IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 7, 2017, accepted December 19, 2017, date of publication December 22, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2786584

Robust Massively Parallel Dynamic State Estimation of Power Systems Against Cyber-Attack

HADIS KARIMIPOUR[1], (Member, IEEE), AND VENKATA DINAVAHI[2], (Senior Member, IEEE)
[1]Engineering Systems and Computing Group, School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada
[2]Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada

Corresponding author: Hadis Karimipour (hkarimi@uoguelph.ca)

This work was supported by the Natural Science and Engineering Research Council of Canada.

ABSTRACT The evolution of traditional energy networks toward smart grids increases security vulnerabilities in the power system infrastructure. State estimation plays an essential role in the efficient and reliable operation of power systems, so its security is a major concern. Coordinated cyber-attacks, including false data injection (FDI) attack, can manipulate smart meters to present serious threats to grid operations. In this paper, a robust state estimation algorithm against FDI attack is presented. As a solution to mitigate such an attack, a new analytical technique is proposed based on the Markov chain theory and Euclidean distance metric. Using historical data of a set of trusted buses, a Markov chain model of the system normal operation is formulated. The estimated states are analyzed by calculating the Euclidean distance from the Markov model. States, which match the lower probability, are considered as attacked states. It is shown that the proposed method is able to detect malicious attack, which is undetectable by traditional bad data detection (BDD) methods. The proposed robust dynamic state estimation algorithm is built on a Kalman filter, and implemented on the massively parallel architecture of graphic processing unit using fine-grained parallel programming techniques. Numerical simulations demonstrate the efficiency and accuracy of the proposed mechanism.

INDEX TERMS Bad data detection, cyber-attack, false data injection, dynamic state estimation, graphic processing units, large-scale systems, Markov chain, parallel programming, SCADA, PMUs.

I. INTRODUCTION

A smart power grid is a typical cyber-physical system (CPS) which integrates a physical power transmission system with the cyber computation and communication infrastructure. Although the advancement of cyber technologies in sensing, communication and smart measurement devices significantly enhanced power system operation and reliability, its dependency on data communication makes it vulnerable to cyber-attacks [1]. Coordinated false data injection (FDI) attacks [2] manipulate power system measurements in a way that emulate the real behaviour of the system and remain unobservable, which misleads the state estimation process, and may result in power outages and even system blackouts. Different aspects of constructing FDI attacks and their effect on the system are comprehensively reviewed in [3] and [4].

The increasing demand for reliable and economical electricity services raises critical challenges in online monitoring and control of future power grids which rely on dynamic state estimation (DSE) [6]; therefore, security of DSE and its vulnerability to cyber-attack is a major concern.

Detecting and identifying bad data in state estimation is traditionally or conventionally done by comparing the telemetered measurements from supervisory control and data acquisition system (SCADA) with the estimated values of the states. Traditionally, bad data are assumed to be caused by random errors resulting from a fault in a meter and/or its attendant communication system [8]. These errors were modeled by a change of variance in the Gaussian noise, which is detectable using Chi-squares and largest normalized residuals (LNR) test. Many researchers have considered the problem of bad data detection (BDD) in power systems, however conventional BDD approaches usually fail when the network malfunction is intentionally caused by an attacker [9], [10].

Vulnerabilities of power system to cyber-attacks can be classified into three main categories [11], [12]:

- *Data integrity analysis*- this research area investigates the possibility of the attacks from the attacker's point of view by exploiting weaknesses in BDD techniques. The problem of unexpected cyber-attack and its automatic recovery based on residual signals is investigated in [13]. Using computational intelligence technique [14] proposed a detection method to identify compromised data belonging to critical infrastructures. A detection algorithm is proposed in [15] to identify anomalous behavior from a compromised relay using the generator's dynamic state estimation. This problem is also formulated as identification of a subset of the measurements which are more vulnerable and easier to be attacked [17]–[19]. The results show that FDI attacks are easier to detect using the dc model of the system compared to the ac model. However, considering the large size of electric grids, selecting such subsets is a highly complex and computational intensive problem.
- *Consequence analysis*- the effect of false data attack within different functions of the energy management systems and smart grids such as optimal power flow calculations, congestion analysis, automatic generation control and energy pricing is investigated in this research area [20]–[22]. The current situation of security in industrial control systems considering a satisfactory degree of security for a distributed industrial system, with respect to the system characteristics, and the current standardization techniques and the adoption of suitable controls is presented in [23]. The impacts of cyber-attacks on the tie-line, frequency measurements, and on generator DSE are investigated in [24] and [25].
- *Attack prevention analysis*- specifically this research area is interested in finding the critical measurements and protecting them by improving the security of the communication system. In order to identify the vulnerability of a power grid, conditions were defined to quantify the minimum number of measurements required for a stealth attack [26]. Also, efforts have been made to develop a security-oriented cyber-physical state estimation framework using off-line information in [27] and [28] which could identify the compromised set of measurements. A risk mitigation strategy based on the DSE is proposed in [29] to eliminate threat levels from the potential cyber-attacks. To reduce the chance of successful attack, a game- theoretic framework and a security gateway was introduced in [30] and [31], respectively, to investigate the optimal strategies for both the sensor and the attacker.

One important fact which is neglected in the above works is that the cyber-security analysis should be performed in a timely manner, in order to solve the data attack construction problem efficiently. Otherwise it will slow down the process of state estimation, online monitoring and control of the system behaviour. In such cases even if the attack is detected, there is no time to take an action and prevent further casualties. Another main concern related to most of the above approaches is that they are not tested on large-scale power systems so the complexity and efficiency of the proposed approaches in practical systems is unclear. Overall, the computational complexity of the proposed approaches grows exponentially with the size of the power network which may make them unpractical for realistic systems.

The main motivation of this work is to design an attack detection method based on the history of the network behavior to be implemented on the massively parallel architecture of the graphic processing unit (GPU). An important fact in this type of analysis is to reduce the execution time as much as possible to save time for preventive action to take place. So we need a simple and effective method which matches the single instruction multiple data (SIMD) architecture of the GPU to accelerate the whole process. The existing methods in cyber-attack analysis are mathematically complicated which makes them difficult for GPU implementation. To overcome the effect of cyber-attacks, in this paper considering the stochastic nature of the system disturbances a cyber-physical model of the power system utilizing the Markov chain theory [32] is proposed. To the best of the author's knowledge such work has not been reported yet. It is the first time that robust state estimation against cyber-attack is modeled using Markov-chain and implemented on GPU. The power system can be modeled as a stochastic hybrid dynamical system where the stochastic nature of generation and state is explicitly included. Considering the results of state estimation from a group of trusted measurements, a set of possible states along with the probability of each state is generated. A Markov chain based on these states is then defined. After each estimation process all states are checked on the Markov chain. If the estimated states are close to a value with low probability or out of the Markov chain, the possibility of the cyber-attack is deemed high. Furthermore, to increase the security of the system, critical measurements are identified and protected. Changing the critical measurements using updated information decreases the chance of successful cyber-attack. The proposed attack detection methodology was built upon a parallel Kalman filtering algorithm for DSE. In order to speed up the whole process, the proposed robust DSE is implemented on GPU which are specially designed to deal with large amount of data. GPUs have already found applications for accelerating different power system applications [33], [34]. In summary, the main contributions of the proposed approach are as follows:

- Massively parallel implementation of robust DSE against FDI.
- Critical measurements identification and protection.
- Localization of false data injection attack using Markov chain model.

The organization of this paper is as follows. Section II provides formulation and the state estimation model used in this work. Section III explains the proposed robust parallel DSE against FDI and its implementation. The simulation

results are provided in Section IV followed by conclusion in Section V.

II. MATHEMATICAL FORMULATION

A. DYNAMIC STATE ESTIMATION USING EXTENDED KALMAN FILTER

The state-space model of the power system DSE can be described as:

$$\mathbf{x}_{t+1} = \mathbf{F}_t\mathbf{x}_t + \mathbf{b}_t + \boldsymbol{\omega}_t, \quad \mathrm{E}[\boldsymbol{\omega}_t\boldsymbol{\omega}_t^T] = \mathbf{O}_t, \tag{1}$$

$$\mathbf{m}_{t+1} = \mathbf{h}(\mathbf{x}_{t+1}) + \boldsymbol{\varepsilon}_{t+1}, \quad \mathrm{E}[\boldsymbol{\varepsilon}_t\boldsymbol{\varepsilon}_t^T] = \mathbf{R}_t, \tag{2}$$

where $\mathbf{x}$ is the vector of system states comprised of voltage magnitudes and phase angles at all buses except the slack bus where $\mathrm{V}_1 = 1\angle 0°$ p.u. For a system with n buses and m lines, $\mathbf{F}$ represents the $(2n-1)\times(2n-1)$ state transition matrix between two time frames, $\mathbf{b}$ is the $(2n-1)\times 1$ vector representing the behavior of the state trajectory, and $\boldsymbol{\omega}$ is the $(2n-1)\times 1$ Gaussian noise vector with zero mean and covariance matrix $\mathbf{O}$. Sudden changes due to different types of the noises will not affect the results. $\mathbf{h}(\mathbf{x})$ and $\mathbf{m}$, are vectors of nonlinear measurement functions and measurement data, respectively. Phase angle of the slack bus is considered 0 as a reference for other buses, so that there are $2n-1$ states to be estimated. There are $2m+2n+1$ elements in each measurement vector: $2m$ power flows, $2n$ power injections, and slack bus measurements. $\boldsymbol{\varepsilon}$ is the measurement noise assuming normal distribution with zero mean, and $\mathbf{R}$ is the $(2m+2n+1)\times(2m+2n+1)$ measurement error covariance matrix.

