

Learning the Unobservable: High-Resolution State Estimation via Deep Learning

Kursat Rasim Mestav and Lang Tong

Abstract—The problem of achieving the fast-timescale state estimation for a power system with limited deployment of phasor measurement units is considered. A deep neural-network architecture that integrates bad-data detection, data cleansing, and the minimum mean squared error state estimation is developed. It includes a universal bad-data detection and a Bayesian state estimation subnetworks. A novel universal bad-data detection technique is proposed that requires no knowledge about data distributions under regular and irregular operating conditions. The subnetwork for universal bad-data detection consists of an inverse generative model and a coincidence test. It is implemented through the training of a generative adversary network and an auto-encoder using slow-timescale historical data. The Bayesian state estimation subnetwork is trained through a generative adversary network with embedded physical models of the power system. Comparing with the conventional weighted least squares approach to state estimation, the proposed minimum mean-squared error state estimator does not require observability. Simulations demonstrate orders of magnitude improvement in estimation accuracy and online computation costs over the state-of-the-art solutions.

Index Terms—Deep learning, neural networks, regression learning, state estimation, phasor measurement unit (PMU), generative adversary networks, anomaly detection, bad-data detection, and Bayesian inference.

I. INTRODUCTION

We consider the problem of state estimation for a power system that has limited phasor measurement units (PMUs). PMUs provide synchronized measurements of the system state at a rate two orders of magnitude faster than that of the traditional supervisory control and data acquisition (SCADA) system. PMU technology offers the potential of wide-area situational awareness, (near) real-time control, fast timescale security monitoring, and enhanced network resilience.

Unfortunately, despite over billion-dollar investments by governments and industries worldwide, the cumulative deployment of PMU remains limited, sufficient only for some high voltage networks and vastly inadequate for most transmission systems. The state-of-the-art is that most power systems are not PMU-observable, *i.e.*, PMU measurements alone are insufficient for state estimation at the PMU timescale.

There have been attempts to combine traditional SCADA and PMU measurements to achieve PMU-timescale state estimation. Such attempts have not been successful in providing accurate fast-timescale monitoring because there is a

significant model mismatch for data from the slow and fast timescales.

A long-standing challenge in state estimation is the ability to handle bad and malicious data; the latter can be constructed to avoid detection [1]. Existing techniques are ineffective in two critical aspects. First is the ability of state estimation techniques to deal with multiple (possibly interacting) bad. Because bad data do not necessarily follow a specific probability distribution consistently, and few (if any) samples of bad data that can be used to characterize the behavior of data anomaly, one faces a seemingly intractable statistical inference problem. Second is the computation cost of bad data detection. Conventional approaches often iterate between state estimation using computationally expensive weighted least squares (WLS) techniques. The cost of such iterative schemes can be prohibitive for large systems.

A. Summary of Results and Contributions

We present a deep learning approach to high-fidelity and high-resolution state estimation for power systems that are PMU-unobservable. By high-resolution state estimation, we mean that the states are estimated at the PMU-timescale. To this end, we adopt a Bayesian inference approach that overcomes the unobservability barrier faced by conventional WLS methods, enabling the use of limited PMUs for system-wide state estimation at the fast timescale. By high-fidelity state estimation, on the other hand, we mean the ability for the state estimation algorithm to mitigate bad and malicious data.

The main contribution of this work is threefold. First, we propose an integrated deep neural-network architecture for joint bad-data detection, bad-data cleansing, and state estimation. Second, we develop a generative adversary network (GAN) learning framework for Bayesian state estimation and bad data detection. Finally, we propose a novel deep learning approach to *universal bad-data detection (UBD)*. By universal bad-data detection, we mean specifically that the data distributions in either regular or abnormal conditions are unknown. We assume some historical data are available under the regular operating condition, and no abnormal data samples are available.

