Chapter 4

REAL-TIME ESTIMATION OF ATC USING PMU DATA AND ANN

4.1 INTRODUCTION

An ANN architecture for real-time estimation of ATC has been reported in this chapter. The real-time data obtained from PMU is utilized to generate target output (ATC) using Pattern Search Optimization-based method. The set of information provided as input to the Pattern Search based ATC optimizer along with its output forms the input and target output for ANN training. The input information consists of active and reactive power injected along with voltage and current vectors measured at PMU buses. The ATC optimizer is functional as long as ANN is under training. Once the ANN is trained, it receives input set directly from PMU and produces ATC values. PMU emulation is employed for archiving the PMU data. The proposed method is tested on modified IEEE 24-Bus, IEEE 30-Bus, and IEEE 118-Bus test system. The proposed method has also been implemented on Real-Time Digital Simulator (RTDS). The major contributions of the present research work are enumerated as under: -

- Real-time ATC estimator development that employs measurement data that are coming from PMU.
- Feature extraction for ATC estimation using Sparse Filter Algorithm has been proposed for reducing the size of input features.
- The feasibility of real-time applicability of the proposed method has been authenticated by implementation on RTDS.

The method developed in this chapter utilizes the offline ATC Assessment and PMU emulation techniques discussed in Chapters 1 and 2 for generating the training data of the proposed real-time estimator.

4.2 OFFLINE ATC EVALUATION AND TRAINING DATA GENERATION

The technique used for generating training and testing data using the offline ATC evaluation process is schematically shown in the flowchart of Figure 4.2. Initially, the loading in the sink area of the network under consideration has been considered as a lower limit (lb), while the upper limit (ub) has been taken as the maximum loading limit of load buses. Pattern Search (PS) has been employed for generating different loading conditions, solutions for which have been obtained with the help of Newton Raphson (NR) method. The process for evaluation ATC from source to sink has been illustrated in Figure 4.1.

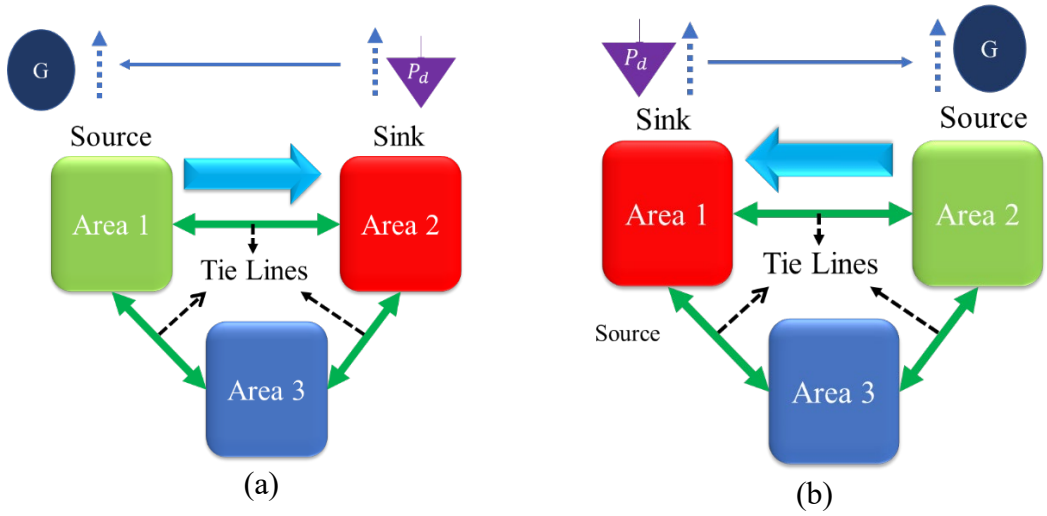


Figure 4.1 Schematic Representation of ATC evaluation.

For evaluation of ATC from Area 1 (Source) to Area 2 (Sink), the loading in the sink area is increased, and the corresponding generation is increased in the source area. The increment in sink and source area is done so as to maintain the generation load balance mechanism. The detailed discussion of offline ATC estimation used for generating offline data for training the ANN-based ATC estimator has been discussed in section 2.4 of Chapter 2.

