensemble of neural networks is an effective machine learning paradigm where multiple neural networks are constructed to solve the same problem. In general, an ensemble of networks can provide more robust results than standalone ones, and it has been demonstrated to outperform single neural networks in a collection of learning problems [34].

In this work, we devise a GRU ensemble-based predictor to predict unknown and future synchrophasors using available ones. The implemented predictor comprises N GRU networks, which contains two GRU layers as presented in Fig. 3 and another layer with fully connected neurons. In each network, the GRU layers extract the temporal and spatial relationship from the input x_t, and the calculated information is output in the form of h_t. The hidden layer then translates h_t to synchrophasors. So for the same synchrophasor $V_{i,\tau}$, N predictions are calculated from these networks, denoted by $V^+_{i,\tau,1}, \cdots, V^+_{i,\tau,N}$. The final predicted synchrophasor $V^+_{i,\tau}$, which is an entry in S^+_τ, is developed by

$$V^+_{i,\tau} = \frac{1}{N}\sum\nolimits_{n=1}^{N} V^+_{i,\tau,n}.$$

This concludes the data processing and calculation of the predictor.

1) Offline Predictor Training: As presented in (3), the arithmetic computations in GRU blocks involves GRU parameters. Their values can be calculated by training GRU networks with bus voltage phasors obtained from power system time-series simulations off-line. Considering a power system whose PMUs sample 60 system states per second, the time-stamped data x_τ is defined as $[\mathbf{V}_\tau, \boldsymbol{\theta}_\tau]$, where $V_{i,\tau} = \mathrm{V}_{i,\tau}\angle\theta_{i,\tau}$, $\mathbf{V}_\tau = [\mathrm{V}_{1,\tau}, \cdots, \mathrm{V}_{|\mathcal{V}|,\tau}]$, and $\boldsymbol{\theta}_\tau = [\theta_{1,\tau}, \cdots, \theta_{|\mathcal{V}|,\tau}]$. For any arbitrary t_τ in the time-domain simulation, the standardized $[x_{\tau-q}, \cdots, x_{\tau-1}]^\mathsf{T}$ is utilized as the input $\mathbf{x}$ of each training data case. The corresponding prediction output is accordingly constructed by standardized $[x_{\tau-q+1}, \cdots, x_\tau]^\mathsf{T}$ Consequently, the trained network can predict any arbitrary x_τ with $x_{\tau-q}, \cdots, x_{\tau-1}$, and thus satisfies (1).



(a) Sequential processing of each GRU network calculation in the predictor

(b) Parallel processing of each GRU network calculation in the predictor

Fig. 4. Parallel processing of the predictor.

The training process aims to optimize the system variables of the GRU blocks and neurons in each GRU network of the proposed predictor. GRU networks are trained simultaneously and independently. In this work we adopt the Adam optimizer [35] as the neural network training optimizer due to its superior convergence performance, and the objective is to minimize the mean-squared-error (MSE) of the prediction.

2) Online Prediction: The online prediction of synchrophasors is conducted in a similar manner as the training process. As described in Section IV-A, system states of t_{p-1} to t_{p-q} are complete. Therefore, utilizing the trained GRU parameters and (3), the system state of t_p instance calculated.

3) Parallel Processing and Prediction Caching: The computing process described above can serve the prediction requirement of the predictor. However, it is not optimal in terms of computational efficiency. In practice, SRPF may request the predictor to make predictions upon the receipt of any sampled synchrophasors. This can result in extensive computations within a very short time span. To address this problem, two practical solutions are proposed.

The first scheme makes the whole computing process parallel. As the most time-consuming task in the predictor is in calculating predictions, each GRU network can be handled by a separate computing unit, making the ensemble parallel. This scheme is depicted in Fig. 4.

The second one is to make use of previous prediction results when performing new predictions. If no system states are changed for a time instance and for its previous ones after receiving new synchrophasors, the predictions solely based on them remain unchanged. In such cases, these predictions, which are developed in previous calculations, can be considered as known in current calculations. Both schemes save a significant amount of time, which is illustrated in Section VI.

Note that the proposed predictor depends on the availability of historical data and/or time-domain simulated data as training data. If neither historical data nor power network topology/configuration is available, the mechanism cannot provide reliable predictions. However, in many modern power systems equipped with synchrophasor measurement devices, it is highly possible that the measured data can be properly stored. Therefore, the proposed mechanism can be applied to most power systems.

C. Estimator Implementation

As designed in Section IV, the estimator aims to merge predictions and PMU samples to create system state estimations. These sets of data have the following properties:

- *Predicted Synchrophasors* are complete predictions of system states, but they will have estimation error.
- *PMU Sampled Synchrophasors* are very accurate, but there are missing phasors in the system states as analyzed in Section II.