The objective function $\mathbf{J}(\mathbf{x})$ was chosen to minimize both estimation and prediction errors, given as:

$$\mathbf{J}(\mathbf{x}) = \underset{\mathbf{x}}{\operatorname{argmin}}[\mathbf{m}-\mathbf{h}(\mathbf{x})]^T\mathbf{R}^{-1}[\mathbf{m}-\mathbf{h}(\mathbf{x})] + [\mathbf{x}-\tilde{\mathbf{x}}]^T\tilde{\boldsymbol{\rho}}^{-1}[\mathbf{x}-\tilde{\mathbf{x}}], \tag{3}$$

The following equation satisfies the first-order optimality condition at the minimum of $\mathbf{J}(\mathbf{x})$:

$$\mathbf{g}(\mathbf{x}) = \mathbf{H}^{\mathrm{T}}(\mathbf{x})\mathbf{R}^{-1}[\mathbf{m}-\mathbf{h}(\mathbf{x})] - \tilde{\boldsymbol{\rho}}^{-1}[\mathbf{x}-\tilde{\mathbf{x}}] = 0, \tag{4}$$

where $\mathbf{g}(\mathbf{x})$ is the $(2n-1)\times 1$ vector of gradient of the objective function, and $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ is the $(2m+2n+1)\times(2n-1)$ Jacobian matrix. Expanding the non-linear function $\mathbf{h}(\mathbf{x})$ into its Taylor series around the state vector $\tilde{\mathbf{x}}_0$ results in the following:

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\tilde{\mathbf{x}}_0) + \mathbf{H}(\tilde{\mathbf{x}}_0)[\mathbf{x}-\tilde{\mathbf{x}}_0] + \ldots, \tag{5}$$

In general, $\tilde{\mathbf{x}}_0$ is the initial state or current predicted state of the system, which can be replaced by $\tilde{\mathbf{x}}$. Neglecting the higher order terms and substituting $\mathbf{h}(\mathbf{x})$ in Eq.(4):

$$\mathbf{G}(\mathbf{x})\Delta\mathbf{x} = \mathbf{H}^{\mathrm{T}}(\tilde{\mathbf{x}})\mathbf{R}^{-1}[\mathbf{m}-\mathbf{h}(\tilde{\mathbf{x}})],$$
$$\mathbf{G}(\mathbf{x}) = \mathbf{H}^{\mathrm{T}}(\tilde{\mathbf{x}})\mathbf{R}^{-1}\mathbf{H}(\tilde{\mathbf{x}}) + \tilde{\boldsymbol{\rho}}^{-1}, \tag{6}$$

where $\Delta\mathbf{x} = \hat{\mathbf{x}} - \tilde{\mathbf{x}}_0$ is the $(2n-1)\times 1$ state mismatch vector and $\mathbf{G}(\mathbf{x}) = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ is the $(2n-1)\times(2n-1)$ gain matrix. The state estimation algorithm given by (3)-(6) should be solved iteratively until convergence of $\Delta\mathbf{x}$ to a specified threshold.

In order to predict the estimation value, model of the system should be identified. In this work Holt's exponential smoothing technique is used. Considering the quasi-static nature of the power system, recent state of the system are closet one to the current state of the system. Exponential Smoothing assigns exponentially decreasing weights in a way that recent observations are given relatively more weight in forecasting than the older observations [35]. $\mathbf{F}$ and $\mathbf{b}$ are described as follows:

$$\mathbf{F}_t = \mathfrak{B}\mathbf{I}_{idn}, \quad 0 < \mathfrak{B} < 2,$$
$$\mathbf{b}_t = \mathfrak{C}\tilde{\mathbf{x}}_t - \mathfrak{D}\boldsymbol{\gamma}_{t-1} + (1+\mathfrak{D})\boldsymbol{\xi}_{t-1}, \quad 0 < \mathfrak{C} < 2,$$
$$\boldsymbol{\gamma}_t = \mathfrak{F}\hat{\mathbf{x}}_t + (1-\mathfrak{F})\tilde{\mathbf{x}}_t, \quad 0 < \mathfrak{F} < 1,$$
$$\boldsymbol{\xi}_t = \mathfrak{D}(\boldsymbol{\gamma}_t - \boldsymbol{\gamma}_{t-1}) + (1-\mathfrak{D})\xi_{t-1}, 0 < \mathfrak{D} < 1, \tag{7}$$

where $\mathfrak{B}$, $\mathfrak{C}$, $\mathfrak{D}$ and $\mathfrak{F}$ are smoothing parameters. To select the value of the smoothing parameters we search for values that minimizes the size of the combined forecast errors of the currently available series. The goal is to find the smoothing parameters that result in minimum forecast error. The forecast error is the difference between the forecast of the current period made at the last period and the value of the series at the current period. In this work, $\mathfrak{D}$ and $\mathfrak{F}$ were selected as 0.8 and 0.7, respectively. The other two smoothing parameters are functions of $\mathfrak{D}$ and $\mathfrak{F}$. The $\mathbf{I}_{\mathrm{idn}}$ is the $(2n-1)\times(2n-1)$ identity matrix. The $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ represent the predicted and estimated values, respectively.

Using the measurement and estimated states at the time instant t, the predicted value $\tilde{\mathbf{x}}_{t+1}$ can be formulated as:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{F}_t\hat{\mathbf{x}}_t + \mathbf{b}_t, (\mathbf{x}_t - \hat{\mathbf{x}}_t)\sim\mathrm{N}(0, \boldsymbol{\rho}_t),$$
$$\tilde{\boldsymbol{\rho}}_{t+1} = \mathbf{F}_t\boldsymbol{\rho}_t{\mathbf{F}_t}^T + \hat{\mathbf{O}}P_t, \quad (\mathbf{x}_t - \tilde{\mathbf{x}}_t)\sim\mathrm{N}(0, \tilde{\boldsymbol{\rho}}_t), \tag{8}$$

where $\boldsymbol{\rho}$ and $\boldsymbol{\rho}$ are $(2n-1)\times(2n-1)$ error covariance matrices for estimated and predicted values, respectively. For simplicity $\mathbf{O}$ is assumed to be constant.

Finally, utilizing extended Kalman filter (EKF) the predicted values can be updated using the next set of measurements at the time instant $t+1$. The updated state through EKF can be written as:

$$\hat{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(\mathbf{m}_{t+1} - \mathbf{h}(\tilde{\mathbf{x}}_{t+1})),$$
$$\mathbf{K}_{t+1} = \tilde{\boldsymbol{\rho}}_{t+1}\mathbf{H}^{\mathrm{T}}{}_{t+1}[\mathbf{H}_{t+1}\tilde{\boldsymbol{\rho}}_{t+1}\mathbf{H}^{\mathrm{T}}{}_{t+1} + \mathbf{R}]^{-1},$$
$$\boldsymbol{\rho}_{t+1} = \tilde{\boldsymbol{\rho}}_{t+1} - \mathbf{K}_{t+1}\mathbf{H}_{t+1}\tilde{\boldsymbol{\rho}}_{t+1}, \tag{9}$$

where $\mathbf{K}$ is the $(2n-1)\times(2m+2n-1)$ Kalman gain matrix.

B. BAD DATA DETECTION

Even under normal operating conditions the measurements may be corrupted by random errors. The process of detecting exceptional errors is called BDD. Traditionally BDD tries to detect measurements errors using the statistical properties of the weighted measurement residual. Generally, the presence

of bad data is determined if

$$r_i^N = \frac{|r_i|}{\sigma_{ii}} \leq \varkappa, \tag{10}$$

where r_i is the measurements residual, and r_i^N is the largest normalized residual (LNR), σ_{ii} is the standard deviation of the i^{th} component of the residual vector and $\varkappa$ is the threshold [36].

It should be noted that measurement redundancy is a key issue in the performance of BDD which means it is necessary to have more measurements than the minimum number required for system observability. However, existing measurement configurations may not always yield such desired level of redundancy which makes the BDD impractical.