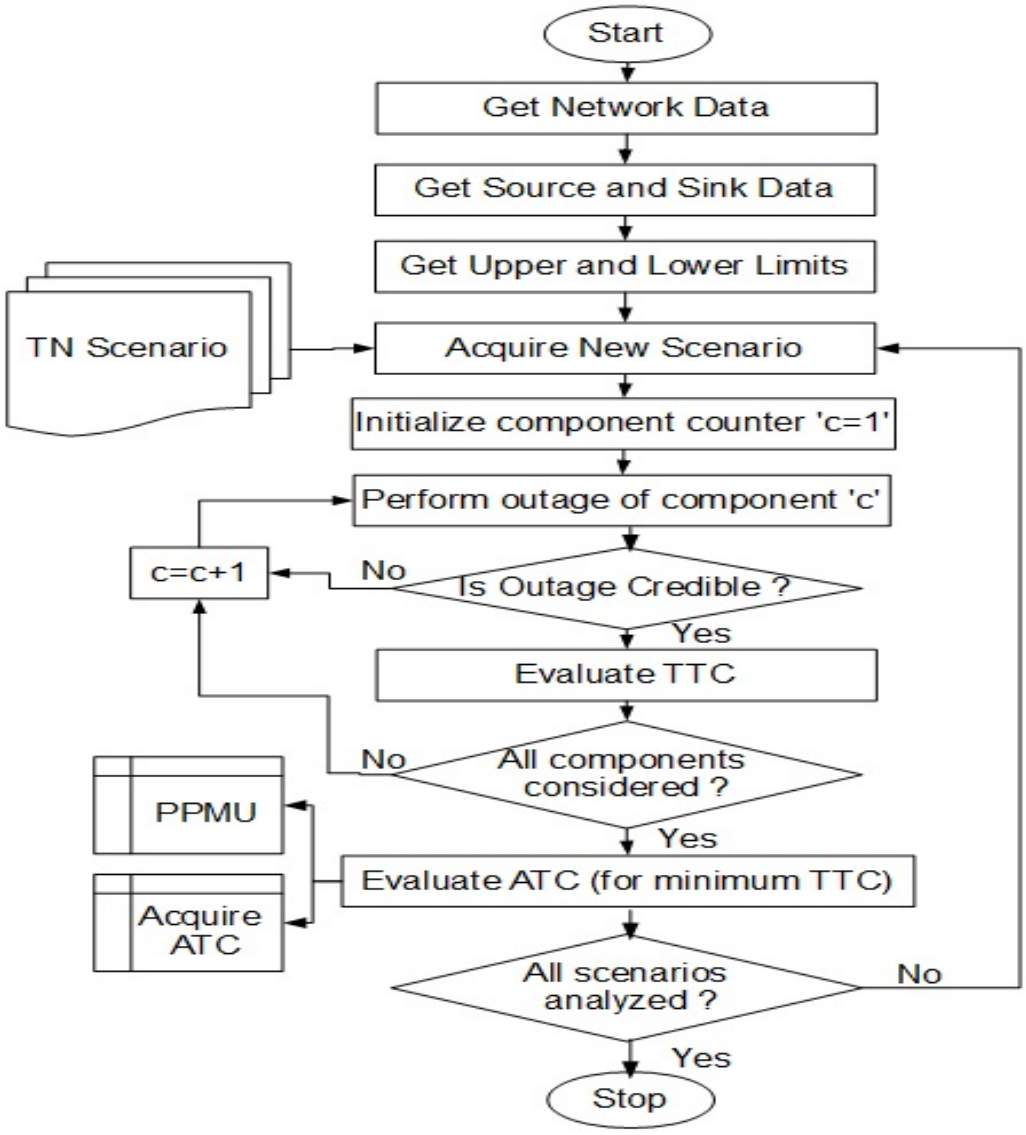


Figure 4.2 Flowchart of offline ATC evaluation and training data generation.

4.3 LINEAR STATE ESTIMATOR

Linear State Estimation is a substitute for Conventional State Estimation for real-time operation of the power system. Linear State Estimation can be deployed to measure the Voltage and Current states of nodes where the PMUs are not placed, provided the PMU placement has been done for full observability of the network under consideration [150]. The problem formulation for Linear State Estimation can be mathematically expressed as: -

$$Min\ \mathcal{R}^T \mu \mathcal{R} \tag{4.1}$$

$$st: z = \mathcal{H}X + \mathcal{R}$$

$$= \begin{pmatrix} \mathcal{H}_{1,1} & \cdots & \mathcal{H}_{1,n} \\ \vdots & \ddots & \vdots \\ \mathcal{H}_{m,1} & \cdots & \mathcal{H}_{m,n} \end{pmatrix} X + \begin{pmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_m \end{pmatrix} \tag{4.2}$$

Here R is the error residue, z is the measured vector, X is the unknown vector, μ is the weight matrix with all entries as real numbers, and μ_i represents the weight for each measurement.

$$\mu_i = \begin{pmatrix} \sigma_{z,real}^2 & 0 \\ 0 & \sigma_{z,imag}^2 \end{pmatrix} \tag{4.3}$$