Utilizing these characteristics, in this work we consider a simplistic estimator implementation. For each synchrophasor in the investigated system state, if the corresponding PMU data is available at the control center, the PMU Sampled Synchrophasor is used. Otherwise, the predicted synchrophasor by predictor is considered accurate and placed in the estimated state.

This implementation is simple and intuitive, yet it can yield outstanding performance in terms of estimation accuracy and computation time, as demonstrated in Section VI. Note that in this design, only S_τ^+ and $S_{\tau,t}^-$ are utilized. In this way, the computational complexity is minimal at the expense of losing extra information in the remaining received states. It is possible to devise advanced schemes to employ all available information to develop $\hat{S}_{\tau,t}$, which is beyond the scope of this paper.

Note that in conventional power systems, residual-based bad data detection is carried out when all synchrophasors/SCADA measurements are received by the control center. Otherwise the residual cannot be computed. In the proposed mechanism, bad data detection can be conducted when the mechanism outputs an estimated state. However, due to prediction errors, the residual threshold should be larger than that in conventional bad data detections. This drawback can be resolved by incorporating more complex estimator design, such as appending a Kalman Filter after developing estimated states. We will investigate such estimator designs for the proposed mechanism in our future work.

VI. Case Study

To assess the performance of the proposed SRPF and its implementation, we conduct three sets of simulations on the New England 10-machine system [36]. In the first test, we study the synchrophasor data recovery and prediction accuracy, and analyze its sensitivity to the time of recovered/predicted system states. In the second test, we investigate the impact of communication delays on the system performance. Finally, we analyze the execution time of the proposed implementation as well as its time optimization schemes, and discuss the training time of the system.

In the simulation, we employ 30 GRU networks to construct the predictor, and each network has 512 GRU nodes in both of the GRU layers. All numerical simulations are conducted on a personal computer with an Intel Core i7 CPU working at 3.4 GHz and 8 GB RAM. The GRU networks are trained with an nVidia GTX 1080 graphic card with 8 GB RAM. The proposed SRPF is implemented with Python 3 on Linux.

A. Test System and Data Generation

In the New England 10-machine system, PMUs are installed on the following buses to maintain an $N-1$ observability: 2, 3, 6, 8, 10, 12, 14, 16, 17, 19, 20, 22, 23, 25, 26, 29, and 39. We assume that all PMUs measure 60 time-stamped system states in each second, and each one sends its measurements to the control center with stochastic communication latency. We consider the system dynamics of a large number of system faults that may potentially lead to transient system stability issues, as they are among the most important cases in power system operation: 5000 three-phase short circuit contingencies are generated with a random load level between 80% and 120%, a random fault clearance time between 0.1 second and 0.3 second, and a random fault location (on transmission lines or buses). The system dynamics of each contingency is calculated with time-series simulation using the TSAT software [37]. PMU measurements are developed based on the calculated true system dynamics. Furthermore, we adopt the real latency values derived from FNET/GridEye records [27] in this work, which are the voltage and frequency measurements by Frequency Disturbance Recorders (FDRs) across the States. For PMU measurements with the same timestamp, their latencies are randomly selected from the latency values of all FDRs at an arbitrary time. We adopt the open-loop latency values from the dataset, since SRPF aims to address the open-loop latency issue, i.e., from measurement devices to control center. After SRPF processes the received synchrophasors, the system operator may utilize the estimated system states for subsequent protection and control actions. The instructions are later sent back to control devices, which (closed-loop latency) is actually out of the scope of this paper and SRPF. All test cases are divided into two groups with a 3:1 ratio as suggested by previous literature, see [5], [9] for examples. The former is used for training, while the latter for testing. While insufficient samples cannot characterize the system properties, too many samples can significantly increase the training time, and may potentially lead to over-fitting problem. In this work, the training and testing set sizes are 3750 and 1250, respectively. The simulation results in the following sub-sections indicate that the selected sizes are appropriate. With this configuration, the over-fitting problem can be easily observed, where the training set yields superior accuracy but the testing set performance is unsatisfactory.