C. FALSE DATA INJECTION ATTACK

In false data injection (FDI) attacks, the adversary who has the knowledge of the network configuration can inject some of the meter readings from SCADA and manipulate the state variables arbitrarily. This type of malicious attacks can effectively bypass the existing BDD technique. The assumption is that: 1) attacker will compromise minimum number of measurements to achieve his goal, and 2) attacker have partial knowledge on system topology and security mechanisms.

The general rule for a hidden attack is that the attacker must alter the data so that the measurements can plausibly correspond to the physical properties of the system. The main idea of FDI attack is to add a nonzero attack vector $\mathbf{a}$ to the original measurements vector $\mathbf{m}$ which results in a false estimation $\hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c}$ is the error added to the original estimation [2], [16], [18]. Considering the measurement residual, a necessary condition to hide an attack can be derived as follows:

$$\begin{aligned}\Delta\hat{\mathbf{x}}_a &= \mathbf{G}^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{m}_a - \mathbf{h}] = \mathbf{G}^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{m} + \mathbf{a} - \mathbf{h}],\\ &= \underbrace{\mathbf{G}^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{m} - \mathbf{h}]}_{\Delta\hat{\mathbf{x}}} + \underbrace{\mathbf{G}^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{a}]}_{\mathbf{G}^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{Hc}},\\ &= \Delta\hat{\mathbf{x}} + \mathrm{c} \;\rightarrow\; \hat{\mathbf{x}}_a = \hat{\mathbf{x}} + c \end{aligned} \tag{11}$$

$$\begin{aligned}\mathbf{r}_a &= \|\mathbf{m}_a - \hat{\mathbf{m}}_a\| = \|\mathbf{m}_a - \mathbf{H}\hat{\mathbf{x}}_a\| = \|\mathbf{m} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\|\\ &= \|\mathbf{m} - \mathbf{H}\hat{\mathbf{x}} + \underbrace{(\mathbf{a} - \mathbf{Hc})}_{\mathrm{a=Hc}}\| = \|\mathbf{m} - \mathbf{H}\hat{\mathbf{x}}\|. \end{aligned} \tag{12}$$

where $\Delta\hat{\mathbf{x}}_a$ refers to attacked state mismatch vector, $\mathbf{m}_a$ is the measurement vector under the attack, $\hat{\mathbf{m}}_a$ refers to estimated measurement vector under the attack, and $\hat{\mathbf{x}}_a$ represent the attacked state.

The above equality constraint results in $\mathbf{a} = \mathbf{Hc}$. A structured sparse attack like $\mathbf{a} = \mathbf{Hc}$ will result in the same residual and will not be detected by BDD. In this case, the system operator would mistake $\hat{\mathbf{x}} + \mathbf{c}$ for a valid estimate.

Definition: The sparse attack vector $\mathbf{a} = [a_1, \ldots, a_m]^{\mathrm{T}}$ is called false data injection attack if and only if it satisfies the relation $\mathbf{a} = \mathbf{Hc}$, where $\mathbf{c} = [c_1, \ldots, c_n]^{\mathrm{T}}$ is a arbitrary nonzero vector [2].

Fig. 1 shows a possible cyber-attack on an energy control center. For example, assume that the attacker wants to alter

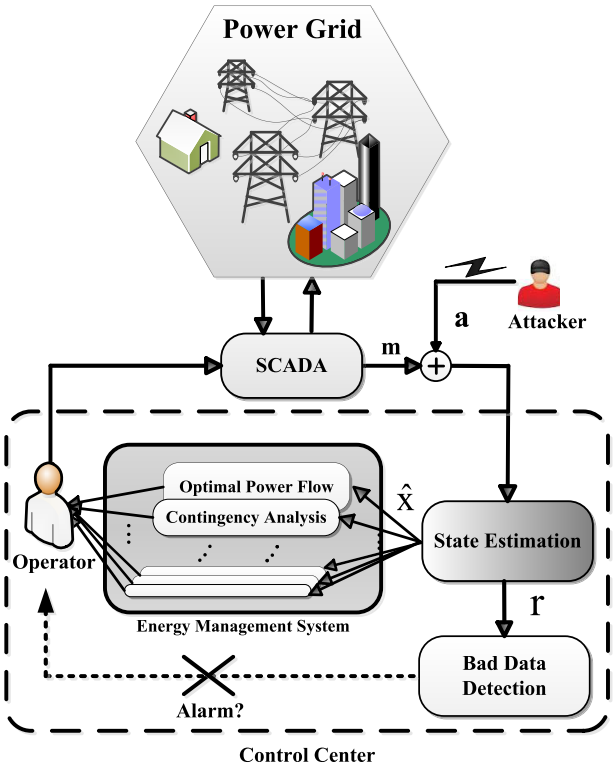

FIGURE 1. Dynamic state estimation under cyber-attack, a: attack vector, m: measurement, r: measurement residual, $\hat{\mathbf{x}}$: estimated state.

the active power flow on the line connecting bus i and j. Based on the following equation the attacker has to at least change one of the four variables, voltage magnitudes: $\mathrm{v_i}$, $\mathrm{v_j}$, and phase angles: θ_i, θ_j.

$$\mathrm{P_{ij}} = \mathrm{v_i^2}.\mathrm{g_{ij}} - \mathrm{v_i v_j}(\mathrm{g_{ij}}\cos(\theta_i - \theta_j) - \mathrm{b_{ij}}\sin(\theta_i - \theta_j)). \tag{13}$$

Imagine that the attacker wants to adjust the estimated value for $\mathrm{v_j}$ to $\mathrm{v_j^a}$, the following equation must be solved in order to find the voltage magnitude which will yield the desired power flow:

$$\mathrm{P_{ij}} = \mathrm{v_i^2}.\mathrm{g_{ij}} - \mathrm{v_i v_j^a}(\mathrm{g_{ij}}\cos(\theta_i - \theta_j) - \mathrm{b_{ij}}\sin(\theta_i - \theta_j)). \tag{14}$$

where $\mathrm{g_{ij}}$ and $\mathrm{b_{ij}}$ represent line admittance parameters. Since power flow and power injection are functions of voltage magnitudes and phase angles, the value of other measurements can be calculated considering the relationship between power flow and power injection. Also, the attacker must change all the measurements which are functions of $\mathrm{v_j}$. In another words, the following should be satisfied in order to keep the attack hidden:

$$\begin{aligned}\sum_i \Delta\mathrm{P_i} + \Delta\mathrm{L_P} &= 0,\\ \sum_i \Delta\mathrm{Q_i} + \Delta\mathrm{L_Q} &= 0, \end{aligned} \tag{15}$$

where ΔP and ΔQ represent the alterations in active power flow/power injection and reactive power flow/power injection, respectively. ΔL represents the power losses.

D. CRITICAL MEASUREMENT IDENTIFICATION

Large-scale power grids contain thousands of meters which makes the protection of measurements highly expensive. In order to reduce the cost, we identify the critical meters to protect them based on optimal PMU placement. Critical measurements are the ones whose elimination make the entire system unobservable. One of the important properties of critical measurements is that its measurement residual will always be zero [36]. Without loss of generality, we can define $\mathbf{\Delta m} = \mathbf{h(x)} + \boldsymbol{\varepsilon} \simeq \mathbf{H\Delta x}$. The measurement residual can be written as follows:

$$\begin{aligned}\mathbf{res} &= \mathbf{\Delta m} - \mathbf{\Delta \hat{m}} = \triangle \mathbf{m} - (\mathbf{h}(\hat{\mathbf{x}}) + \boldsymbol{\varepsilon}) \\ &\backsimeq \mathbf{\Delta m} - (\mathbf{H\Delta \hat{x}} + \boldsymbol{\varepsilon}).\end{aligned} \tag{16}$$

Without loss of generality, $\boldsymbol{\varepsilon}$ can be removed from the equation. Substituting (6) in (16) we obtain,

$$\mathbf{res} = \mathbf{\Delta m} - \mathbf{H}\mathbf{G}^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{\Delta m} = (\mathbf{I} - \mathbf{S})\mathbf{\Delta m} \tag{17}$$

For residual equal to zero, the diagonal element $\mathrm{S_{ii}}$ of matrix $\mathbf{S}$ should be 1, which implies that the $\mathrm{i^{th}}$ measurement is critical.