B. Related Work

For PMU-unobservable systems, a standard approach to state estimation is to combine PMU and SCADA measurements to form an augmented WLS problem [2]–[4]. There is, however, a model mismatch between the two

Kursat Rasim Mestav and Lang Tong are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, 14850 USA e-mail: {krm264, lt35}@cornell.edu.

This work is supported in part by the National Science Foundation under Award 1809830, Award 1932501, and the Power Systems and Engineering Research Center (PSERC).

types of measurements at different timescales [4]–[6]. The mismatched modeling error can be quite significant when the system states change rapidly. Indeed, even when both types of measurements are noiseless, the augmented WLS estimator cannot capture the fast varying system states.

Another line of approaches is to generate PMU pseudo measurements. One of the earliest such techniques is based on SCADA injection measurements [7]. From an estimation theoretic perspective, generating pseudo measurements can be viewed as predicting missing PMU measurements. Thus the pseudo-measurement techniques are part of the so-called forecasting-aided state estimation [8]. To this end, machine learning techniques in load forecasting can be tailored to produce pseudo measurements [9]. The issue of model mismatch remains for these techniques.

There is also a third long and active line of approaches based on dynamic state estimation proposed in the late 60s by Deb and Larson [10] under a state-space model of power systems. Although obtaining a dynamic model may be complicated, the fact that one well-placed sensor can make a linear system observable makes dynamic state estimation particularly attractive [11]. This is an extensive and still growing literature on this topic [12].

Bayesian approaches to state estimation are far less explored even though the idea was already proposed in the seminal work of Schweppe [13]. Bayesian state estimation generally requires the computation of the conditional statistics of the state variables. One approach to calculate conditional statistics is based on a graphical model of the distribution system from which belief propagation techniques are used to generate state estimates [14]. Such techniques require a dependency graph of the system states and explicit forms of probability distributions. Another approach is based on a linear approximation of the AC power flow [15].

Bad-data detection and identification has been studied extensively [16]. Conventional methods are *post-estimation* techniques by first disregarding the presence of bad data. The resulting state estimate is then used to compute the residue error between actual measurements and ones predicted by the state estimates. The presence of bad data is declared if the residue error exceeds a certain threshold. For such techniques, system observability is a prerequisite. The effects of bad data in the initial state estimate cannot be ignored, however, which often leads to misidentification of bad data and the removal of good data.

In contrast to post-estimation bad-data detection, the method proposed in this paper belongs to the less well-known class of *pre-estimation* detection and filtering techniques. Several such techniques [17]–[19] are based on exploiting a dynamic model to predict the current measurement using past measurements, from which the prediction error becomes test statistics for bad-data detection.

There is an expanding literature on the use of deep learning for anomaly detection [20]. Among existing approaches in the literature, the work in [21] has an architectural connection with our approach; both involving GAN learning and auto-encoder. The main difference between the technique

proposed here and that in [21] is the test used for detection: a coincidence test used in this work, the mismatch between the input and output of the auto-encoder is used in [21]. Our detector is consistent in the sense that the detection error probability approaches to zero; the detector used in [21], in contrast, does not provide such a guarantee.

The proposed technique in this paper builds on to our work on deep learning approach to state estimation for unobservable distribution systems [22]. Main differences between the technique proposed in this work and that in [22] are the learning of generative model and the bad-data detection techniques.

II. SYSTEM MODEL AND BAYESIAN STATE ESTIMATION

The power system state x_t at time t is a column vector consisting of real and imaginary parts of the voltage phasors. A PMU at a particular bus measures directly the voltage and current phasors on the bus. Therefore, the PMU measurement vector z_t and the system state x_t satisfy a linear model

$$z_t = Hx_t + w_t, \quad (1)$$

where H is the observation matrix defined by the network topology, the location of PMUs, and network parameters such as the network admittance matrix, and w_t the measurement noise.