B. Synchrophasor Recovery and Prediction Accuracy

We first test the accuracy of recovered past synchrophasors and predicted future ones. For each of the testing cases, SRPF is executed whenever a new measurement arrives at the control center. The system recovers all incomplete system

TABLE II
SYNCHROPHASOR RECOVERY AND PREDICTION ACCURACY

	Voltage Magnitude Deviation (p.u.)				Voltage Angle Deviation (rad)				TVE	Percentage of Unknown Data
	Mean	Min.	Max.	RMSD	Mean	Min.	Max.	RMSD		
S_{-5}	4.822E-06	0.000E+00	2.229E-04	7.488E-06	3.518E-05	0.000E+00	1.048E-03	6.368E-05	0.02%	2.45%
S_{-4}	5.619E-06	0.000E+00	1.448E-03	2.406E-05	4.012E-05	0.000E+00	6.793E-03	2.282E-04	0.02%	6.22%
S_{-3}	3.683E-05	0.000E+00	9.501E-03	1.940E-04	2.971E-04	0.000E+00	2.652E-02	1.717E-03	0.11%	52.01%
S_{-2}	3.363E-04	0.000E+00	1.178E-02	6.291E-04	2.572E-03	0.000E+00	6.061E-02	5.193E-03	0.96%	76.18%
S_{-1}	6.547E-04	0.000E+00	1.185E-02	9.594E-04	6.039E-03	0.000E+00	1.511E-01	1.028E-02	1.03%	95.22%
S_0	8.716E-04	0.000E+00	1.460E-02	1.239E-03	9.058E-03	0.000E+00	1.569E-01	1.543E-02	1.04%	99.91%
S_1	1.076E-03	3.842E-08	1.722E-02	1.504E-03	1.149E-02	1.384E-08	1.547E-01	2.031E-02	1.07%	100.00%
S_2	1.281E-03	4.516E-08	1.897E-02	1.787E-03	1.334E-02	6.070E-07	1.845E-01	2.506E-02	1.10%	100.00%
S_3	1.485E-03	5.299E-08	1.890E-02	2.074E-03	1.531E-02	2.400E-09	1.974E-01	2.992E-02	1.11%	100.00%
S_4	1.692E-03	2.416E-07	2.023E-02	2.378E-03	1.695E-02	1.352E-06	2.187E-01	3.718E-02	1.12%	100.00%
S_5	1.912E-03	8.067E-08	2.672E-02	2.730E-03	1.798E-02	4.241E-06	2.120E-01	4.282E-02	1.14%	100.00%
S_6	2.133E-03	9.780E-08	3.239E-02	3.093E-03	2.022E-02	3.350E-07	2.118E-01	4.989E-02	1.16%	100.00%
S_7	2.356E-03	1.268E-07	3.228E-02	3.449E-03	2.176E-02	2.338E-06	2.155E-01	5.384E-02	1.19%	100.00%
S_8	2.592E-03	1.061E-07	4.732E-02	3.831E-03	2.401E-02	6.900E-07	2.125E-01	6.092E-02	1.25%	100.00%
S_9	2.812E-03	5.152E-08	5.237E-02	4.183E-03	2.602E-02	1.661E-07	2.162E-01	6.702E-02	1.44%	100.00%

state on and before t_0, and makes predictions from S_1 to S_9. For each missing synchrophasor in these system states, the recovered/predicted value is compared with the ground truth generated in time-series simulations. The mean, minimum, maximum, and root mean square deviation (RMSD) [8] of voltage magnitude and angle deviations between S_τ and $\hat{S}_\tau$ are calculated. In addition, we present the average total vector error (TVE) of voltage phasors in S_τ and $\hat{S}_\tau$. Let $\hat{S}_\tau = \{\hat{\mathrm{V}}_{i,\tau}\angle\hat{\theta}_{i,\tau} | i \in \mathcal{V}\}$. For voltage magnitude, we have:

$$\text{Mean} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} |\mathrm{V}_{i,\tau} - \hat{\mathrm{V}}_{i,\tau}|,$$

$$\text{RMSD} = \sqrt{\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} |\mathrm{V}_{i,\tau} - \hat{\mathrm{V}}_{i,\tau}|^2},$$

$$\text{TVE} = \frac{|V_{i,\tau} - \hat{V}_{i,\tau}|}{|V_{i,\tau}|}.$$

The summarized results are presented in Table II. In this table, each row presents the accuracy of SRPF in recovering the corresponding system state labeled in the first column at the recovery time. For reference, the percentage of unknown synchrophasors in the past system states at the current time is also listed in the last column.

From the simulation results it can be concluded that SRPF can accurately recover the past synchrophasors and predict the future ones. For both voltage magnitude and angle recovery, the estimation error is correlated with the percentage of unknown synchrophasors. When recovering the system states in which most synchrophasors have been received, the data recovered can be extremely accurate, e.g., S_{-3} to S_{-5}. With the increase of missing synchrophasor percentage, the accuracy is slightly undermined. However, even for S_0 where almost no measurement is known, there is only approximately 1.5×10^{-3} p.u. error in voltage magnitude estimation, and approximately $1.5\times10^{-2}\,\mathrm{rad}$ error in voltage angle estimation on average. Both cases demonstrate the capability of SRPF in recovering past synchrophasors.