In the next step, by optimal PMU placement at strategic buses in the system, we try to increase the accuracy of the system, protecting most of the critical measurements, and providing a subset of trusted buses for use in the attack detection algorithm. In other words, PMUs work as backup measurements to increase the security of critical measurements. So those buses that are observable through PMU placements are considered trusted bus. The objective of PMU placement problem is to accomplish this task by using a minimum number of PMUs. This problem can be formulated and solved as shown below:

$$\begin{aligned}&\text{min.} \sum_{\mathrm{i}=1}^{n} \Lambda_{\mathrm{i}} \times \Gamma_{\mathrm{i}}, \\ &\text{subject to :} \sum_{\mathrm{j}=1}^{\mathrm{n}} \mu_{\mathrm{i,j}} \times \Gamma_{\mathrm{i}} \geq 1 \text{ at bus i}\end{aligned} \tag{18}$$

$$\Gamma_{\mathrm{i}} = \begin{cases} 1 & \text{If PMU is installed at bus i,} \\ 0 & \text{Otherwise,} \end{cases} \tag{19}$$

where $\mu_{\mathrm{i,j}}$ is the element of connectivity matrix which is 1 if bus i and bus j are connected, and 0 otherwise. Λ_{i} is the cost of PMU installation at bus i.

E. MARKOV-CHAIN FORMULATION

Consider a physical system that has k possible states and at any given time, the system is in one of its k states. Defining a set of states as C_i, for any $\mathbf{s}_i$, a stochastic process which fulfils the following properties is called an l^{th} order Markov-chain [32]:

$$\begin{aligned}\Pr(\boldsymbol{C}_{\mathrm{t+1}} &= \mathbf{s}_{\mathrm{i}} | \boldsymbol{C}_{\mathrm{t}} = \mathbf{s}_{\mathrm{i_0}}, \ldots, \boldsymbol{C}_0 = \mathbf{s}_{\mathrm{i_t}}) \\ &= \Pr(\boldsymbol{C}_{\mathrm{t+1}} = \mathbf{s}_{\mathrm{i}} | \boldsymbol{C}_{\mathrm{t}} = \mathbf{s}_{\mathrm{i_0}}, \ldots, \boldsymbol{C}_{\mathrm{t}-l} = \mathbf{s}_{\mathrm{i}_l})\end{aligned} \tag{20}$$

where $\mathrm{s_{i_0}}, \ldots, \mathrm{s_{i}}_l, \ldots, \mathrm{s_{i_t}} \in \mathrm{S}$, Pr refers to probability function and t represent the time. We are trying to model the system behavior based on the historical data. When there are only k possible state, k is the limit for the number of data that we can use in our Markov chain modeling. So, in this process the probability of getting into the next state depends upon the l previous states, where $l \leq k$.

To define a Markov model, the following probabilities have to be specified: the transition probability matrix $\mathbf{TP} = [\mathrm{tp_{ij}}]_{\mathrm{k\times k}}$ and initial probabilities $\pi_{\mathrm{i}}^{0} = \Pr(\boldsymbol{C}_0 = \mathbf{s}_{\mathrm{i}})$, however, in our case study we are dealing with l-th order Markov chain, so we do not use initial probability at time zero.

$$\mathrm{tp_{ij}} = \Pr(\boldsymbol{C}_{\mathrm{t+1}} = \mathbf{s}_{\mathrm{i}} | \boldsymbol{C}_{\mathrm{t}} = \mathbf{s}_{\mathrm{j}}), \quad j \in 0, 1, 2, \ldots, k \tag{21}$$

with $\sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{tp_{ij}} = 1$ and $\mathrm{tp_{ij}} \geq 0$. In this work tp_{ij} is the probability of future state (i) to be equal to j-th data set in historical data, so the total probability considering all historical data should add up to 1. In l-th order Markov chain the conditional probability of observing $\boldsymbol{C}_{\mathrm{t+1}} = \mathbf{s}_{\mathrm{i}}$ is a linear combination of contributions from each of $\boldsymbol{C}_{\mathrm{t}}, \ldots, \boldsymbol{C}_{\mathrm{t}-l}$. The effect of each previous state can be considered separately resulting in following conditional probability:

$$\begin{aligned}Pr(\boldsymbol{C}_{\mathrm{t+1}} &= \mathbf{s}_i | \boldsymbol{C}_t = \mathbf{s}_{i_0}, \ldots, \boldsymbol{C}_{t-l} = \mathbf{s}_{i_l}) \\ &= \sum_{j=0}^{l} \lambda_j Pr(\boldsymbol{C}_{\mathrm{t+1}} = \mathbf{s}_i | \boldsymbol{C}_{t-j} = \mathbf{s}_{i_j}) \\ &= \sum_{j=0}^{l} \lambda_j tp_{ii_j}\end{aligned} \tag{22}$$

where λ is the weight parameter considering the effect of each previous state separately, and $\sum_{j=0}^{l} \lambda_j = 1$.

F. DETECTION OF POTENTIAL ATTACK

The proposed parallel DSE using EKF calculates the state of the system using the equations described in Section II. The Euclidean distance of the historical data and estimation of the trusted buses are calculated. Historical data actually refers to possible state of the system which has been observed during long term system performance, therefore it acts like a database of the system behavior. Since the data behaviour changes over time, the historical data representing the normal behaviour is updated dynamically.

The Euclidean method compares the difference between the two sets of data $(\mathbf{x}_1, \mathbf{x}_2)$ based on the distance metric as given in (23):

$$\mathrm{ED}(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathrm{x_{1,1}} - \mathrm{x_{2,1}})^2 + \ldots + (\mathrm{x_{1,n}} - \mathrm{x_{2,n}})^2} \tag{23}$$

In the proposed method C and S represent set of states of trusted buses and estimated states, respectively. Transition probability is calculated as follows:

$$\mathrm{tp_{ij}^{f}} = \frac{\sum_{\mathrm{j=1,j\neq f}}^{M} \mathrm{ED(C_i, C_j)}}{(\mathrm{M}-1)\sum_{\mathrm{j=1}}^{M} \mathrm{ED(C_i, C_j)}}, \quad f \in 1, 2, \ldots, M \tag{24}$$

where $\sum_{j=1}^{k} tp_{ij}^{f} = 1$. M is the number of data set in historical data. C_i represent the current estimation of trusted buses and C_j refers to estimated state for the same buses in j-th historical data among all M available data set. In another words, higher probability is assigned to the measurement set with a smaller Euclidean distance. In the next step all of the projected estimates and the Markov model are compared by the detector. If the difference between two (the projected estimates and data with high probability in Markov model) is above a pre-computed threshold, an alarm is triggered to notify of a possible attack or failure. The results with probability less than 0.1 are assumed to be corrupted with cyber-attack. In other words the probability of the attack increases when the results match to data set with a lower probability.

The results of state estimation are compared with normal expected state of the system in Markov model. In case the difference between them is above the predefined threshold, the detector triggers an alarm. The threshold is defined considering the probability of state estimation in Markov model. The results with probability less than 0.1 are assumed to be corrupted with cyber-attack. However, to avoid false alarms due to measurement or system errors, the threshold was set to filter 99% of noise. The same criteria was also considered for the LNR test. The threshold and detector outputs for both LNR and the proposed method are normalized. Also, in case of the load change, the change in voltage magnitude or phase angle can be predicted, so that the model parameters can be adjusted to reflect the change in the voltage due to the load change. Fig. 2 shows the overall block diagram of the proposed method.

III. MASSIVELY PARALLEL IMPLEMENTATION OF THE ROBUST DSE AGAINST FDI

In this section, a trust-aware scheme for DSE is proposed that is robust under FDI attack. In the first step, critical measurements are identified. Utilizing optimized PMU placement for critical measurements, a group of trusted buses are introduced into the network. The assumption is that trusted measurements are secured and can not be affected by adversary. Considering high level of security and backup PMU installation on these measurements it is unlikely that the adversary can attack these measurements. Secondly, using historic data of the trusted buses normal activities a Markov-chain model representing the ''normal'' behavior of the network is created. Thus, given an observed sequence, the system has to decide if there is a cyber-attack or system in under normal operation condition. The Euclidean distance of the results from the Markov-chain model is then calculated. The higher the distance the observed activities receive from the Markov-chain model of the normal profile, the more likely the observed activities are anomalies resulting from cyber-attacks, and vice versa. Because many cyber-attacks require a series of related events to accomplish, an l^{th} order Markov-chain is used to improve attack detection performance by incorporating the continuous events.