The frequentist approach to state estimation is to model x_t as deterministic. A standard state estimation technique is the *weighted least squares (WLS)*, which minimizes the difference between the observed measurements and the predicted measurement by the system state. Under the PMU measurement model, the WLS state estimator is given by

$$\hat{x}_t^{\text{WLS}} = \arg \min_x \|z_t - Hx\|.$$

When there are not enough PMUs installed, or some PMUs are faulty, matrix H becomes column-rank deficient and the system *PMU-unobservable*, or unobservable for short. For such a system, even if the *model prediction error* $\mathcal{E}_{z_t} = z_t - Hx_t$ is small, the *estimation error* $\mathcal{E}_{x_t} = \hat{x}_t - x_t$ can be arbitrarily large. Thus, the WLS approach applies to systems that are observable, or additional constraints on the state (such as sparsity) are necessary.

We adopt the Bayesian approach that assumes the system state x_t is a *random vector*, jointly distributed with the observation vector z_t . Instead of minimizing the prediction error used by WLS, the Bayesian approach minimizes the estimation error directly. Using the l_2 -norm* to measure the size of error, the Bayesian estimator minimizes the mean-squared error. The *minimum mean-squared error (MMSE) estimator* is given by the deceptively simple solution in the form of the *conditional mean*:

$$\min_{\hat{x}(\cdot)} \mathbb{E}(\|x - \hat{x}(z_t)\|_2^2) \rightarrow \hat{x}_t^{\text{MMSE}} = \mathbb{E}(x_t|z_t) \quad (2)$$

A major advantage of the Bayesian formulation is that system observability is no longer required. The difficulty

*Other norms such that l_1 - norm or the probability of error can also be used.

of implementing the MMSE estimator, however, is that the underlying joint probability distribution governing x_t and z_t is often unknown. Even when such a functional form of the joint distribution is known, the computation of conditional mean may be intractable.

III. BAYESIAN STATE ESTIMATION VIA DEEP LEARNING

A. Deep learning architecture for Bayesian state estimation

The architecture of the proposed deep learning solution to Bayesian state estimation with anomaly data detection and data cleansing is shown in Fig. 1.

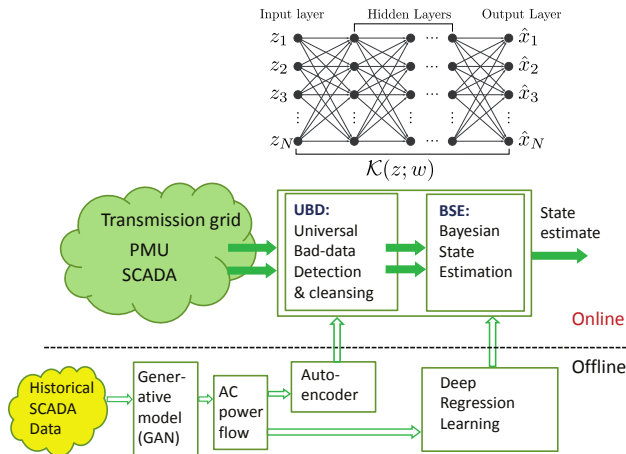


Fig. 1: A deep learning architecture.

For online state estimation, PMU and (possibly SCADA) measurements z_t are input of a deep neural network, denoted by $\mathcal{K}(\cdot, w)$, that produces the state estimate $\hat{x}_t = \mathcal{K}(z_t, w)$, where w is the weight matrix of the deep neural network. When the neural network is sufficiently large and its weights properly trained, the neural network $\mathcal{K}(\cdot, w)$ can approximate arbitrarily well the MMSE estimator.

The lower layers of the network is a *data pre-processing subnetwork* that implements a *universal bad-data detection* (UBD) and data cleansing. The upper layers is the *Bayesian state estimation subnetwork* (BSE) that implements the MMSE state estimator. The two subnetworks are trained separately.

The deep neural network used for online state estimation is trained offline. A generative model of power injections is learned using historical SCADA data via a generative adversary network (GAN). The learned generative model can then be used to generate a set of training samples $\mathcal{S} = \{(x_t, z_t)\}$. Two separate deep neural networks are trained using \mathcal{S} : one is an auto-encoder for UBD, the other a regression network for BSE.