SRPF is also outstanding at predicting future system states. The prediction accuracy goes down slightly with time, but the decrease is insignificant. This is mainly due to the fact that the system state predictions are calculated based on their previous predictions, and prediction errors will be amplified in later calculations. Despite this, the overall prediction accuracy is still satisfactory even for S_9 predictions, where the predicted synchrophasors will only suffer from $0.003\text{p.u.}\angle 1.5^\circ$ error on average.

TABLE III
COMPARISON ON SYNCHROPHASOR RECOVERY ACCURACY

	Total Vector Error			Recoverable Cases	
	SRPF	ICMC	SVT	SRPF	ICMC & SVT
S_{-5}	0.02%	0.01%	0.12%	100%	100.0%
S_{-4}	0.02%	0.14%	0.42%	100%	99.8%
S_{-3}	0.11%	1.46%	0.99%	100%	76.5%
S_{-2}	0.96%	3.43%	1.15%	100%	49.9%
S_{-1}	1.03%	4.69%	1.90%	100%	9.0%
S_0	1.04%	4.30%	1.73%	100%	1.8%

In the meantime, one may note that the average TVE values of S_{-1} and beyond exceed 1%, which is the required PMU measurement accuracy in [3]. Recall that the data quality issue involves data accuracy, availability, and timeliness [13], SRPF can obviously improve the data availability and timeliness at the expense of some data accuracy. Hence, the current implement of SRPF can be better employed in power system applications which rely more on fast measurements. Given that the developed synchrophasors are the only system state data system operators can obtain with the existing information, they can still be valuable in power system applications. Nonetheless, it is possible to improve the data accuracy with advanced design of both the predictor and estimator in the system, which will be investigated in the future work.

We also present the post-contingency voltage dynamics of some randomly selected buses, and the predicted dynamics using SRPF. The results of an arbitrary contingency are depicted in Fig. 5, where the true system state is plotted against the recovered current system state and five predictions at different times. In this figure, the curves labeled with "Recovery" represents the recovered system dynamics at the current timestamp, which corresponds to the performance of S_0 in Table II. Curves labeled with "τ-th Prediction" denotes the predicted dynamics at τ cycles ago. From the figure it can

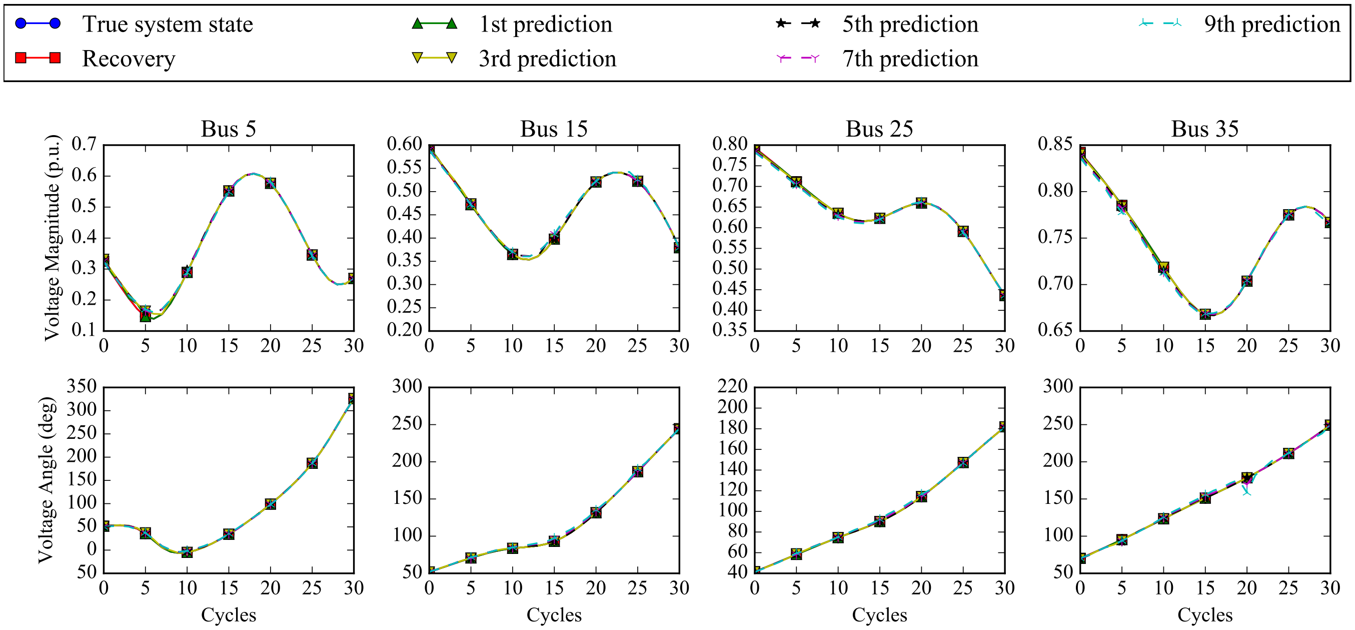


Fig. 5. Recovered and predicted system states for bus 5, 15, 25, and 35 of an arbitrary contingency case.

be observed that the accuracy for the recovery and different ahead-of-time predictions accords with the summarized results presented in Table II. While the recovery is very accurate compared with the true system state, the 9th prediction performs slightly worse. Yet, all predictions clearly resemble the trend of power system dynamics in the future.