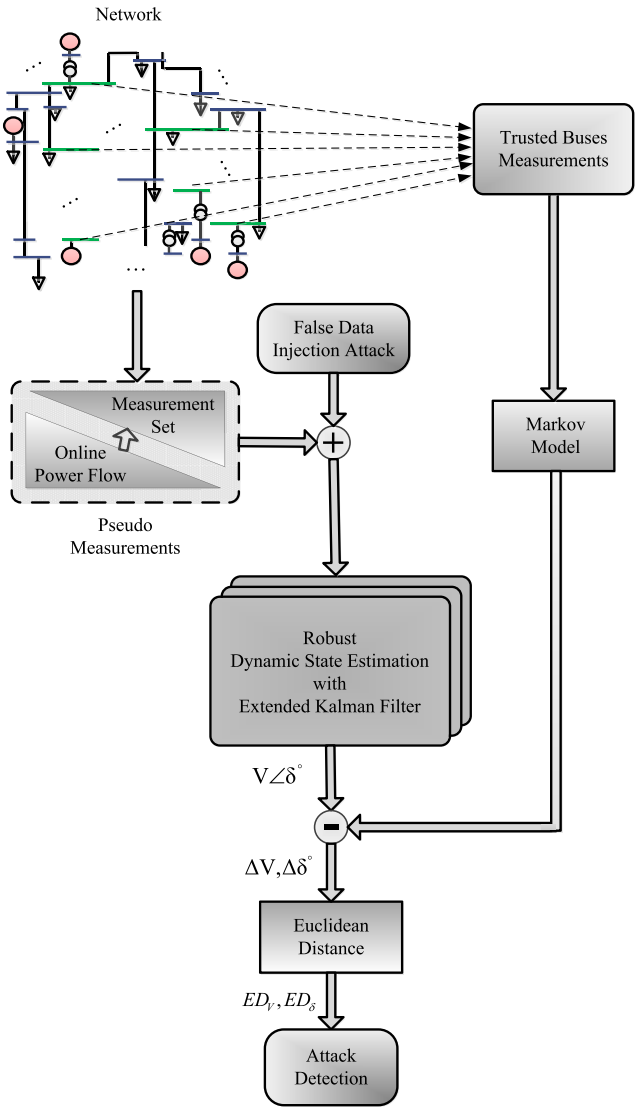

FIGURE 2. Overall block diagram of the proposed robust DSE method.

A. GPU ARCHITECTURE AND PROGRAMMING INTERFACE

The application of parallel processing in power system analysis is motivated by the desire for faster computation and the structure of the problems [37]. It should be noted that a GPU's architecture is different from that of a CPU. The CPU consists of several cores comprising of heavy-weight processing threads which are able to handle complex tasks with longer processing time; however, a GPU consists of thousands of cores that comprise massively parallel light-weight threads which can process simple operations with much lower latencies. GPU cores are more like arithmetic logic units rather than an actual processing cores. Thus, the structure of the problem and the solution algorithms should be redesigned in a way to breakdown into simple operations to take advantages of the GPU's architecture.

The programmer divides work into threads, threads into thread blocks, and thread blocks into grids. The actual

execution of a thread is performed by the abstracted CUDA cores which are a number of single precision floating point units.

B. IMPLEMENTATION OF ROBUST DSE AGAINST FDI ON GPU

The proposed robust DSE combines several aspects of parallelism to utilize the full capability of the GPUs as efficiently as possible. Initializations are done on the CPU. After that all of the data are transferred to the GPU for executing the robust DSE algorithm. In the first step, the traditional serial algorithm is converted into smaller independent tasks which results in task parallelism to be solved in parallel. All of the independent tasks in the three main steps of EKF are calculated in parallel to accelerate the algorithm. In the parameter identification step, $\mathbf{b}$ and $\boldsymbol{\gamma}$ do not rely on each other's result, so that they are calculated in parallel to accelerate the algorithm. In the state prediction stage $\tilde{\mathbf{x}}$, $\tilde{\boldsymbol{\rho}}$, $\mathbf{G}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are parallelizable. Finally in the state filtering step, $\hat{\mathbf{x}}$ and $\boldsymbol{\rho}$ can be calculated simultaneously.

In order to take advantage of the single instruction multiple data (SIMD)based architecture of the GPUs for basic computations data parallelism is used for matrix-vector and matrix-matrix products. The proposed cyber-attack detection algorithm is composed of matrix summation and inner products for a large set of data. For a large-scale system the size of the data is the main bottleneck for computational efficiency. These tasks can be broken down into simple operations to be assigned to an individual kernel to run in parallel. In addition to that all operations can be done in parallel for all data sets. By assigning each independent *for* loop to individual threads, the tasks can be executed in parallel by converting into a kernel.

In other word, underlying implementation of the robust DSE algorithm (vector updates, inner products, matrix vector products) takes advantage of the fine-grained parallelism using the CUDA parallel programming paradigm. These tasks can be assigned to an individual kernel to run in parallel. Each kernel is responsible for the calculation of that specific task. As the number of independent threads is a lot more than the CPU cores, this type of parallelization is not possible on the CPU. To illustrate how the CUDA works, consider a very simple matrix product on CPU and GPU.

Consider a function that takes two $N \times N$ matrices A and B and multiply them in a third matrix C. On the CPU, three *for* loops are used over all array elements as follows:

$$
\begin{aligned}
&for\ \ (i = 1\ :\ N)\\
&\quad for\ \ (j = 1\ :\ N)\\
&\quad\quad for\ \ (k = 1\ :\ N)\\
&\quad\quad\quad C[i][j] = A[i][k] * B[k][j] + C[i][j]\\
&\quad\quad end\\
&\quad end\\
&end
\end{aligned}
$$

The computation on the GPU can be performed by separating the outer loop from the inner calculations. First of all, enough memory space on device memory should be allocated for each matrix using *CudaMalloc* commands:

$CudaMalloc((void**)\&d_A, (sizeof(float) * N * N));$
$CudaMalloc((void**)\&d_B, (sizeof(float) * N * N));$
$CudaMalloc((void**)\&d_C, (sizeof(float) * N * N));$

d_A and d_B specifies the location of the matrix A and B in device memory. The next step is to transfer data to the GPU by executing the following command:

$cudaMemcpy(d_A, h_A, (sizeof(float) * N * N), \ldots);$
$cudaMemcpy(d_B, h_B, (sizeof(float) * N * N), \ldots);$

h_A and h_B specify the location of matrix on host memory. Arrays of the matrices will be stored in vector format in the GPU. The kernel code to perform the operation can be written as:

```
_global_ void MatMult(float*A, float*B, float*C, int N)
{
  float sum = 0;
  int id.x = blockIdx.x * blockDim.x + threadIdx.x;
  int id.y = blockIdx.y * blockDim.y + threadIdx.y;
  if(id.x < N || id.y < N)
  for(int i = 0; i < N; ++i)
  sum+ = A[id.y * N + i] + B[i * N + id.x];
  C[id.y * N + id.x] = sum;
  __syncthreads(); }
```

The global qualifier *__global__* specifies that the kernel is callable from the CPU and will be executed on the GPU. *id.x* and *id.y*, are defined to control the execution of the kernel. *blockDim.x* returns the number of threads in each block. Every thread in a block and every block in a grid has a unique index which is accessible through the *threadIdx* and *blockIdx*, respectively. The *__syncthreads*() call ensures that all threads are synchronized. To invoke this kernel from a CPU-based code we need to add a syntax as below:

$MatMult \lll grid, block \ggg (d_A, d_B, d_c, N);$

The grid dimensions and the block dimension in execution configuration ($\lll \ggg$) are defined by *grid* and *block*, respectively. At the end the result can be transferred to host memory using the *cudaMemcpy* command.

Sparse matrix-vector multiplication and sparse triangular solve is used for GPU implementation using cuSPARSE library [38]. Fig. 3 shows the flowchart of the proposed robust DSE method.

IV. CASE STUDIES

A. TEST SYSTEMS

To explore the efficiency of the GPU based robust DSE against FDI, large-scale systems were constructed for simulations. The IEEE 39-bus, IEEE 118-bus, and IEEE 2496-bus systems were implemented on the GPU for simulation studies. Case 3 (2496-bus) has build by duplicating and interconnecting the IEEE 39-bus system. To demonstrate the performance of the proposed method in terms of