The training of a deep neural network for BSE is standard. An empirical risk minimization is used on the training set \mathcal{S} , as in [22]. We find it useful to set the input layer as the linear MMSE estimator as a way to precondition the input.

The auto-encoder for UBD consists of a bank of scalar auto-encoders, one for each measurement variable. The auto-encoder is a cascade of an encoder and a decoder; the encoder maps the input to a uniformly distributed latent variable, and the decoder that maps the latent variable back to the input. The decoder plays the role of a generative model G that maps a uniformly distributed random variable to the distribution of the measurement. The encoder is the inverse generative model G^{-1} that maps the distribution of the measurement to the uniformly distributed latent variable.

In training the auto-encoder, the decoder is trained using the z_t -samples of \mathcal{S} using the GAN technique. The encoder is trained using the samples generated by the decoder via an empirical risk minimization.

The use of uniform distribution for the latent variable is essential, as discussed in the next section.

B. Universal bad data detection and data cleansing

We present a novel technique for UBD—the universal bad-data detection—under a binary hypothesis testing framework. Once bad data are detected, they are replaced by the (unconditional) mean values.

For simplicity, consider a scalar measurement z under the null hypothesis \mathcal{H}_0 for regular operating conditions and the alternative \mathcal{H}_1 for anomaly:

$$\begin{aligned} \mathcal{H}_0 : z \sim f_0 \quad \text{vs.} \quad \mathcal{H}_1 : z \sim f_1 \in \mathcal{F}_\epsilon \\ \mathcal{F}_\epsilon := \{f, \|f - f_0\| \geq \epsilon\} \end{aligned} \quad (3)$$

where f_0 and f_1 are probability distributions under the two hypotheses, and \mathcal{F}_ϵ is the set of probability distributions that are ϵ distance (divergence) away from f_0 . The null hypothesis \mathcal{H}_0 is a simple and the alternative \mathcal{H}_1 composite.

We assume that neither f_0 nor f_1 is known. Under \mathcal{H}_0 , however, we have some training samples distributed in f_0 . No training samples for the abnormal data under \mathcal{H}_1 .

1) *Coincidence test*: UBD is inspired by the coincidence test [23] that tests the (discrete) uniform distribution with alphabet size M under \mathcal{H}_0 against an arbitrary non-uniform (discrete) distribution of the same alphabet size, ϵ distance away from the uniform.

Suppose that a vector \tilde{u} of N i.i.d. samples are collected. By coincidence it means that, among N samples in \tilde{z} , there are at least two have the same value. The obvious connection is the birthday coincidence when there are two people among N having the same birthday. When the birthdays are uniformly distributed and $N = 23$, there is a roughly 50-50 chance that there are two people have the same birthday. When the birthday distribution is not uniform, the chance of the birthday coincidence is higher [24]. It is this extreme property of the uniform distribution that serves the basis of testing uniform against non-uniform distributions.

Paninski proposed in [23] a consistent test based on the the number of samples that do not have coincidence, *i.e.*, the number of samples that have unique values. Let $K_1(\tilde{u})$ be such a number in an N sample realization $\tilde{u} = (\tilde{u}_1, \dots, \tilde{u}_N)$.

It is shown that the threshold test on $K_1(\tilde{u})$

$$K_1(\tilde{u}) \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\geq}} T_\epsilon := \frac{N^2\epsilon^2}{2M} - N\left(1 - \frac{1}{M}\right)^{N-1}$$

achieves consistency, *i.e.*, detection error goes to zero so long as $N^2\epsilon^4/M \rightarrow \infty$.

2) *From uniform to general distribution*: Key steps that allow us to adapt the coincidence test to achieve UBD is shown in Fig 2. The first step is to transform the observation random variable z_t with general distribution f_0 under \mathcal{H}_0 to a uniformly distributed random variable $u \sim \mathcal{U}(0, 1)$. This is accomplished by the inverse generative model G^{-1} obtained through the training of the auto-encoder described in Sec III-A where G is the function that maps a uniform distributed random variable u to f_0 distributed z .