Finally, we implement the previously proposed matrix completion methods in [19], namely, information cascading matrix completion (ICMC) and singular value thresholding (SVT), which are utilized to recover the previously tested incomplete system states. As illustrated in [19], such methods can only recover the system states with partial existing measurements. Hence, they cannot be applied to predict future states, and a majority of recent states, e.g., S_0 and S_{-1}, cannot be recovered since no measurements have been received yet. We present the simulation results of these methods in Table III, in which TVE and percentages of recoverable cases are summarized for SRPF and the compared methods. From the results it is obvious that SRPF outperforms SVT and ICMC in most tasks, with the only exception at S_{-5} where ICMC develops more accurate system states. Furthermore, neither ICMC nor SVT can recover system states without existing measurements. They can recover much less cases than the proposed method on recovering S_{-3} to S_0. Hence, SRPF is more suitable for recovering missing data caused by communication latencies.

C. Impact of Communication Delay on System Performance

In the previous test, we evaluate the accuracy of SRPF in WAMS with normal communication infrastructures. It is also of interest to investigate its sensitivity to bad communication conditions. In this test, we assess the accuracy of SRPF with different random latency generation schemes, which represents various practical data transmission performance of the power system communication infrastructures. Specifically, based on the benchmark delay data (labeled by "default") as introduced in Section VI-A, we manipulate the latency values and construct the following scenarios:

TABLE IV
SYSTEM PERFORMANCE WITH VARIOUS LATENCY PATTERNS

	Total Vector Error			
	Default	High Latency	Lossy Link	Bad Comm.
S_{-5}	0.02%	0.44%	0.09%	0.48%
S_{-4}	0.02%	0.73%	0.16%	0.75%
S_{-3}	0.11%	0.84%	0.23%	0.90%
S_{-2}	0.96%	1.01%	0.91%	1.07%
S_{-1}	1.03%	1.05%	1.02%	1.11%
S_0	1.04%	1.08%	1.09%	1.13%
S_1	1.07%	1.10%	1.11%	1.18%
S_2	1.10%	1.13%	1.13%	1.22%
S_3	1.11%	1.15%	1.16%	1.21%
S_4	1.12%	1.15%	1.18%	1.27%
S_5	1.14%	1.19%	1.21%	1.30%

- *High Latency*: In this setting, all latency values are increased by 50%. This setting emulates systems where the latency is significantly high and stable.
- *Lossy Link*: In this setting, the latency values are identical to the default scenario. However, 30% of the PMU measurement data packets are dropped and not recoverable. This setting emulates systems where the communication infrastructure is unstable.
- *Bad Communication*: In this setting, all latency values are increased by 50%, and 30% of the data packets are lost. This setting emulates a bad communication infrastructure which suffers from all previous issues, and can be considered as a worst case scenario.

We use the same experimental configurations as stated in Section VI-A for all latency configurations, and the simu-

TABLE V
EXECUTION TIME COMPARISON OF SRPF WITH DIFFERENT DEGREE OF TIME OPTIMIZATIONS

	Average Computational Time (ms) to	
	recover current state S_0	predict one future state
Benchmark	33.7	9.61
Parallel Processing	1.52	0.49
Prediction Caching	19.6	9.61
Full Optimization	0.85	0.49

lation results are presented in Table IV. Due to the space limitation, only TVE values are demonstrated, and system states of five future time instances are predicted since different latency patterns do not have great impact on the prediction accuracy. From the table it can be concluded that while bad communication links of WAMS have a negative impact on the synchrophasor recovery and prediction accuracy, the influence is not significant. Comparing with the *default* latency pattern, other settings generate comparable accuracies for S_{-2} to S_5 recovery and prediction. While the recovery accuracy on S_{-5} to S_{-3} is not as outstanding, it is still tolerable. This is due to the fact that the system states, whose percentage of unknown data is insignificant, tend to suffer more from larger latency values. Despite that, even in the worst case scenario (*Bad Communication*), the insignificant recovery and prediction error still suggests that SRPF can be applied to all kinds of WAMS regardless of the quality of their underlying communication infrastructures.