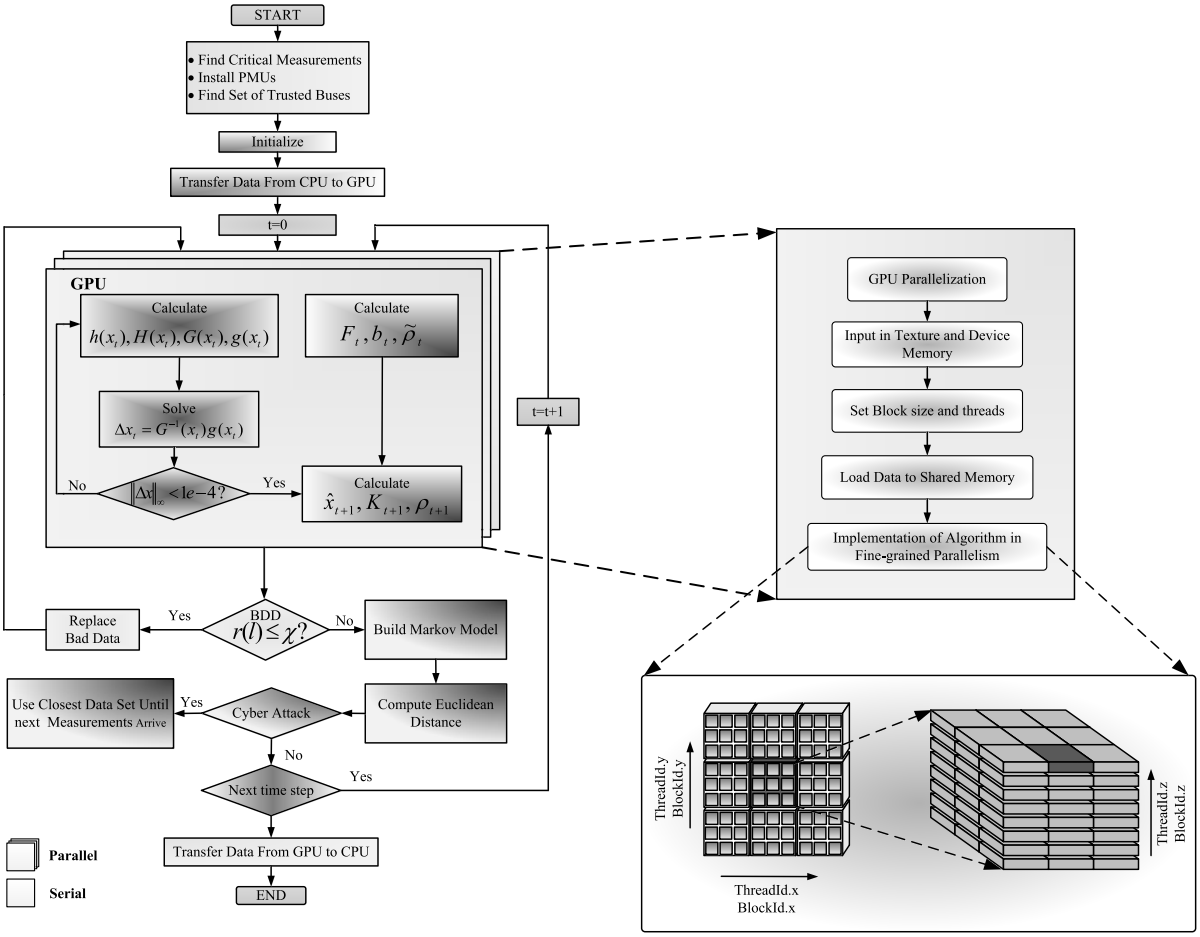

FIGURE 3. Flowchart of the proposed robust DSE method implemented on the GPUs.

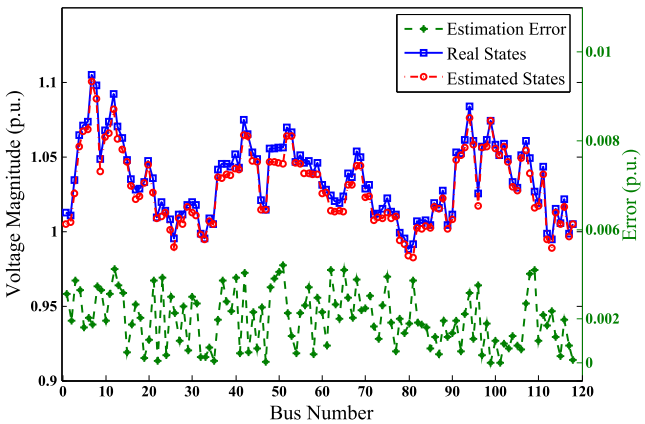

FIGURE 4. Voltage magnitude under normal operating condition.

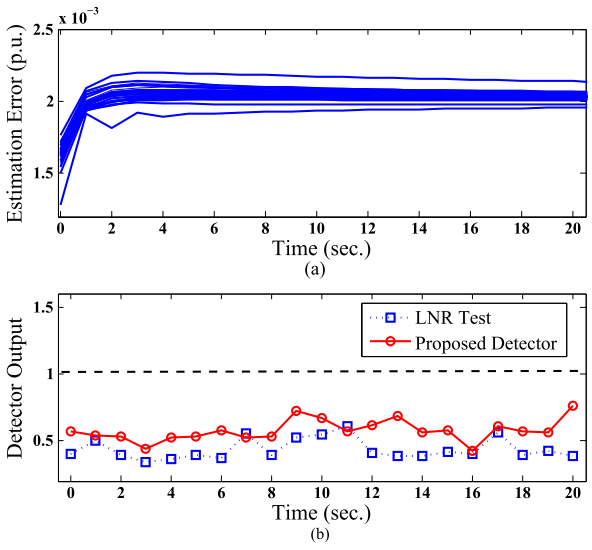

FIGURE 5. a) Estimation error, and b) detector output under normal operating condition.

speed-up, the above test systems were used to perform simulations on Tesla^{TM} S2050 GPU server from NVIDIA® with 4 Fermi GPUs, and 448 cores in each GPU. CUDA version 5.0 with compute capability 2.0 is used for programming. The CPU is the quad-core Intel® Xeon^{TM} E5-2620 with 2.0 GHz core clock and 32 GB memory, running 64-bit Windows 7® operating system. For accuracy analysis estimated states are verified using power flow analysis by PSS/E®.

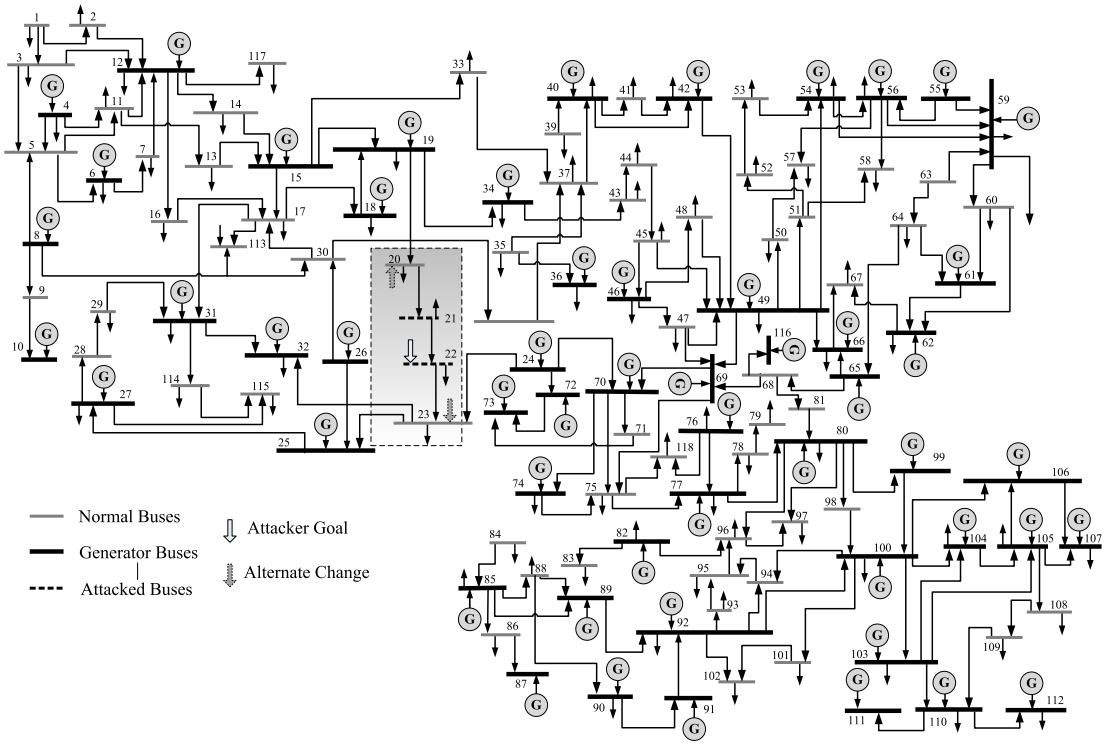

FIGURE 6. IEEE 118-bus test power system.

B. PERFORMANCE EVALUATION

In order to evaluate the accuracy of the proposed method, the results of the state estimation under normal operating conditions are plotted in Fig. 4 and Fig. 5. As there is no attack in the system, the results of state estimation are close enough to PSS/E® (real states). Fig. 5 (a) shows the estimation error for all 118 buses. It is also shown in Fig. 5 (b) that both LNR and proposed Markov Chain Detector (MCD) test were result in lower values than the threshold indicating that there was no attack in the system. The small differences compared to PSS/E® results are justifiable considering the fact that the order of block execution in each GPU grid is undefined in kernel definition. Therefore, it leads to slightly different results if different CUDA blocks perform calculations on overlapping portions of data. The same experiment is performed for all case studies, however only results of IEEE-118 bus system, are plotted for brevity. Details of the case studies along with average estimation error for voltage magnitude ($E_V^{Ave.}$) and phase angle ($E_\delta^{Ave.}$) are shown in Table 1.

