Fast AI-Based Power Flow Analysis for High-Dimensional Electric Networks

Ali R. Al-Roomi
Department of Electrical and Computer Engineering
Dalhousie University
1459 Oxford Street, Halifax, NS B3H 4R2 Canada
Email: Ali.Ridha@dal.ca | Tel.: +1(902) 989-4589

Mohamed E. El-Hawary
Department of Electrical and Computer Engineering
Dalhousie University
1459 Oxford Street, Halifax, NS B3H 4R2 Canada
Email: ElHawary@dal.ca | Tel.: +1(902) 473-6198

2020 IEEE Electric Power and Energy Conference (EPEC) | 978-1-7281-6489-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/EPEC48502.2020.9320057

***Abstract*—It is not revealing a secret to say that most of the power system studies highly depend on power flow (PF) analysis. This tool provides a frozen picture of dynamic electric networks under certain conditions. Nowadays, many iterative techniques are available to solve PF problems. The most popular one is built based on the Newton-Raphson (NR) algorithm. Although NR can obtain highly accurate solutions, its processing speed significantly decreases as the problem dimension increases. Thus, a wise selection should be taken to compromise between the processing speed and the solution accuracy. For some critical studies, such as contingency analysis, the processing speed is a very important factor that forces some energy management systems (EMS) to apply DC and AC-DC approximations. This paper studies the processing speed when artificial neural networks (ANNs) are adopted to solve PF problems. First, the standard 9-bus test system is used. Then, the processing speed of ANNs is examined through a very large virtual network. This AI-based technique can hit multiple birds with one stone. Adding to the processing speed and the solution accuracy, it can also solve the uncertainty issue by feeding ANNs with actual PF readings measured from some mounted instrument devices.**

***Index Terms*—power operation, power flow analysis, load flow analysis, Newton-Raphson, artificial neural networks.**

I. Introduction

THE TERM "power flow (PF)" as in some references [1]–[12], or "load flow (LF)" as in others [2], [9], [13]–[21], is frequently used as a subject of the most important tool in electric power systems engineering. Some nice sentences that can be heard are: Saadat on page 189 of [7]: "*Power flow studies are the backbone of power system analysis and design*.", Bergen on page 150 of [11]: "*It is an integral part of studies in system planning and operation and is, in fact, the most common of power system computer calculations*.", and finally Grainger-Stevenson on page 329 of [5]: "*Power flow studies are of great importance in planning and designing the future expansion of power systems as well as in determining the best operation of existing systems*." This repeated meaning gives a solid conclusion that all electric power systems need some sorts of PF studies to have the ability to measure, monitor, analyze, estimate, predict, and control many variables and parameters to maintain them secure and at their optimal values [20]. This is clearly highlighted by El-Hawary on page 319 of [22]: "*An ubiquitous EMS application software is the power flow program, which solves for network state given specified conditions throughout the system*." Therefore, to understand the importance of power flow studies, the following questions should be raised first: What does the "power flow" term mean? What are the techniques used to solve it? What are the pros and cons of each one of these techniques?

It is well known that any electric power system consists of three principal parts: 1) power generation, 2) power transmission, and 3) power distribution. These interconnected systems are represented by branches and nodes with some injected sources and consumption points. The injected sources represent all the types of generating units (thermal, nuclear, hydro turbine-based generators, solar stations, wind farms, etc). The branches are called powerlines (transmission, sub-transmission, and distribution lines), which are connected with each other through some nodes called busbars. The consumption points are defined as loads. They could be sub-transmission customers (26 kV to 69 kV), primary customers (13 kV to 4 kV), and secondary customers (120 V / 60 Hz "American Standard" or 240 V / 50 Hz "European Standard"); or even the consumption of auxiliary plants (air and gas compressors, lube oil cooling systems, lightings, etc) of power stations themselves. Also, batteries, ultra-capacitors, and flywheels are special bi-directional elements. They act as loads when there is enough power flowing in the grid, and act as power sources in case there is a shortage of electricity production to meet the demand.

