Stanford University

Team ID: 702

Machine Learning Models for Inverse Power Flow in the Grid

Jennifer Cardona[1], Thomas Gill[2], Siobhan Powell[3]

1. jcard27@stanford.edu , 2. tgill3@stanford.edu , 3. spowell2@stanford.edu

Motivation

Impact of Renewables on the Grid

The addition of distributed energy resources such as solar PV and wind, and electric vehicles is challenging traditional control methods, straining grid infrastructure, and adding variability which makes voltage regulation and planning difficult using conventional modeling tools. Without measurements at every node in the system and good estimates of all line parameters, utilities cannot accurately model the flow of electricity to meet this new need using the known forward mapping:

$$\begin{aligned} p_i &= \textstyle\sum_{j=1}^{n} |v_i||v_j|(G_{ij}\cos(\theta_i-\theta_j)+B_{ij}\sin(\theta_i-\theta_j)) \\ q_i &= \textstyle\sum_{j=1}^{n} |v_i||v_j|(G_{ij}\sin(\theta_i-\theta_j)-B_{ij}\cos(\theta_i-\theta_j)) \end{aligned}$$

Response

We propose to use only node measurement data for the real and reactive power, *p* and *q*, to develop a dynamic, robust, numerical model for the inverse power flow at each bus, *i*:

$$|v_i| = f_i(\vec{p}, \vec{q})$$

Data

- 16 bus truncated section of the IEEE 123 node test feeder
- Feeder was designed to provide voltage drop problems and minimal convergence problems [1]
- Based on this network model, the data was created in Matpower [2] using randomized input load values for real and reactive power, *p* and *q*, between [0,1] or [-1,1] per unit.
- *m* = 8760 samples, representing one year of hourly data.

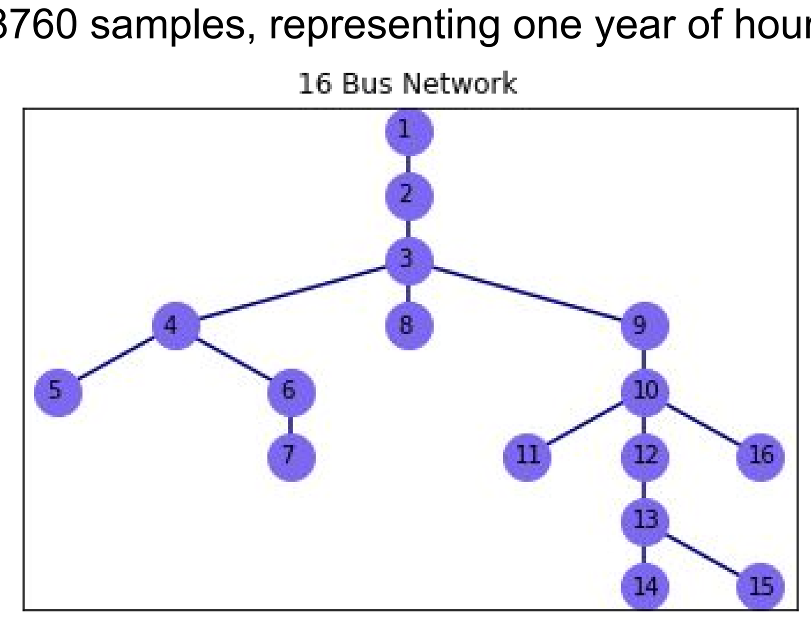


Figure 1. Sketch of network configuration for 16-bus system. Buses and lines are represented by nodes and edges. Bus 1 is the reference or slack bus, and buses 2 - 16 are pq load buses.

Features

Inputs:

- Real power, *p*, at all 16 buses and
- Reactive power, *q*, at all 16 buses
- 32 total features
- No historic values for *p* or *q* are used for the current time step prediction of $|v|$

Outputs:

- Model used to predict voltage, $|v_i|$, at each of the 16 buses

Models

Weighted Linear Regression

- Used gaussian scheme to give buses 'nearest' the query point heaviest weighting
- Tuned width parameter to yield weight of >10% for all buses, solved weighted normal equations

$$\begin{aligned} w^i &= exp(-(x^i-x)^2)/2\tau^2) \\ \theta &= (X^TWX)^{-1}X^TWY \\ \hat{y} &= \theta^T X \end{aligned}$$

Support Vector Regression

- Used four different kernel types, K: linear, quadratic, cubic, and RBF (gaussian)
- Coefficients used to predict v as shown
- *C* gives regularization, ϵ selects support vectors only outside small band [3]
- Coefficient $(\alpha_k-\alpha_k^*)_t$ relates training sample x_t to prediction for bus *k*, is zero for non SVs

$$\begin{aligned} &\underset{\alpha,\alpha^*}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j)+\epsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \\ &\text{subject to} \quad \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \\ &\qquad\qquad 0 \le \alpha_i,\alpha_i^* \le C \qquad \forall i=1,\ldots,N \end{aligned}$$

$$v_k = \sum_{t=1}^{N}(\alpha_k-\alpha_k^*)_t K(x_t,x)+b_i$$

Neural Network

- Network structure and learning rate were tuned using 3-fold cross validation
- Sigmoid function used as activation function for hidden layers
- Output layer activated with identity function