The second step is the analog-to-digital (A/D) conversion that transforms the uniformly distributed continuous random variable u to a uniformly distributed discrete random variable $\tilde{u} \sim \mathcal{U}(M)$ with M alphabets. In other words, putting the N variables into M equally sized bins. The rest of steps follow the coincidence test.

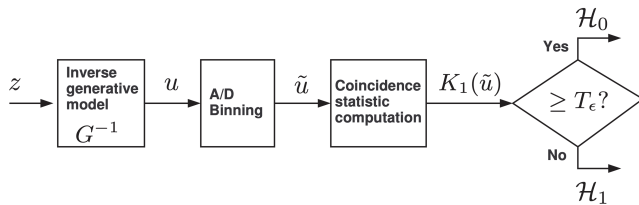


Fig. 2: Universal bad data detection.

For practical implementations, M and N are parameters. If we assume that the bad/malicious data do not change distribution within one tenth of a second, we collect say, $N \approx 10$ samples and putting them in $M \approx 100$ bins, and compute $K_1(\tilde{u})$ —the number of bins with a single sample. Satisfactory performance was achieved in our simulations. The threshold parameter T_ϵ is another parameter that trades off the false-alarm and miss-detection probabilities.

IV. SIMULATIONS RESULTS AND DISCUSSIONS

A. Simulation Settings

a) *Systems simulated*: The simulations were performed on the IEEE-118 Bus Power Flow Test Case [25]. The PMU timescale load distribution of the buses in the EPFL Smart Grid Project dataset [26] was used as the load distribution for the buses. In the simulated system, the load distribution of each bus was scaled by a different mean and a variance to make each bus has a unique load distribution. We limited our data set using a specific time interval, from 5 pm to 6 pm for the days in May 2018. There should be separate networks trained for other hours and seasons.

Using the load distributions, we generated 10,000 training samples and 10,000 test samples. The power flow equations

are solved using MATPOWER toolbox [27] to obtain the states and the measurements from the power injections for the training and the test sets.

Two types of measurement devices were assumed: (i) PMUs were assumed to be on some of the buses to measure the complex voltage at the bus and the currents on branches to the neighbor buses. (ii) SCADA meters were assumed to be at all buses to measure complex power to/from the transmission grid. The additive measurement noise without bad data was assumed to be independent and identically distributed Gaussian with zero mean and variance set at 1% of the average net consumption value.

b) *Performance measure*: The performance of the tested algorithms was measured by the per-node average squared error (ASE) defined by,

$$ASE = \frac{1}{MN} \sum_k \|\hat{x}[k] - x[k]\|^2, \quad (4)$$

where M is the number of Monte Carlo runs, k the index of the Monte Carlo run, N the number of nodes, $\hat{x}[k]$ and $x[k]$ the estimated and the state vectors, respectively.

c) *Distribution Learning via generative deep learning*: We used a generative network with 2 hidden layers and 100 neurons at each layer. Batch normalization and dropout with rate 0.2 were used in hidden layers. Leaky-Rectified Linear Units (ReLU) at hidden layers and a linear activation function at the final layer were used as the activation functions. For the discriminative network, we used two hidden layers with 30 neurons. Leaky-ReLU at hidden layers and a linear activation function at the final layer are used as the activation functions. Adam optimization [28] algorithm with mini-batches was used as the optimizer. Wasserstein GAN with clipping parameter $[-0.01, 0.01]$ [29] was implemented with uniform noise.

We standardized all the power injection measurements at different buses by subtracting the sample mean and dividing by the sample standard variation. We trained the generative model to learn the power injection distribution from SCADA measurements. After the training, we used DvoretzkyKiefer-Wolfowitz (DKW) inequality with 95 % confidence interval. We plotted the cumulative distribution function (CDF) of the samples generated by the generator and the confidence interval we obtained from DKW. Fig. 4 shows the generator's output is in the confidence interval.