After modeling these components from their physical structures to some mathematical expressions, the interconnected system can then be represented as a power electric circuit. Although network equations can be formulated in different forms, the most commonly one used for power system analysis is called the "node voltage method" [17], [22]. If the given network is formulated in a nodal admittance form, then it can be expressed in linear algebraic equations by its node currents. But, practically, electric systems are represented by power values instead of currents, which results in a set of nonlinear algebraic equations called "power flow equations". These equations can be solved by iterative techniques [6], [7], [23]. That is, solving power flow equations leads to knowing about the voltage magnitude ($|V|$) and its phase angle (δ) at each busbar as well as the real power (P) and reactive power (Q) flowing through each line. Moreover, from these essential data, many other information can be easily calculated in some sub-algorithms embedded within EMS, which can be used later for many other studies [5].

Based on this brief description, it can be said that the power flow analysis is carried out to ensure that the following requirements are satisfied [6], [8], [24]:

1) Each bus voltage magnitude is close to its rated value:
$$|V| \approx V^{\text{rated}} \tag{1}$$
2) The total power generation (P_T) should meet the total consumed power as follows:
$$P_T = P_D + P_L \tag{2}$$
where P_D and P_L are the real load demand and power loss, respectively.
3) All the generators should not exceed the specified real and reactive power limits:
$$P^{\min} \leq \quad P_G \quad \leq P^{\max} \tag{3}$$
$$Q^{\min} \leq \quad Q_G \quad \leq Q^{\max} \tag{4}$$
4) Lines and transformers are not overloaded:
$$I \leq I_L^{\max} \times \text{OLF} \quad , \quad \text{where} \quad \text{OLF} = 1.25 \text{ to } 1.5 \tag{5}$$
where I and $I_L^{\max}$ denote the measured and maximum allowable currents, and OLF stands for overload factor [24].

Thus, the independent variables here are $|V|$, δ, P, and Q [8], [23]. In PF, each bus has two known (specified) variables and two unknown (unspecified) variables. The type of busbar depends upon the known variables, which can be summarized in Table I.

Nowadays, there are many PF techniques proposed in the literature. Some of these are reported in [7], [12], [21], [25]–[30]. In terms of accuracy, the worst method is the DC load flow, which becomes the best one in terms of processing time. This method is just used in some special applications, like contingency analysis and quick optimal pricing calculations. Also, it is good for getting a general figure or initial point to estimate some online scenarios where the processing time is the most critical factor where some decimal places of tolerance can be sacrificed. It is important to say that, for accurate and precise calculations, this method is totally discarded [31]. Instead, if Tellegen's theorem is applied here, as in [20], [25], then it might provide good results with very limited usage of memory. The Gauss methods have simple calculation steps, which make them easy to program. Also, they require less memory and processing time. However, their sensitivity can be affected by the selection of the slack bus. Moreover, as the network size increases the algorithms utilize more iterations, which is the case faced with real power networks. This phenomenon creates a bold usage limitation [23]. The most popular one is NR method. Some of its main advantages are its high accuracy and quick convergence rate without depending on the network size or the slack bus selection. However, this technique is very hard to implement in some applications because it consumes a large amount of CPU time and data storage [23]; especially with radial systems where most of the Jacobian matrix elements are zero. This means that if the lower and upper off-diagonal non-zero elements (σ) are equal or three times that of the diagonal elements (i.e., $\sigma = 1 \to 3$), then with a 5000×5000 matrix the following useless memory can be faced:

- The total matrix elements: $n^2 = 5000^2 = 25,000,000$
- The total non-zero elements: $n + 2\sigma n = (1 + 2\sigma)\, n = 15,000$ to $35,000$

Imagine! There are 24,965,000 elements, which are saved just as zeros!! This logical astonishment can be clearly seen during coding NR in any specialized numerical programming language.

As a summary, PF analysis can be translated as a "frozen" picture of one moment, condition, or scenario of a dynamic interconnected electric power system [9].

This paper tries to solve the entire PF problem by using ANNs that accept the same input and output variables of classical PF solvers. To validate the working principle of this scheme, two numerical experiments are covered here. One is conducted based on the Western System Coordinating Council (WSCC) 9-bus test system. The other one is a very large virtual test system (consists of 100,000-bus), which is just used to test the processing speed of ANNs.

A. *Paper Organization*

The remaining parts of this paper are arranged as follows: Section II presents the procedure used to create the database required for ANNs. Section III shows the neural network configurations, and followed by some numerical experiments in Section IV. Finally, the paper is concluded in Section V.