Layer Inputs:

$$z_1 = W_1^T x + b_1 \\ \vdots \\ z_n = W_n^T a_n + b_n$$

Layer Outputs:

$$a_1 = g_1(z_1) \\ \vdots \\ a_{n-1} = g_{n-1}(z_{n-1}) \\ \hat{y} = g_n(z_n)$$

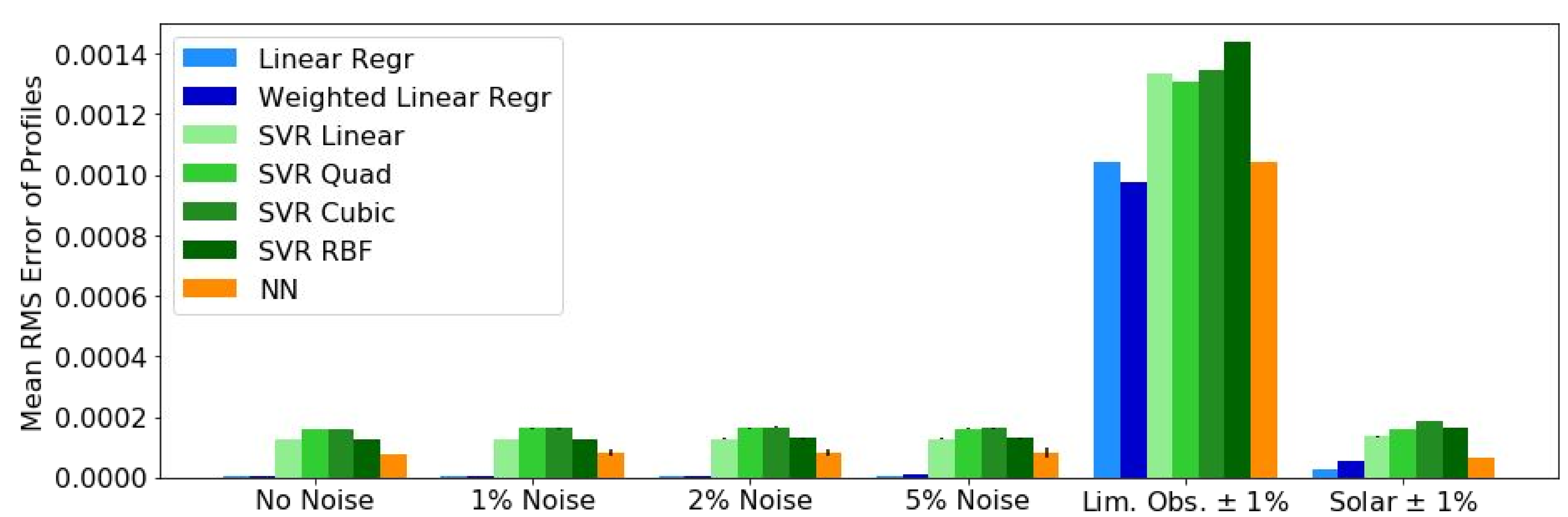
Activation Functions:

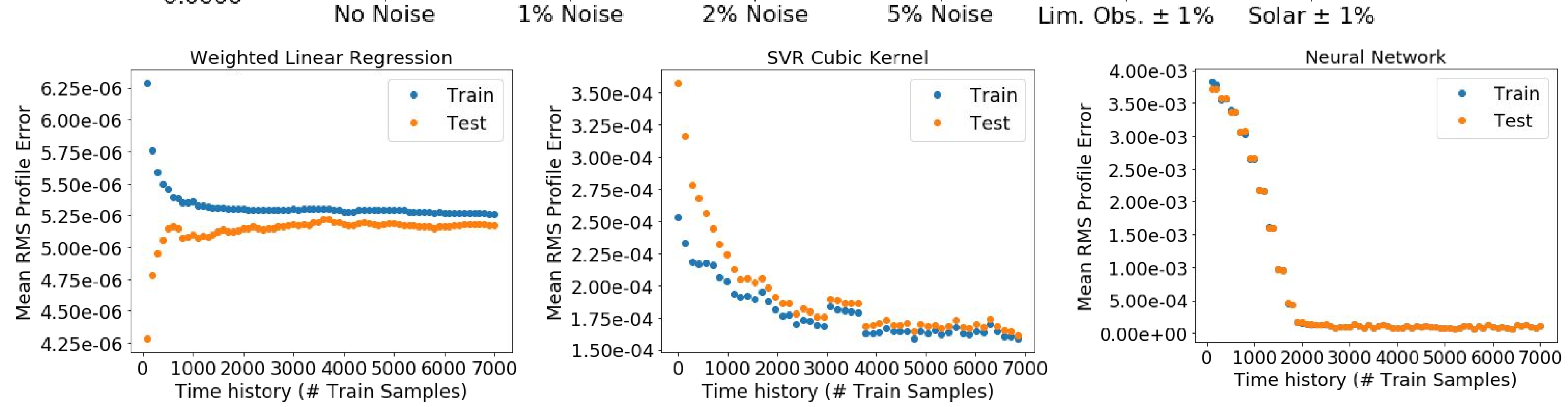
$$g_1(z) = \frac{1}{1+\exp(-z)} \\ \vdots \\ g_{n-1}(z) = \frac{1}{1+\exp(-z)} \\ g_n(z) = z$$