The solution to the above state estimation is obtained without iteration:

$$X = (\mathcal{H}\mu\mathcal{H})^{-1}\mathcal{H}^T \mu z \tag{4.4}$$

4.3.1 Formation of $\mathcal{H}$ matrix

The $\mathcal{H}$ matrix used in linear state estimation is obtained from the admittance matrix formed after the network information is archived and processed by the network

topology processor. The dimension of the $\mathcal{H}$ matrix is $(m_u \times m_k)$ where m_u is the number of unknown measurements and m_k represents the number of known measurements. Let n_b be the number of buses at which PMUs are not placed and n_p be the buses at which the PMUs have been placed, with n_{pb} being the number of branches connected at each bus. Now, $m_u = n_b - n_p$ and $m_k = n_p + i = \sum_{i=1}^{n_p} n_{pb}(i)$. Figure 4.3 illustrates an example of four buses interconnected network where bus 1 is the $PMU_{locationbus}$, line $lid(1)$ with tap changing transformer, and other lines connected to far end buses $2, 3\ and\ 4$. Now, the network admittance matrices are formed using: -

$$Y_s = \frac{1}{R_{lid} + jX_{lid}}; \quad Y_{tt} = Y_s + \frac{jB_{c,lid}}{2} \tag{4.5}$$

$$Y_{ff} = \frac{Y_{tt}}{tap \times conj(tap)}; Y_{ft} = \frac{-Y_s}{conj(tap)}; Y_{tf} = \frac{-Y_s}{tap} \tag{4.6}$$

$$Y_F = \begin{bmatrix} Y_{ff} & Y_{ft} \\ Y_{tf} & Y_{tt} \end{bmatrix}; Y_T = \begin{bmatrix} Y_{tt} & Y_{tf} \\ Y_{ft} & Y_{ff} \end{bmatrix} \tag{4.7}$$

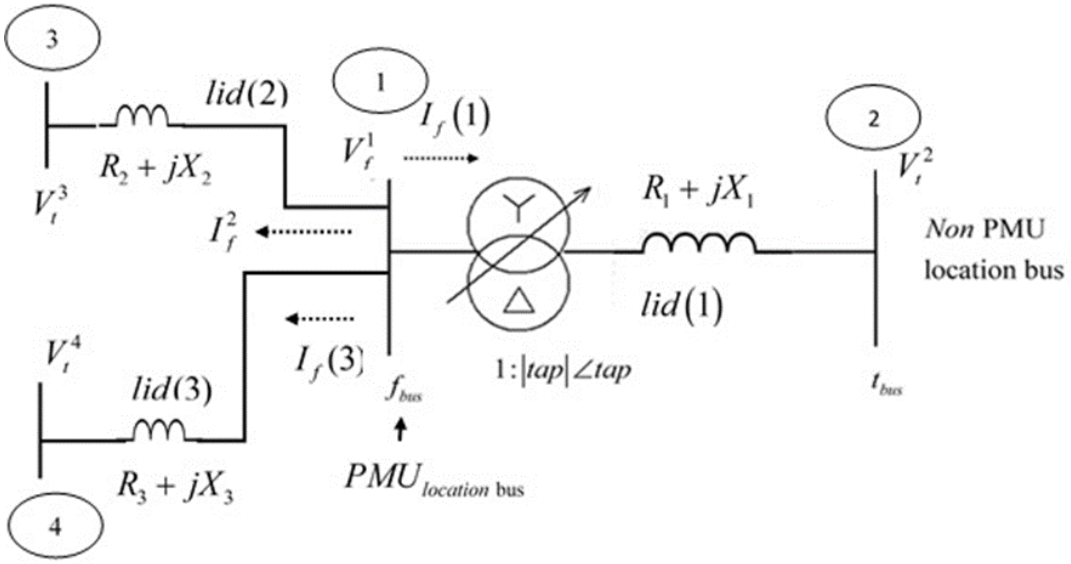


Figure 4.3 Illustrative example showing three buses connected to a PMU bus.

The voltage V_f and branch current I_f are known at the PMU location bus. This information can be used to obtain the information of far end bus using (32).

$$I_{ft} = Y_{ff}V_t + Y_{ft}V_f; \Rightarrow V_t = \underbrace{\frac{I_{ft}}{Y_{ff}}}_{H_{ft}} - \underbrace{\frac{Y_{ft}V_f}{V_{ff}}}_{H_{ff}} \tag{4.8}$$

The measurement vector Z with size $(1 \times m_k)$ is obtained from the PPMU measurements as: -

$$Z = \begin{bmatrix} PPMU(1).V \\ PPMU(1).I_{ft^1} \\ PPMU(1).I_{ft^{n1}} \\ \vdots \\ PPMU(np).V \\ PPMU(np).I_{ft^1} \\ PPMU(np).I_{tf^{n1}} \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \\ m_{k\_1} \\ \underbrace{m_k}_{cc} \end{matrix} \tag{4.9}$$