II. Stage No.1: Creating ANNs Database

To make ANNs applicable in any numerical problem, it is important to feed these networks with a matrix of input values (predictors or independent variables) and a matrix of output values (responses, targets, or dependent variables). Thus, to implement ANNs for solving PF problems, the preceding strategy should be applied here. Many studies have been reported in the literature, which use ANNs to solve many highly complicated power system problems. For this particular problem, it is important to say that some wise steps should be considered during creating the data-set. Some of these essential steps are:

TABLE I
Bus Types and Their Known/Unknown PF Variables

Busbar Type	**Known**	**Unknown**
Swing bus[a]	$\|V_i\|$ and δ_i	P_i and Q_i
Generator bus[b]	P_i and $\|V_i\|$	Q_i and δ_i
Load bus[c]	P_i and Q_i	$\|V_i\|$ and δ_i,

[a]Also called slack or reference bus
[b]Also called voltage-controlled or PV bus
[c]Also called PQ bus

- Different settings of generating units and loads should be provided through a random process.
- Reasonable predictors should be added in the input matrix:
  - $|V|$ and δ of the slack bus.
  - P and $|V|$ of the generator buses.
  - P and Q of the load buses.
- The status of all the branches should be considered too.
- The output matrix, which contains the actual responses or targets, should be produced by using some highly accurate PF solvers, such as NR.

Summing all these steps together in a systematic process will result in constructing a general flowchart similar to that shown in Fig. 1.

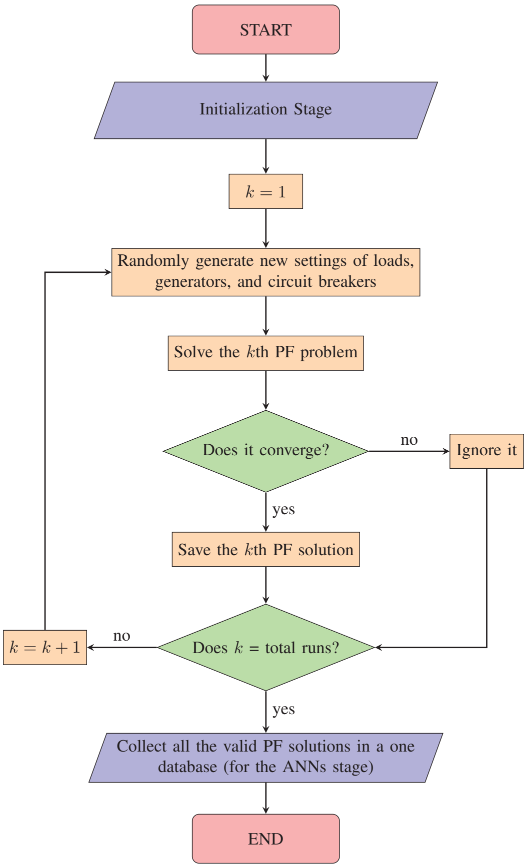


Fig. 1. General flowchart used to create a data-set for NNPF.

Besides, highly advanced ANN-based power flow (NNPF) solvers can be created by considering the variations on system parameters due to the dynamic disturbances on the system frequency and the surrounding weather conditions [32], [33]. Furthermore, real-time readings measured by EMS can be directly utilized to have a realistic NNPF solver that can strongly stand against the uncertainty of the system.

III. Stage No.2: Learning Process of ANNs

Although the input/output (I/O) data-set can be created by many options, this study uses the same configuration implemented in classical PF solvers. That it, to get the PF solution, the I/O variables tabulated in Table I should be provided. This instruments-free power estimator (IFPE) is illustrated in Fig. 2.

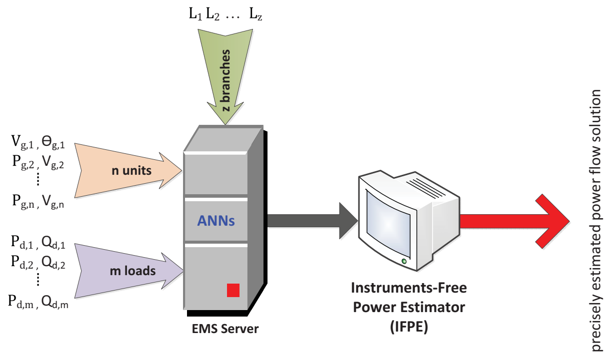


Fig. 2. Mechanism of the proposed IFPE technique.