After training the generator, we generated power injection samples using it and renormalized them for each bus. We solved the power flow equations to obtain the PMU measurements and the states for each case. We separated the 100,000 training samples we obtained into 70,000 samples to train the deep learning network and 30,000 samples as the validation set.

d) *Neural network specification and training*: We trained a deep neural network using the PMU measurements as input and the state values as output. ReLU activation function was used for neurons in the hidden layers and linear activation functions in the output layer. The Adam algorithm was used to train the neural network with mini-batches of

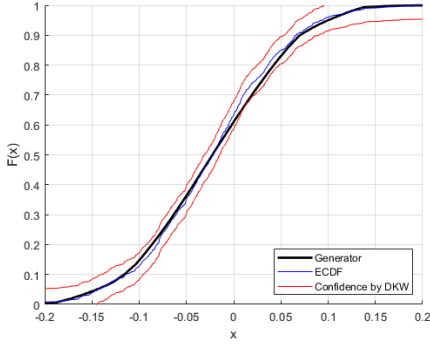


Fig. 3: The output of the generator (black), empirical CDF (blue) and the confidence interval given by the DKW inequality.

60 samples. Early stopping was applied by monitoring the error on the validation set. To have a better regularization, batch normalization and dropout with 0.3 dropping rate are used at all layers. The training algorithm was implemented in Python 3.7.3 using Keras v2.2.4 with Tensorflow v 1.14 as the backend [30].

e) Comparison of Performances: For this network, 32 optimally placed PMUs make the system observable. We repeated the experiment with 8, 14, 20 and 26 PMUs which are guaranteed to be unobservable. We tuned the parameters of the neural network separately for each case. We observed using 3 to 5 hidden layers and 100-200 neurons per layer after the LMMSE layer gives the best performance.

The Fig. 4 shows ASE comparison for the 118-bus system among the augmented WLS, interpolated WLS [7] and the Bayesian neural network solutions. It is demonstrated that for highly unobservable systems, our method performed significantly better than conventional WLS methods.

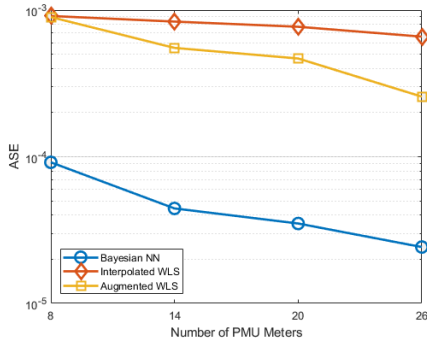


Fig. 4: The comparison of algorithms on 4 test cases.

f) Bad Data Detection and Data Cleansing: We tested bad data detection methods in two scenarios. In the first one, we used a Gaussian noise on some of the measurements with a much bigger variance, Fig. 5 (a). In the second one, we used a Gaussian noise on some of the measurements with a big non-zero mean, Fig. 5 (b). On both scenarios, we used 8 PMUs in the system. We compared the performance of universal bad data detection (UBD) algorithm with, $J(x)$

test and reconstruction error of the autoencoder test (RET). Since $WLS - J(x)$, requires observability to work, we used the augmented WLS solution to achieve observability using unsynchronized SCADA measurements.

We used $N = 50$ samples to detect bad data. We used the universal bad data detection method with $m=2500$ bins. We calculated the false positive rate (FPR) and the true positive rate (TPR) by varying the thresholds of each algorithm. In Fig. 5 (c) and (d) we plotted the receiver operating characteristic (ROC) curves for scenario 1 and 2 respectively.