C. ATTACK DETECTION ANALYSIS

In the second scenario, the proposed approach was evaluated for both random attack and intelligent FDI attack. For the cyber-attack, the goal of the attacker was to change the power injection at bus 22 by influencing the estimated values for the state variables at this bus in the IEEE 118-bus system shown in Fig. 6. For this attack to remain hidden other measurements have to be changed as well. In order to satisfy (13), (14), and (15), power injections at buses 20 and 23 need to be changed. Also, the power flows on the 21-22 and 22-23 connecting lines need to be adjusted as well which will change the power flow on line 20-21. As a result the attacker has to consider changing the estimated value for bus 21 on his attack modeling to keep the attack hidden. Fig. 7 and Fig. 8 show the behavior of the system under random attack. It is shown in the results that the estimation does not match with the measured values. The estimation error for all 118 buses are plotted in Fig. 8 (a). As cab be seen from Fig. 8 (a) during the first 8 seconds of the simulation estimation error is very small which ensure the accuracy of the estimator under normal operation condition. However, when the system is affected by a random fault the estimation error is suddenly increased. Since the attack is not intelligent, this increase in the estimation error will appear in the measurements residual and trigger both LNR and the proposed detector as shown in Fig. 8 (b).

Fig. 9 and Fig. 10 show the behavior of the LNR and proposed FDI test under the cyber-attack. It is clear from these results that the estimation does not match with the measured values. The same way as previous case study, Fig. 10 (a) shows the trend of changes in estimation error during normal operation condition and under cyber-attack. Once the system is under cyber-attack estimation error is suddenly changed, however, these changes easily bypass the LNR detector. Since the measurement residual remained the same, the LNR detector resulted in values below threshold and thus it was not able

TABLE 1. Summary of DSE results under FDI attack.

Case Case	No. of buses	No. of meas.	Jacobian (H)	Gain (G)	$E_V^{Ave.}$	$E_\delta^{Ave.}$	DO_{LNR}^{Max}	DO_{MCD}^{Max}	T_{CPU}	T_{GPU}	S_p
1	39	171	171×77	77×77	0.0027	0.059	0.6	1.95	0.31s	0.21s	1.47
2	118	609	609×235	235×235	0.0022	0.049	0.69	2.1	2.6s	0.55	4.7
3	2496	11553	11553×4991	4991×4991	0.0021	0.051	0.67	2.3	243s	31s	7.8

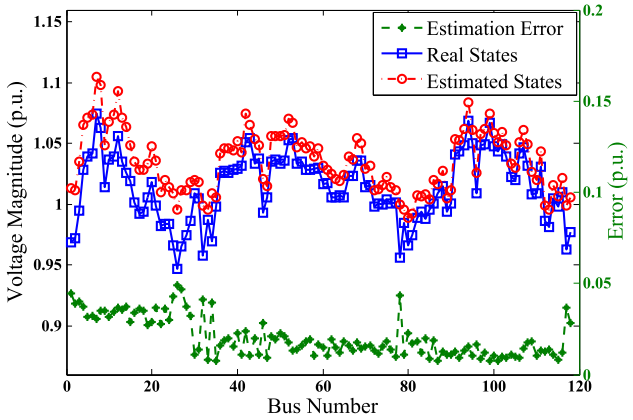

FIGURE 7. Voltage magnitude under random attack.

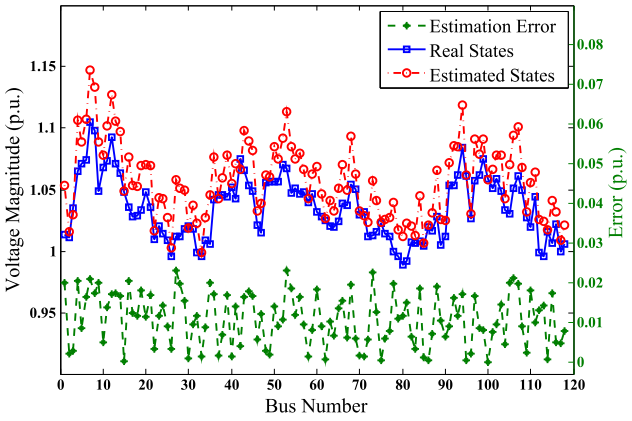

FIGURE 9. Voltage magnitude under FDI attack.

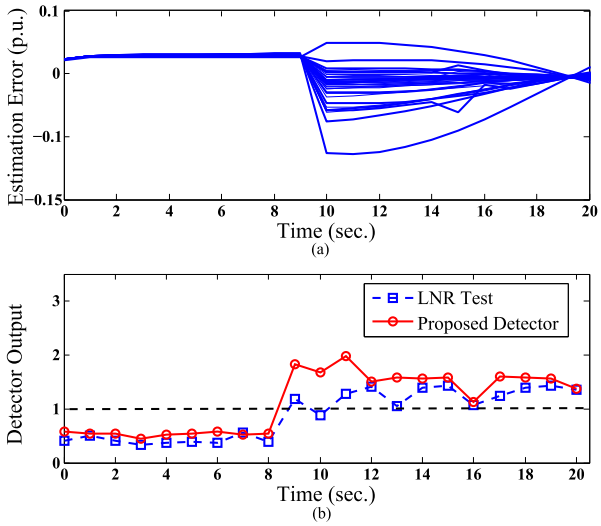

FIGURE 8. a) Estimation error, and b) detector output under random attack.

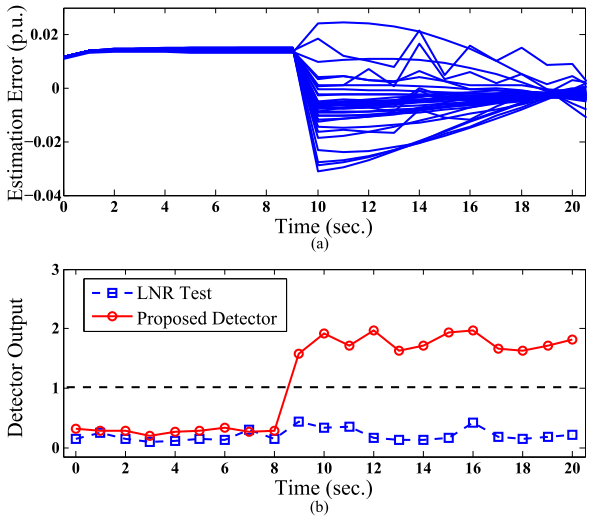

FIGURE 10. a) Estimation error, and b) detector output under FDI attack.

to detect the attack in the system as shown in Fig. 10 (b). The reason of failure in traditional bad data detection methods is that they mainly rely on residual in the measurements vector, however, as it is shown in Section II.C a cyber-attack does not leave any trace in measurements residual. In the same setup, the proposed Markov detector exceeds the threshold; hence, the FDI attack can be detected. The same experiment is performed for all case studies resulted in similar results, providing the effectiveness of the proposed approach.

The same analysis was done in all case studies. Details of the case studies along with average estimation error for voltage magnitude ($E_V^{Ave.}$) and phase angle ($E_\delta^{Ave.}$) after attack elimination are shown in Table 1. Maximum detector output for LNR test (DO_{LNR}^{Max}) and proposed Markov chain detector (DO_{MCD}^{Max}) under FDI attack are also reported in Table 1. The maximum detector output shows that in all comparative test simulations the two traditional method resulted in an output less that the specified threshold which will not trigger the alarm. In general, any type of FDI attack in measurement set, transition line or system topology results in the same changes in the network with slight modification. Therefore, a detector that can identify above attack will be able to detect similar cyber-attacks that are coming from different sources.

TABLE 2. GPU resource occupancy.

Case	Occupancy	No. of cores	Max. no. of grids
1	51.4%	230	7
2	63.2%	283	53
3	84.8%	379	9384

D. COMPUTATIONAL EFFICIENCY AND RESOURCE DISTRIBUTION

In order to certify the efficiency of the proposed GPU-based robust DSE the speed-up ratio is defined as $S_p = T_{CPU}/T_{GPU}$, where T_{CPU} and T_{GPU} are the execution times of the serial algorithm running only on CPU and parallel algorithms on the GPU, respectively. $E_V^{Ave.}$ and $E_\delta^{Ave.}$ are errors of estimation for voltage magnitude and phase angle, respectively. As the results reported in Table 1 show, the advantage of utilizing GPU for parallelization is significant when the size of the system increases. It is obvious that execution time on CPU follows a high order complexity as the system size grows. However, the execution time of the robust estimator on GPU increases almost linearly with respect to the system size as a result of fine grained parallelism on GPU. Therefore it is expected to see higher speed-up for larger case studies. Based on Gustafson's law [39], the maximum achievable speed-up by parallelization is proportional to the number of CPU cores in the system. Unlike the CPU, there is no fixed law to predict the maximum achievable speed-up using GPU. As it is obvious from results, more cores increases the processing power and throughput of the GPU and results in significantly faster algorithm. So it is expected to see even better performance using GPU for larger case studies which make it suitable for real implementation. The resource distribution from CUDA on the Tesla S2050 GPU server is shown in Table 2. As can be seen from the results, the number of cores increase dynamically as the size of the system increases. Distribution of threads, blocks and memory varies in different kernels. Typically, the number of thread per block is a constant number which was 128 in our case studies. The number of blocks per grid is different based on the problem size in each case study. The maximum number of blocks per grid in each dimension was 16. The maximum number of grids for each Case study is reported in Table 2.