Now, by combining Fig. 2 with the guidance given in the preceding section, any network topology can be easily constructed. For example, Fig. 3 explains how to generate a data-set of the WSCC 9-bus test system by varying its settings. For this particular test system, there are 18 dependent variables and 18 independent variables. Thus, the input and output layers of any ANN should have 18 connections for each. Thus, if one layer composed of 30 neurons is used, then this shallow topology can be depicted in Fig. 4.

It has to be noted that the classical PF solver used during constructing the data-set is executed only one time and it is an off-line process. The other option is to use the real-data fed by EMS, which is an on-line process.

IV. Numerical Experiments and Discussion

To validate the process of the proposed NNPF technique, the WSCC 9-bus test system shown in Fig. 3 is used in the first experiment. The data-set, created by the algorithm shown in Fig. 1, has a size of 60,000 PF solutions, which is generated by the classical NR algorithm with a minimum acceptable tolerance of $\varepsilon = 10^{-14}$ (i.e., early stopping criterion). In this

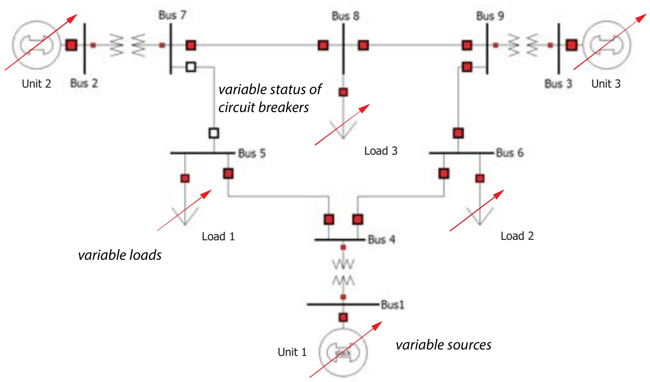


Fig. 3. Operational and topological changes on the WSCC 9-bus test system.

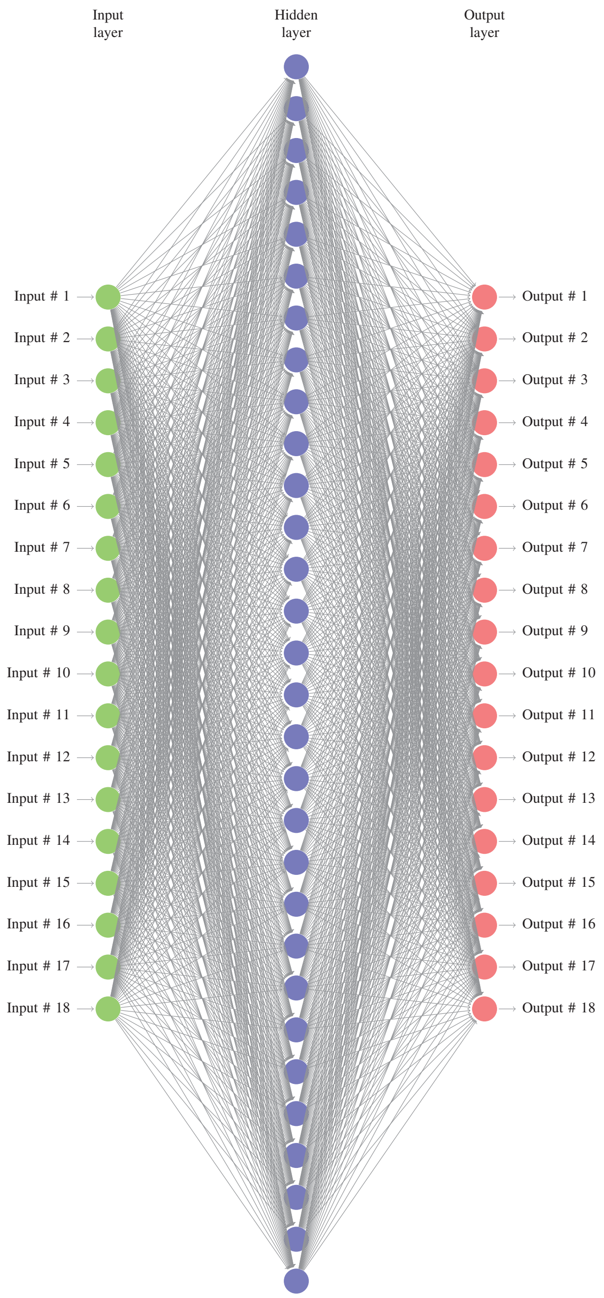


Fig. 4. Neural network of the WSCC 9-bus test system.