D. Execution Time Analysis

One of the main objectives of the proposed SRPF is to provide timely system states for system operators and other power system applications. So the execution time for SRPF is critical. In Section V-B we propose two practical schemes to boost the computational speed of the predictor, namely *Parallel Processing* and *Prediction Caching*. In this subsection we investigate the synchrophasor recovery and prediction speed of SRPF, and the improvements resulting from both of the proposed schemes.

We employ the original design without any execution time optimization schemes as *benchmark*, and compare its execution time with 1) *Parallel Processing*, 2) *Prediction Caching*, and 3) *Full Optimization* with both schemes enabled. The simulation results are demonstrated in Table V.

From the comparison it is clear that both time optimization schemes significantly contribute to the execution time reduction. On average, *Parallel Processing* can shorten the computation time of synchrophasor recovery to around one-thirtieth of *benchmark* test, and *Prediction Caching* can introduce a further one-third reduction. On the one hand *Prediction Caching* focuses on alleviating the repetitive computation in recovering the past synchrophasors, it does not have an influence on the data prediction speed. On the other hand, *Parallel Processing* effectively reduces the GRU ensemble testing time, thus also helps boost SRPF prediction speed. To conclude, the calculation speed with both optimization schemes is capable of providing real-time system states.

Last but not least, the training speed of SRPF is also critical. As the training data already considers random operation conditions, the trained SRPF can effectively cope with changing power system conditions without modifying its model. Meanwhile, significant power network changes, such as topology changes and power electronic component upgrades, may adversely influence SRPF accuracy. In such cases, the mechanism needs to be re-trained. Thanks to the relatively simple neural network structures, the proposed SRPF implementation can be re-trained with new training dataset within 30 minutes with GRU ensemble trained in parallel. Hence, the mechanism can handle topology changes with time intervals larger than 30 minutes, which in many cases are planned by the system operator hours or days before. This renders SRPF capable of adapting to significant changes in the grid, making it a pragmatic solution to address the delayed synchrophasor issue in modern power systems. We will also investigate methodologies to update the neural networks instead of re-training in order to further reduce the computation cost brought by topology changes in future studies.

VII. CONCLUSION

In this paper we propose a novel synchrophasor recovery and prediction framework to address the data transmission delay of synchrophasor measurements in modern power systems. As typical emergent power system data-centric applications can suffer from communication latency, the proposed framework aims to recover real-time power system states and predict future states. The framework is composed of two modular sub-systems, namely, predictor and estimator, which cooperate to manipulate available synchrophasor data and recover the unknown ones. To demonstrate the efficacy of the proposed framework, we implemented SRPF employing recent advanced machine learning techniques. A series of simulations are conducted on the New England 10-machine system, and the results demonstrate the superior data recovery and prediction performance of SRPF with the proposed implementation. In addition, the impact of communication delay on the system performance is investigated, and the execution time analysis illustrated that the proposed framework can be executed in an online manner with parallelized design.

ACKNOWLEDGEMENT

This work was supported by the Theme-based Research Scheme of the Research Grants Council of Hong Kong, under Grant No. T23-701/14-N.

REFERENCES

[1] M. Gol and A. Abur, "A Robust PMU Based Three-Phase State Estimator Using Modal Decoupling," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2292–2299, Sep. 2014.

[2] F. Aminifar, M. Fotuhi-Firuzabad, A. Safdarian, A. Davoudi, and M. Shahidehpour, "Synchrophasor Measurement Technology in Power Systems: Panorama and State-of-the-Art," *IEEE Access*, vol. 2, pp. 1607–1628, 2014.

[3] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," IEEE Std C37.118.2-2011, Dec. 2011.

[4] P. Bhui and N. Senroy, "Online Identification of Tripped Line for Transient Stability Assessment," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2214–2224, May 2016.

[5] T. Guo and J. V. Milanovic, "Online Identification of Power System Dynamic Signature Using PMU Measurements and Data Mining," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1760–1768, May 2016.

[6] I. Kamwa and L. Gerin-Lajoie, "State-space system identification-toward MIMO models for modal analysis and optimization of bulk power systems," *IEEE Trans. Power Syst.*, vol. 15, no. 1, pp. 326–335, Feb. 2000.

[7] J. Zhao, G. Zhang, K. Das *et al.*, "Power System Real-Time Monitoring by Using PMU-Based Robust State Estimation Method," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 300–309, Jan. 2016.

[8] F. Aminifar, M. Shahidehpour, M. Fotuhi-Firuzabad, and S. Kamalinia, "Power System Dynamic State Estimation With Synchronized Phasor Measurements," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 2, pp. 352–363, Feb. 2014.