Results

Generating Results:

- The data set was split 80%-20% into train and test sets, and cross validation was applied within the training set for development.
- Models were first implemented by hand in MATLAB, then compared to python scikit-learn (scikit-learn was used from there on)
- Model parameters were optimized with GridSearchCV for each run
- To replicate the results under different levels of **measurement noise**: random noise was added to the input features, the model was fit using the noisy inputs, the test error was calculated through comparison to true output values from the test set, and this test error was averaged over 15 runs of adding noise
- **Limited observability** case (Lim. Obs.) uses only leaf nodes (buses 1, 5, 7, 8, 11, 14, 15, 16)
- **Solar** case uses data which includes negative real loads





Results (cont.)

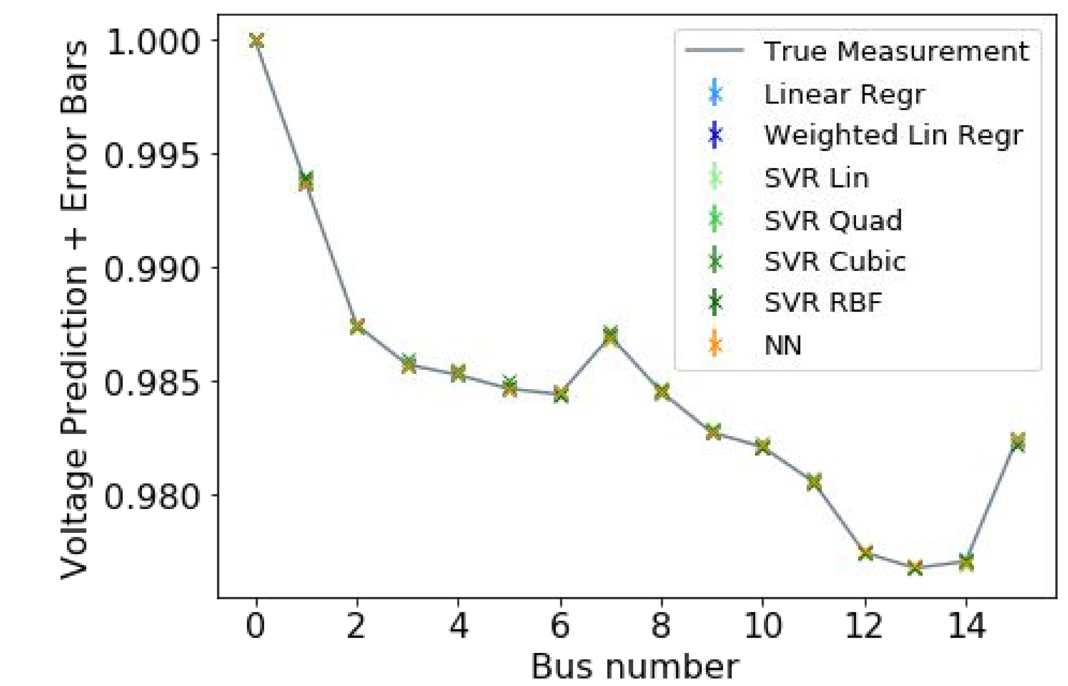


Figure 2. Voltage magnitude predictions on the test set sample with the largest voltage drop. All model types are shown to match the profile very well, with standard deviation between estimates too small to see.

Discussion

- Of all models tested, linear regression performed best, the locally weighted implementation winning the observability test.
- That the limited observability case proved most difficult for modeling shows that collecting as many measurement points as possible is more important than the choice of model.
- It is interesting that in observability test, the neural network model matched the linear regression and the ordering of the kernels in SVM changed, suggesting networks with limited or unknown observability are better captured with more complicated models.
- The linear model corresponds to the lossless version of the power flow equations, where it is assumed that no energy is lost moving along the lines.
  - The losses that do exist are directly a function of the inputs used: surprising the other models could not capture them.
- Beyond accuracy, operators may compare models based on simplicity, computational cost, robustness to their case, or other metrics:
  - Ease of online updating, calculation complexity varies between models with similar accuracies shown here
  - The simple linear model is much better for optimization and analysis, eg. in voltage regulation

Future Work

- Testing on larger models, where greater variability and loss values are expected to require more complicated models
- Using the simple, optimal model for each application to conduct control, optimization, power network analysis

References

[1] J. Fuller, "Distribution Test Feeders," *IEEE PES: Power & Energy Society*. [Online]. Available: https://ewh.ieee.org/soc/pes/dsacom/testfeeders/ . [Accessed: 11-Dec-2017].

[2] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[3] Deng N, Tian Y, Zhang C. Support vector machines: optimization based theory, algorithms, and extensions. CRC press; 2012 Dec 17.