experiment, the same topology shown in Fig. 4 is adopted here. Also, the Resilient back-propagation (BP) algorithm is used to train ANNs with the following hyperparameters:

- Maximum number of epochs to train: 100,000
- Performance goal: 0
- Maximum validation failures: 6
- Minimum performance gradient: 1×10^{-7}
- Learning rate: 0.01
- Increment to weight change: 1.2
- Decrement to weight change: 0.5
- Initial weight change: 0.07
- Maximum weight change: 50
- Maximum time to train: ∞
- Ratio of vectors for training = 70%
- Ratio of vectors for validation = 15%
- Ratio of vectors for testing = 15%

The program is coded in MATLAB R2017b using a computing machine having the following specifications: ALIENWARE M14x, 64-bit Windows 10 OS, Intel Core i7-4700MQ CPU 2.4 GHz, and 16 GB RAM. The NNPF performance is evaluated using the following activation functions:

1) `compet`: Competitive transfer function.
2) `elliotsig`: Elliot sigmoid transfer function.
3) `hardlim`: Positive hard limit transfer function.
4) `hardlims`: Symmetric hard limit transfer function.
5) `logsig`: Logarithmic sigmoid transfer function.
6) `netinv`: Inverse transfer function.
7) `poslin`: Positive linear transfer function.
8) `purelin`: Linear transfer function.
9) `radbas`: Radial basis transfer function.
10) `radbasn`: Radial basis normalized transfer function.
11) `satlin`: Positive saturating linear transfer function.
12) `satlins`: Symmetric saturating linear transfer function.
13) `softmax`: Soft max transfer function.
14) `tansig`: Symmetric sigmoid transfer function.
15) `tribas`: Triangular basis transfer function.

It is recommended to set the output layer of ANNs with `purelin` if the goal is to approximate functions [34]; which is the case here.

Table II shows the performance of these 15 activation functions in terms of the mean squared error (MSE), learning ability, and processing speed. It is obvious that the `softmax` activation function is the winner in terms of MSE. This function consumes 14631 epochs to reach its optimal solution. It is much less than that recorded for `elliotsig` by around 46.9%. However, both transfer functions consume almost the same CPU time, which makes `softmax` slower than `elliotsig`. In contrast, `compet` is the fastest one, but it also has the second-worst MSE value after `netinv`.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT ACTIVATION FUNCTIONS

Activation Function	MSE Performance	No. of Epochs	CPU Time[a]
`compet`	0.044999396280284	29	**1.979**
`elliotsig`	0.000078274898843	**31204**	1862.850
`hardlim`	0.017793944764191	1097	64.805
`hardlims`	0.017740511053517	2394	138.288
`logsig`	0.000084297101917	22002	1405.743
`netinv`	0.101490225612570	51	3.810
`poslin`	0.000153187291679	2738	164.170
`purelin`	0.002112044010414	780	43.317
`radbas`	0.000074912310478	9299	584.633
`radbasn`	0.000047250271178	13091	1707.674
`satlin`	0.000125866986436	9466	602.772
`satlins`	0.000111690367707	17045	1107.460
`softmax`	**0.000038978384340**	14631	1863.899
`tansig`	0.000077143655316	15297	978.837
`tribas`	0.000099901261563	6699	429.498
Best	`softmax`	`elliotsig`	`compet`

[a]The unit is in seconds.

As an overall performance of all the 18 output channels of NNPF, Fig. 5 graphically shows the reduction in MSE per epochs for the train, validation, and test sets. The coefficient of determination scored for `softmax` is $R^2 = 99.988\%$, which is shown in Fig. 6. It is impressive with this primitive neural network structure. To view this highly precise approximation, the plots shown in Fig. 7 depict the relation between the actual and predicted readings at some busbars of the WSCC 9-bus test system.