[9] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong, "Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system," *IET Gener. Transm. Distrib.*, vol. 9, no. 3, pp. 296–305, 2015.

[10] L. Wehenkel, T. V. Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for online transient stability assessment of power systems," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 789–800, May 1989.

[11] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Analysis of IEEE C37.118 and IEC 61850-90-5 synchrophasor communication frameworks," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Jul. 2016, pp. 1–5.

[12] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient Compression of PMU Data in WAMS," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2406–2413, Sep. 2016.

[13] R. Quint, K. Thomas, A. Silverstein *et al.*, "Synchrophasor maturity model – extracting value from synchrophasor investments," North American SynchroPhasor Initiative (NASPI), Tech. Rep., 2015.

[14] L. E. Miller, A. Silverstein, D. An *et al.*, "PMU data quality: A framework for the attributes of PMU data quality and a methodology for examining data quality impacts to synchrophasor applications," North American SynchroPhasor Initiative (NASPI), Tech. Rep., 2017.

[15] B. Braden, D. Lutter, A. Silverstein *et al.*, "NASPI 2014 survey of synchrophasor system networks – results and findings," North American SynchroPhasor Initiative (NASPI), Tech. Rep., 2014.

[16] Y. Wu, L. Nordstrom, and D. E. Bakken, "Effects of bursty event traffic on synchrophasor delays in IEEE C37.118, IEC61850, and IEC60870," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2015, pp. 478–484.

[17] B. P. Padhy, S. C. Srivastava, and N. K. Verma, "A Wide-Area Damping Controller Considering Network Input and Output Delays and Packet Drop," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 166–176, Jan. 2017.

[18] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Analysis of communication schemes for advanced metering infrastructure (ami)," in *2014 IEEE PES General Meeting*, July 2014, pp. 1–5.

[19] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing Data Recovery by Exploiting Low-Dimensionality in Power System Synchrophasor Measurements," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1006–1013, Mar. 2016.

[20] Y. Seyedi, H. Karimi, and J. Guerrero, "Centralized Disturbance Detection in Smart Microgrids With Noisy and Intermittent Synchrophasor Data," *IEEE Trans. Smart Grid*, in press.

[21] S. Pal, B. Sikdar, and J. Chow, "Real-time detection of packet drop attacks on synchrophasor data," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2014, pp. 896–901.

[22] Y. Chen, L. Xie, and P. R. Kumar, "Dimensionality reduction and early event detection using online synchrophasor data," in *2013 IEEE Power Energy Society General Meeting*, Jul. 2013, pp. 1–5.

[23] L. Zhu, C. Lu, and Y. Sun, "Time Series Shapelet Classification Based Online Short-Term Voltage Stability Assessment," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1430–1439, 2016.

[24] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li, "Intelligent time-adaptive transient stability assessment system," *IEEE Trans. Power Syst.*, in press.

[25] J. J. Q. Yu, Y. Hou, A. Y. S. Lam, and V. O. K. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Trans. Smart Grid*, in press.

[26] C. Huang, F. Li, D. Zhou *et al.*, "Data quality issues for synchrophasor applications part i: a review," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 342–352, Jul 2016. [Online]. Available: https://doi.org/10.1007/s40565-016-0217-4

[27] "Fnet server web display." [Online]. Available: http://fnetpublic.utk.edu/

[28] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, and V. O. K. Li, "Delay aware intelligent transient stability assessment system," *IEEE Access*, vol. 5, pp. 17 230–17 239, 2017.

[29] A. G. Phadke, B. Pickett, M. Adamiak *et al.*, "Synchronized sampling and phasor measurements for relaying and control," *IEEE Transactions on Power Delivery*, vol. 9, no. 1, pp. 442–452, Jan. 1994.

[30] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987.

[31] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[32] K. Cho, B. Van Merrinboer, C. Gulcehre *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

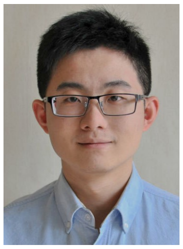
[33] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An Empirical Exploration of Recurrent Network Architectures," in *Proceedings of the 32 Nd International Conference on Machine Learning*, Lille, France, Jul. 2015, pp. 1–9.

[34] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

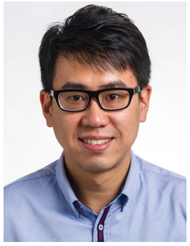
[35] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference for Learning Representations*, San Diego, Jul. 2015.

[36] A. Pai, *Energy Function Analysis for Power System Stability*. Springer Science & Business Media, 1989.

[37] "Dynamic Security Assessment Software," http://www.dsatools.com/.