V. CONCLUSION

In this paper, a robust parallel dynamic state estimation approach utilizing graphic processing units and extended Kalman filter was presented. The proposed approach can detect false data injection attack using trusted set of measurements which were secured using optimized PMU installation. Considering the stochastic nature of the power system, using Markov chain theory and history of the system's dynamic behaviour a Markov model was proposed to check the accuracy of the estimation results using the Euclidean distance metric. Simulation results verify the accuracy of the proposed method both under normal operating condition and under false data injection attack. It should be considered that selecting a different threshold will not change the fact that proposed method can detect FDI attack which bypass the traditional BDD techniques. Large case studies along with parallel implementation on GPUs shows the speed and applicability of the proposed approach for real-time large-scale power systems operation. The primary benefit to control room operations is the ability to process large amounts of data and provide useful information on a much faster time scale and fidelity. For future work, further analysis will be included to identify the type of attack (e.g. low frequency, rate of change of frequency, damping rate). Also historical data selection algorithm will be improved using optimization technique to generate a priority list.

REFERENCES

[1] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "Security evaluation of the cyber networks under advanced persistent threats," *IEEE Access*, vol. 5, pp. 20111–20123, 2017.
[2] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, Chicago, IL, USA, Nov. 2009, pp. 21–32.
[3] Z. Huang, C. Wang, T. Zhu, and A. Nayak, "Cascading failures in smart grid: Joint effect of load propagation and interdependence," *IEEE Access*, vol. 3, pp. 2520–2530, Dec. 2015.
[4] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Trans. Smart Grids*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.
[5] O. Vuković, G. Dán, "Security of fully distributed power system state estimation: Detection and mitigation of data integrity attacks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1500–1508, Jul. 2014.
[6] E. Ghahremani and I. Kamwa, "Dynamic state estimation in power system by applying the extended Kalman filter with unknown inputs to phasor measurements," *Environ. Sci.*, vol. 11, pp. 655–661, 2011.
[7] Y. Huang, M. Esmalifalak, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, "Adaptive quickest estimation algorithm for smart grid network topology error," *IEEE Syst. J.*, vol. 8, no. 2, pp. 430–440, Jun. 2014.
[8] A. Abur and A. G. Exposito, "Bad data identification when using ampere measurements," *IEEE Trans. Power Syst.*, vol. 12, no. 2, pp. 831–836, May 1997.
[9] E. N. Asada, A. V. Garcia, and R. Romero, "Identifying multiple interacting bad data in power system state estimation," in *Proc. IEEE PES*, Jun. 2005, pp. 571–577, doi: 10.1109/JSYST.2014.2341597.
[10] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. Sastry, "Cyber security analysis of state estimators in electric power systems," in *Proc. ACM Conf. Decision Control*, Dec. 2010, pp. 5991–5998.
[11] G. Hug and J. A. Giampapa, "Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1362–1370, Sep. 2012.
[12] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 650–662, Jun. 2015.
[13] W. Ao, Y. Song, and C. Wen, "Adaptive cyber-physical system attack detection and reconstruction with application to power systems," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1458–1468, Aug. 2016.
[14] S. Ntalampiras, "Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 104–111, Feb. 2015.
[15] R. G. Kavasseri, Y. Cui, and N. R. Chaudhuri, "A supervisory approach towards cyber-secure generator protection," in *Proc. Joint Workshop Cyber-Phys. Secur. Resilience Smart Grids*, 2016, pp. 1–6.
[16] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, Sep. 2017.

[17] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 326–333, Jun. 2011.
[18] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.
[19] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, "Real-time detection of false data injection in smart grid networks: An adaptive CUSUM method and analysis," *IEEE Syst. J.*, vol. 10, no. 2, pp. 532–543, Jun. 2016.
[20] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 659–666, Dec. 2011.
[21] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 160–169, Mar. 2013.
[22] D. T. Hoang, P. Wang, D. Niyato, and E. Hossain, "Charging and discharging of plug-in electric vehicles (PEVs) in vehicle-to-grid (V2G) systems: A cyber insurance-based model," *IEEE Access*, vol. 5, pp. 732–754, 2017.
[23] S. Wang, W. Ren, and U. M. Al-Saggaf, "Effects of switching network topologies on stealthy false data injection attacks against state estimation in power networks," *IEEE Syst. J.* , vol. 11, no. 4, pp. 2640–2651, Dec. 2017.
[24] A. Ashok, P. Wang, M. Brown, and M. Govindarasu, "Experimental evaluation of cyber attacks on automatic generation control using a CPS security testbed," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2015, pp. 1–5.
[25] X. Luo, J. Li, Z. Jiang, and X. Guan, "Complete observation against attack vulnerability for cyber-physical systems with application to power grids," in *Proc. 5th Int. Conf. Electr. Utility Deregulation Restruct. Power Technol. (DRPT)*, Nov. 2015, pp. 962–967.
[26] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan, and U. Adhikari, "Modeling cyber-physical vulnerability of the smart grid with incomplete information," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 235–244, Mar. 2013.
[27] S. Zonouz, K. M. Rogers, R. Berthier, R. B. Bobba, W. H. Sanders, and T. J. Overbye, "SCPSE: Security-oriented cyber-physical state estimation for power grid critical infrastructures," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1790–1799, Dec. 2012.
[28] S. Cui, Z. Han, S. Kar, T. T. Kim, H. Poor, and A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 106–115, Sep. 2012.
[29] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du, "Achieving efficient detection against false data injection attacks in smart grid," *IEEE Access*, vol. 5, pp. 13787–13798, Jul. 2017.
[30] M. V. O. De Assis, A. H. Hamamoto, T. Abrão, and M. L. Proença, "A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks," *IEEE Access*, vol. 5, pp. 9485–9496, May 2017.
[31] R. Khan, K. Mclaughlin, D. Laverty, and S. Sezer, "Design and implementation of security gateway for synchrophasor based real-time control and monitoring in smart grid," *IEEE Access*, vol. 5, pp. 11626–11644, Jun. 2017.
[32] S. P. Meyn, R. L. Tweedie, *Markov Chain and Stochastic Stability*. New York, NY, USA: Springer-Verlag, 2005.
[33] Z.-H. Liu, X.-H. Li, L.-H. Wu, S.-W. Zhou, and K. Liu, "GPU-accelerated parallel coevolutionary algorithm for parameters identification and temperature monitoring in permanent magnet synchronous machines," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1220–1230, Oct. 2015.
[34] V. Roberge, M. Tarbouchi, and G. Labonté, "Parallel algorithm on graphics processing unit for harmonic minimization in multilevel inverters," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 700–707, Jun. 2015.
[35] S. Makridakis and S. C. Wheelwright, *Forecasting Methods and Applications*. Hoboken, NJ, USA: Wiley, 1978.
[36] A. Abur and A. Gómez-Expósito, *Power System State Estimation Theory and Implementation*. New York, NY, USA: Marcel Dekker, 2004.
[37] *NVIDIA Tesla: A Unified Graphics and Computing Architecture, NVIDIA CUDA C Programming Guide 4.0.*, NVIDIA, Santa Clara, CA, USA, 2013.
[38] *cuSPARSE Library, NVIDIA Developer*, NVIDIA, Santa Clara, CA, USA, Feb. 2013.
[39] J. L. Gustafson, "Reevaluating Amdahl's law," *Commun. ACM*, vol. 31, no. 5, pp. 532–533, Jan. 1988.

HADIS KARIMIPOUR received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Alberta, in 2016. She was a Postdoctoral Fellow at the University of Calgary, where she was involved in cyber-security of the smart grids. She is currently an Assistant Professor at the Engineering Systems and Computing Group, School of Engineering, University of Guelph, Guelph, ON, Canada. Her research interests include large-scale power system state estimation, cyber-security of the smart grids, demand side management, and parallel and distributed computing.

VENKATA DINAVAHI received the Ph.D. degree from the University of Toronto in 2000. He is currently a Professor at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include the real-time simulation of power systems, large-scale system simulation, and parallel and distributed computing.