To see the benefit of NNPF, let's repeat the I/O matrices of the preceding test system until reaching 100,000 variables for each matrix. This huge data-set represents a very large virtual network; specifically a 50,000-bus system. With its trained ANN, the processing time required to test 100 PF conditions is just 0.933979 second. On the opposite side, the conventional NR solver requires between 33 and 80 seconds to test only one condition of a 20,000-bus system [35]. This simple simulation reveals a possible very important application of NNPF, which is about expediting the processing speed of contingency and other crucial analysis with high accurate readings. By applying NNPF, no need to use any kind of approximations, like DC and AC-DC load flows. That is, all the inherent weaknesses of existing techniques can be solved, permanently; and, at the same time, it can test hundreds of possible PF scenarios within just a very short time.

V. Conclusion and Scope of Future Work

This study implements ANNs to solve PF problems. The data-sets can be created by varying the network parameters either through the computer models or through collecting real data stored in archiving servers of EMS. The first approach can cover a large search space within a very short period, while the other can solve the uncertainty issue. Thus, a combination of off-line and on-line readings can be used. The numerical experiment conducted on the WSCC 9-bus test system shows that NNPF can provide highly significant PF solutions by just initiating the trained network using the same input variables of classical PF solvers. Also, NNPF is tested using a very large virtual network and it shows that this AI-based method can solve PF problems with a very small increment in the processing time. This means that NNPF is suitable for some critical power system studies, like contingency analysis where precise results of thousands of PF scenarios can be covered within a very short time.

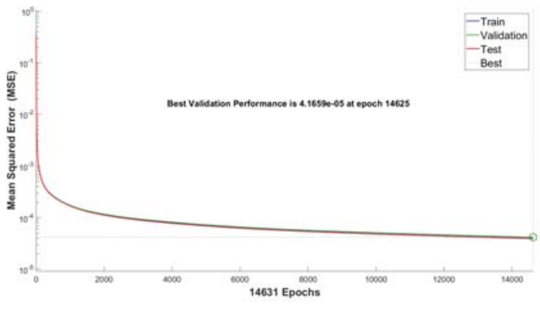


Fig. 5. The MSE performance of the `softmax`-based NNPF.

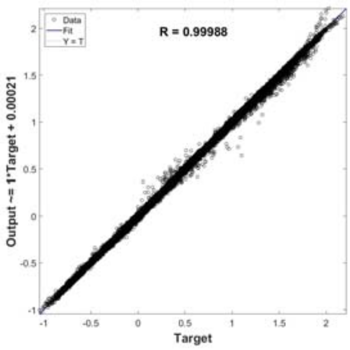


Fig. 6. The overall regression obtained by `softmax`.

The main goal of this study is to prove its feasibility and viability. Of course, many options can be applied to improve its performance, such as using deep-learning structures, optimizing their topologies (number of hidden layers and their associated neurons, types of activation functions, training algorithm and its internal settings, and feedback/recurrent streams), data diversity, data size, etc.

References

[1] James A. Momoh, *Electric Power System Applications of Optimization*, ser. Power Engineering, H. Lee Willis, Ed. New York: Marcel Dekker Inc., 2001.

[2] X.-F. Wang, Y. Song, and M. Irving, *Modern Power Systems Analysis*. New York: Springer US, 2008.

[3] Jizhong Zhu, *Optimization of Power System Operation*, M. E. El-Hawary, Ed. New Jersey: Wiley-IEEE Press, 2009.

[4] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York: John Wiley & Sons Inc., 1996.

[5] J. J. Grainger and J. William D. Stevenson, *Power System Analysis*, ser. Elec. and Comp. Eng., S. W. Director, Ed. McGraw-Hill Inc., 1994.

[6] P.S.R. Murthy, *Power System Analysis*. Hyderabad: BS Pub., 2014.

[7] H. Saadat, *Power System Analysis*. New York: McGraw-Hill Inc., 1999.

[8] J. D. Glover, M. S.Sarma, and T. J. Overbye, *Power System Analysis and Design*, 5th ed. Stamford, CT: Cengage Learning, 2012.

[9] J. C. Das, *Power System Analysis: Short-Circuit Load Flow and Harmonics*, ser. Power Engineering, H. Lee Willis and M. H. Rashid, Eds. New York: Marcel Dekker Inc., 2002.

[10] S. Sivanagaraju and G. Sreenivasan, *Power System Operation and Control*. Chennai: Pearson Education, 2010.