James J.Q. Yu (S'11–M'15) received the B.Eng. and Ph.D. degree in Electrical and Electronic Engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He is currently an honorary assistant professor and post-doctoral fellow at the Department of Electrical and Electronic Engineering, The University of Hong Kong. He is also the Chief Research Consultant of GWGrid Inc. and Fano Labs. His research interests include smart city technologies, deep learning and big data industrial applications, and evolutionary computation.



Albert Y.S. Lam (S'03–M'10–SM'16) received the BEng degree (First Class Honors) in Information Engineering and the PhD degree in Electrical and Electronic Engineering from the University of Hong Kong (HKU), Hong Kong, in 2005 and 2010, respectively. He was a postdoctoral scholar at the Department of Electrical Engineering and Computer Sciences of University of California, Berkeley, CA, USA, in 2010–12. Now he is the Chief Scientist and the acting Chief Technology Officer at Fano Labs, and a honorary assistant professor at the Department of Electrical and Electronic Engineering of HKU. He is a Croucher research fellow. His research interests include optimization theory and algorithms, evolutionary computation, smart grid, and smart city.

David J. Hill (S'72-M'76-SM'91-F'93-LF'14) received the PhD degree in Electrical Engineering from the University of Newcastle, Australia, in 1976. He holds the Chair of Electrical Engineering in the Department of Electrical and Electronic Engineering at the University of Hong Kong. He is also a part-time Professor of Electrical Engineering at The University of Sydney, Australia. During 2005-2010, he was an Australian Research Council Federation Fellow at the Australian National University. Since 1994, he has held various positions at the University of Sydney, Australia, including the Chair of Electrical Engineering until 2002 and again during 2010-2013 along with an ARC Professorial Fellowship. He has also held academic and substantial visiting positions at the universities of Melbourne, California (Berkeley), Newcastle (Australia), Lund (Sweden), Munich and in Hong Kong (City and Polytechnic).

His general research interests are in control systems, complex networks, power systems and stability analysis. His work is now mainly on control and planning of future energy networks and basic stability and control questions for dynamic networks. Professor Hill is a Fellow of the Society for Industrial and Applied Mathematics, USA, the Australian Academy of Science, the Australian Academy of Technological Sciences and Engineering and the Hong Kong Academy of Engineering Sciences.. He is also a Foreign Member of the Royal Swedish Academy of Engineering Sciences.

Yunhe Hou (M'08–SM'15) received the B.E. and Ph.D. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2005, respectively. He was a Post-Doctoral Research Fellow at Tsinghua University, Beijing, China, from 2005 to 2007, and a Post-Doctoral Researcher at Iowa State University, Ames, IA, USA, and the University College Dublin, Dublin, Ireland, from 2008 to 2009. He was also a Visiting Scientist at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA, in 2010. Since 2017, he has been a Guest Professor with Huazhong University of Science and Technology, China. He joined the faculty of the University of Hong Kong, Hong Kong, in 2009, where he is currently an Associate Professor with the Department of Electrical and Electronic Engineering. Dr. Hou is an Editor of the IEEE Transactions on Smart Grid and Journal of Modern Power Systems and Clean Energy.

Victor O.K. Li (S'80–M'81–F'92) received SB, SM, EE and ScD degrees in Electrical Engineering and Computer Science from MIT. Prof. Li is Chair of Information Engineering and Cheng Yu-Tung Professor in Sustainable Development at the Department of Electrical & Electronic Engineering at the University of Hong Kong. He is the Director of the HKU-Cambridge Clean Energy and Environment Research Platform, an interdisciplinary collaboration with Cambridge. He was the Assoc. Dean (Research) of Engineering and Managing Director of Versitech Ltd. He serves on the board of Sunevision Holdings Ltd., listed on the Hong Kong Stock Exchange and co-founded Fano Labs Ltd., an artificial intelligence (AI) company with his PhD student. Previously, he was Professor of Electrical Engineering at the University of Southern California (USC), Los Angeles, California, USA, and Director of the USC Communication Sciences Institute. His research interests include big data, AI, optimization techniques, and interdisciplinary clean energy and environment studies. In Jan 2018, he was awarded a USD 6.3M RGC Theme-based Research Project to develop deep learning techniques for personalized and smart air pollution monitoring and health management. Sought by government, industry, and academic organizations, he has lectured and consulted extensively internationally. He has received numerous awards, including the PRC Ministry of Education Changjiang Chair Professorship at Tsinghua University, the UK Royal Academy of Engineering Senior Visiting Fellowship in Communications, the Croucher Foundation Senior Research Fellowship, and the Order of the Bronze Bauhinia Star, Government of the HKSAR. He is a Fellow of the Hong Kong Academy of Engineering Sciences, the IEEE, the IAE, and the HKIE.