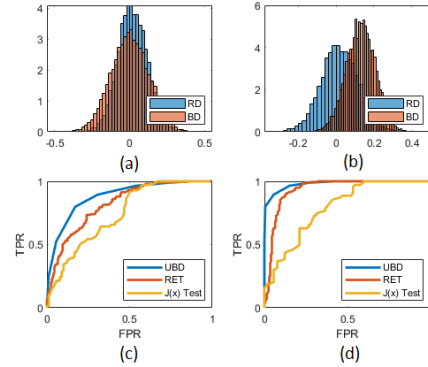


Fig. 5: (a,b): Histograms of regular and bad data, RD: Regular Data, BD: Bad Data. (c,d): ROC curves of bad data detection methods.

In power system applications it is often desired a small FPR. In the next simulation, we fixed the FPR at 0.05. We increased the noise level for both scenarios and calculated the TPR of three algorithms. Fig. 6 shows UBD has consistently better performance under different levels of noise than the other algorithms.

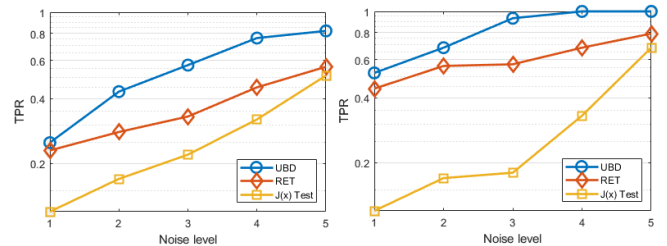


Fig. 6: False positive rates of bad data detection methods.

Finally, we simulated the complete algorithm combining the universal bad data detection and the state estimation on the first bad data scenario. We simulated three cases: (i) Clean data (with only measurement noise): We did not use any bad data on the measurements, (ii) Bad data are present: We used bad data on the measurements with 0.3 probability, but we did not clear any of them and (iii) With bad data and bad data cleansing: We detected the bad data using UBD and cleared it. Fig. 7 shows the performance with bad data detection and clearance improves the performance of state estimation when bad data occurs. Moreover, even with the

bad data, the Bayesian neural network estimation has a better performance than the Augmented WLS solution.

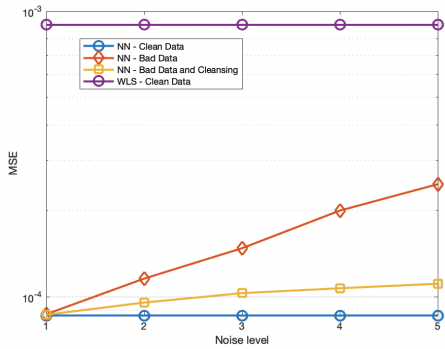


Fig. 7: The performance of Bayesian neural network + universal bad data detection.

V. CONCLUSION

This paper presents a machine learning approach to fast timescale state estimation with limited PMU deployments. We develop a deep learning architecture that integrates bad data detection, data cleansing, and Bayesian state estimation. A major contribution of this work is a universal bad-data detection algorithm for unknown distributions under regular and abnormal operating conditions. Numerical tests show considerable gain over the state-of-the-art benchmark solutions.

REFERENCES

- [1] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, pp. 645–658, Dec 2011.
- [2] M. Zhou, V. A. Centeno, J. S. Thorp, and A. G. Phadke, "An alternative for including phasor measurements in state estimators," *IEEE Transactions on Power Systems*, vol. 21, pp. 1930–1937, Nov 2006.
- [3] L. Vanfretti, J. H. Chow, S. Sarawgi, D. Ellis, and B. Fardanesh, "A framework for estimation of power systems based on synchronized phasor measurement data," in *2009 IEEE Power Energy Society General Meeting*, pp. 1–6, July 2009.
- [4] M. Gol and A. Abur, "A hybrid state estimator for systems with limited number of PMUs," *IEEE Transactions on Power Systems*, vol. 30, pp. 1511–1517, 2015.
- [5] M. Glavic and T. Van Cutsem, "Reconstructing and tracking network state from a limited number of synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 28, pp. 1921–1929, May 2013.
- [6] B. A. Alcaide-Moreno, C. R. Fuerte-Esquivel, M. Glavic, and T. Van Cutsem, "Electric power network state tracking from multirate measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, pp. 33–44, Jan 2018.
- [7] R. F. Nuqui, *State estimation and voltage security monitoring using synchronized phasor measurements*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2001.
- [8] M. B. D. C. Filho and J. C. S. de Souza, "Forecasting-aided state estimation part i: Panorama," *IEEE Transactions on Power Systems*, vol. 24, pp. 1667–1677, Nov 2009.
- [9] A. Bernieri, C. Liguori, and A. Losi, "Neural networks and pseudo-measurements for real-time monitoring of distribution systems," in *Proceedings of 1995 IEEE Instrumentation and Measurement Technology Conference - IMTC '95*, pp. 112–119, April 1995.
- [10] A. S. Debs and R. E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 1670–1678, Sep. 1970.
- [11] A. Abur, "Observability and dynamic state estimation," in *2015 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2015.