[11] Arthur R. Bergen, *Power Systems Analysis*, ser. Electrical and Computer Engineering, Leon O. Chua, Ed. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1986.

[12] Syed A. Nasar, *Schaum's Outline of Electrical Power Systems*. New York: McGraw-Hill Inc., 1990.

[13] G. W. Stagg and A. H. El-Abiad, *Computer Methods in Power System Analysis*. Tokyo: McGraw-Hill Inc., 1968.

[14] J. Arrillaga and C. P. Arnold, *Computer Analysis of Power Systems*. Chichester: John Wiley & Sons Inc., 1990.

[15] J. Arrillaga and N. R. Watson, *Computer Modelling of Electrical Power Systems*, 2nd ed. Chichester: John Wiley & Sons Inc., 2001.

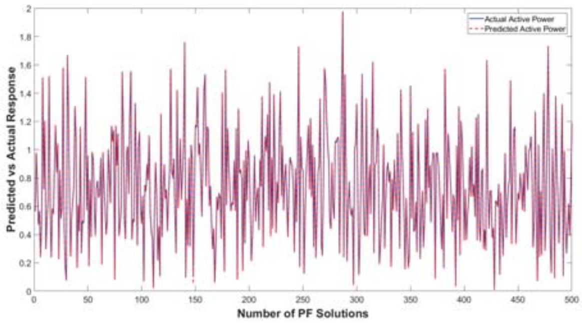
[16] James L. Kirtley, *Electric Power Principles: Sources, Conversion, Distribution and Use*. John Wiley & Sons Inc., 2010.

[17] Debapriya Das, *Electrical Power Systems*. New Delhi: New Age International (P) Ltd., 2006.

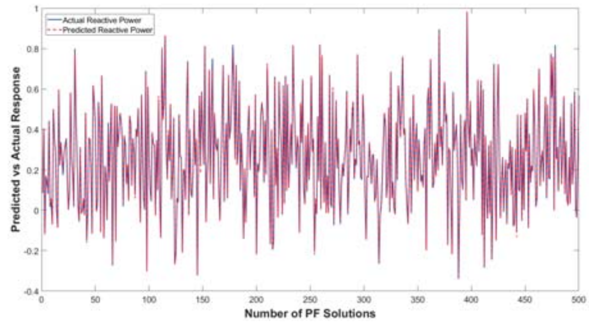
[18] M. E. El-Hawary, *Electrical Power Systems: Design and Analysis*. Reston, Virginia: Prentice-Hall Inc., 1983.

[19] D. P. Kothari and I. J. Nagrath, *Modern Power System Analysis*, 3rd ed. New Delhi: Tata McGraw Hill Education private Limited, 2003.

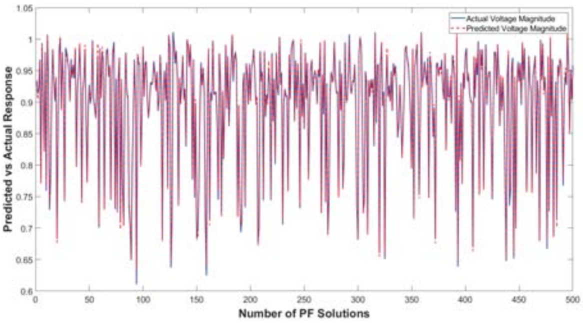
[20] A. S. Debs, *Modern Power Systems Control and Operation*. Boston: Kluwer Academic Publishers, 1988.