- [12] J. Zhao, A. Gmez-Exposito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A. K. Singh, J. Qi, Z. Huang, and A. P. S. Meliopoulos, "Power system dynamic state estimation: Motivations, definitions, methodologies, and future work," *IEEE Transactions on Power Systems*, vol. 34, pp. 3188–3198, July 2019.
- [13] F. C. Schweppe, J. Wildes, and D. P. Rom, "Power system static state estimation, Parts I, II, III," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 120–135, 1970.
- [14] P. Chavali and A. Nehorai, "Distributed power system state estimation using factor graphs," *IEEE Transactions on Signal Processing*, vol. 63, pp. 2864–2876, June 2015.
- [15] L. Schenato, G. Barchi, D. Macii, R. Arghandeh, K. Poolla, and A. V. Meier, "Bayesian linear state estimation using smart meters and pmu measurements in distribution grids," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 572–577, Nov 2014.
- [16] A. Abur and A. G. Expósito, *Power System State Estimation: Theory and Implementation*. CRC Press, 2004.
- [17] D. M. Falcao, P. A. Cooke, and A. Brameller, "Power system tracking state estimation and bad data processing," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, pp. 325–333, Feb 1982.
- [18] K. Nishiya, J. Hasegawa, and T. Koike, "Dynamic state estimation including anomaly detection and identification for power systems," *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 129, pp. 192–198, Sep. 1982.
- [19] A. Abur, A. Keyhani, and H. Bakhtiari, "Autoregressive filters for the identification and replacement of bad data in power system state estimation," *IEEE Transactions on Power Systems*, vol. 2, pp. 552–558, Aug 1987.
- [20] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 949–961, Jan 2019.
- [21] T. Schlegl, P. Seebeck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, vol. 54, pp. 30–44, 2019.
- [22] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "Bayesian state estimation for unobservable distribution systems via deep learning," *IEEE Transactions on Power Systems*, pp. 1–1, 2019.
- [23] L. Paninski, "A coincidence-based test for uniformity given very sparsely sampled discrete data," *IEEE Transactions on Information Theory*, vol. 54, pp. 4750–4755, Oct 2008.
- [24] D. Bloom, "A birthday problem," *Amer. Math. Monthly*, vol. 80, pp. 1141–1142, 1973.
- [25] R. Christie, "Power systems test case archive," *Univ. Washington, Seattle, WA, USA*, 1993. <https://www.ee.washington.edu/research/pstca/pf118/pg.tca118bus.htm>.
- [26] M. Pignati, M. Popovic, S. Barreto, R. Cherkaoui, G. Dario Flores, J. Le Boudec, M. Mohiuddin, M. Paolone, P. Romano, S. Sarri, T. Tesfay, D. Tomozei, and L. Zanni, "Real-time state estimation of the epl-campus medium-voltage grid by using pmus," in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, Feb 2015.
- [27] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011.
- [28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *International Conference for Learning Representations*, 2015.
- [29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan,"
- [30] F. Chollet et al., "Keras." <https://keras.io>, 2015.