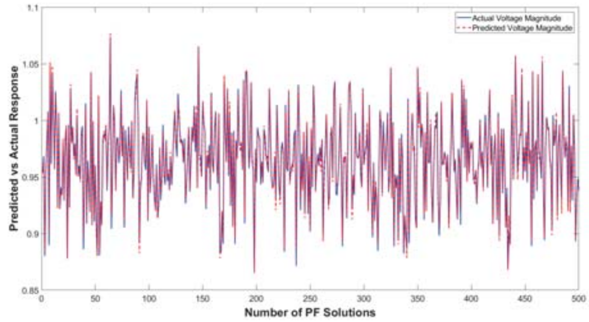
(a) Active power at bus 1



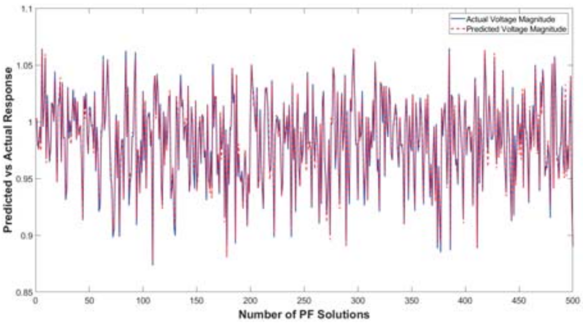
(b) Reactive power at bus 3



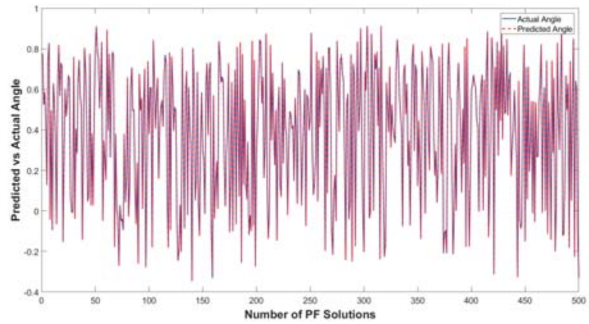
(c) Voltage magnitude at bus 5



(d) Voltage magnitude at bus 6



(e) Voltage magnitude at bus 8



(f) Angle at bus 9

Fig. 7. Some actual and estimated PF measurements at different buses of the WSCC 9-bus test system.

[21] Lynn Powell, *Power System Load Flow Analysis*. New York: McGraw-Hill Inc., 2005.

[22] M. E. El-Hawary, *Introduction to Electrical Power Systems*, Lajos Hanzo, Ed. Hoboken, New Jersey: John Wiley & Sons Inc., 2008.

[23] A. Husain, *Electrical Power Systems*, 5th ed. New Delhi: CBS Publishers & Distributors, 2007.

[24] J. M. Gers and E. J. Holmes, *Protection of Electricity Distribution Networks*, 2nd ed., ser. IEE Power & Energy Series 47. London: Institution Of Engineering And Technology (IET), 2004.

[25] L. A. F. M. Ferreira, "Tellegen's Theorem and Power Systems-New Load Flow Equations, New Solution Methods," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 4, pp. 519–526, apr 1990.

[26] D. A. Knoll and D. E. Keyes, "Jacobian-Free Newton-Krylov Methods: a Survey of Approaches and Applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357–397, 2004.

[27] A. P. S. Meliopoulos, S. W. Kang, G. J. Cokkinides, and R. Dougal, "Animation and Visualization of Spot Prices via Quadratized Power Flow Analysis," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, jan 2003, pp. 7 pp.–.

[28] A. Trias, "The Holomorphic Embedding Load Flow Method," in *Power and Energy Society General Meeting, 2012 IEEE*, jul 2012, pp. 1–8.

[29] T. T. D. Rubira, "Alternative Methods for Solving Power Flow Problems," Stanford, CA, 2012. [Online]. Available: http://www.stanford.edu/class/cme334/docs/2013-10-29-rubira_power_flow.pdf

[30] D. P. Kothari and J. S. Dhillon, *Power System Optimization*, 2nd ed. New Delhi, India: PHI Learning Private Limited, 2011.

[31] A. J. Conejo, "Load Flow," Albacete, Spain, jul 2011. [Online]. Available: http://www.uclm.es/area/gsee/ArchivosPag-web/docencia/aelect/01_LoadFlow_R1.pdf

[32] A. R. Al-Roomi and M. E. El-Hawary, "Effective Weather/Frequency-Based Transmission Line Models—Part I: Fundamental Equations," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Oct 2017, pp. 1–6.

[33] A. R. Al-Roomi and M. E. El-Hawary, "Effective Weather/Frequency-Based Transmission Line Models—Part II: Prospective Applications," in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Oct 2017, pp. 1–8.

[34] Mathworks, "Multilayer Neural Network Architecture (R2017a)," Brochure, 2017, [Accessed Jun. 17, 2017]. [Online]. Available: https://www.mathworks.com/help/nnet/ug/multilayer-neural-network-architecture.html

[35] S. Janković and B. Ivanović, "Application of Combined Newton-Raphson Method to Large Load Flow Models," *Electric Power Systems Research*, vol. 127, pp. 134 – 140, 2015.