The elements of $\mathcal{H}$ matrix are obtained corresponding to the Voltage measurements of the X vector are obtained as: -

$$\mathcal{H}\left(i, n_p(i)\right) = \mathcal{H}\left(i, n_p(i)\right) - \frac{Y_F(PMU(n_p(i).x(lid,1), n_p(i))}{Y_F(PMU(n_p(i)).x(lid,1), PMU\left(n_p(i)\right).z1(lid,1))} \tag{4.10}$$

$$\mathcal{H}(i, cc(i)) = \frac{1}{Y_F(PMU\left(n_p(i)\right).x(lid,1), PMU\left(n_p(i)\right).z1(lid,1))} \tag{4.11}$$

The various structure field of PMU and their accessing methods used in (34) and (35) are given in Table 4.1 to Table 2.3.

.

Table 4.1 Structure Field and accessing method of PMU.

PMU	Parameter	Accessing Method	Variable	Identifier used
Structure Field	id	$PMU(1).id$	PMU location	np
	x	$PMU(1).x$	Lines/branches connected at PMU bus	lid
	$z1$	$PMU(1).z1$	Far end bus of branches connected at PMU bus	NA
	I	$PMU(1).I$	Line currents	NA
	V	$PMU(1).V$	PMU bus voltage	NA

Table 4.2 Structure Field and accessing method of $PMU(id).x$

PMU(id).x

	Accessing Method	Accessed Variable/Parameter
	$PMU(1).x(1,1)$	out-bound/outgoing branch connected to PMU bus
	$PMU(1).x(2,1)$	inbound/incoming branch connected to PMU bus
	$PMU(1).x(lid,1)$	lid^{th} the branch connected to the PMU bus

Table 4.3 Structure Field and accessing method of PMU(id).z1.

$PMU(id).z1$

	Accessing Method	Accessed Variable /Parameter
	$PMU(1).z1(1,1)$	far end bus of out-bound/outgoing branch connected to PMU bus
	$Pmu(1)..z1(2,1)$	far end bus of inbound/incoming branch connected to PMU bus
	$PMU(1).z1(lid,1)$	far end bus of the branch (lid^{th}) connected to the PMU bus

4.4 REAL-TIME ESTIMATION OF ATC USING ANN

4.4.1 Radial basis function neural network

The radial basis function network is a conglomeration of two types of neurons (RBF and linear) arranged in three (input, hidden, and output) layers. The received input information is transmitted through an input layer to the next layer, called the hidden layer. The hidden layer and output layer consist of RBF and linear neurons, respectively. The

linear neurons are realized using a purlin transfer function. Each layer consists of several nodes that are fully connected to the relevant (preceding and/or succeeding) layer. Moore Penrose Generalized pseudo-inverse method has been used to obtain the weights between hidden and the output layers. The Moore Penrose method has been preferred over the other methods such as LMS (Least Mean Square), LLSR (linear least square regression, etc.) due to its shorter training time and generalization ability. This property of the method makes it suitable for real-time applications, i.e., the present problem. The Gaussian kernel has been used as a radial basis function (for pattern recognition). The K_{means} clustering has been employed for determining the center and width of the kernel.

4.4.2 ANN architecture

The architecture of ANN presented in this section has been given in Figure 4.4. The RBF network architecture with the input set for the present problem has also been illustrated in the figure. The input to the network mainly consists of four different types of data; these data are Voltage magnitude and angle of the buses at which the PMUs are located, branch currents, and branch current angels of the branches terminating or originating at the PMU location buses. If there are m PMU's optimally located in the system, then there will be m voltage magnitude $|V|$ inputs $(V_1, V_2, \ldots . V_m)$ and m voltage angle inputs V_α $(\alpha_{V_1}, \alpha_{V_2}, \ldots . \alpha_{V_m})$. Similarly if there are r branches terminating or originating at k^{th} PMU-bus then the $branch\ current$ input of this unit would be $(I_{k1}, I_{k2}, \ldots I_{kr})$ and $branch\ current$ angles would be $(\beta_{I_{k1}}, \beta_{I_{k2}}, \ldots . \beta_{I_{kr}})$ and in this fashion the entire branch current and branch current angle information is fed as input as $(I_{11}, I_{12} \ldots I_{mr})$ and $(\beta_{I_{11}}, \beta_{I_{12}}, \ldots . \beta_{I_{mr}})$. The size of the input vector would be $2 \times m + \sum_{i=1}^{m} 2r_i$. This vector is passed through the LSE for obtaining all the states of the system (an amalgam of measured variables from PMU buses and estimated variables by LSE).

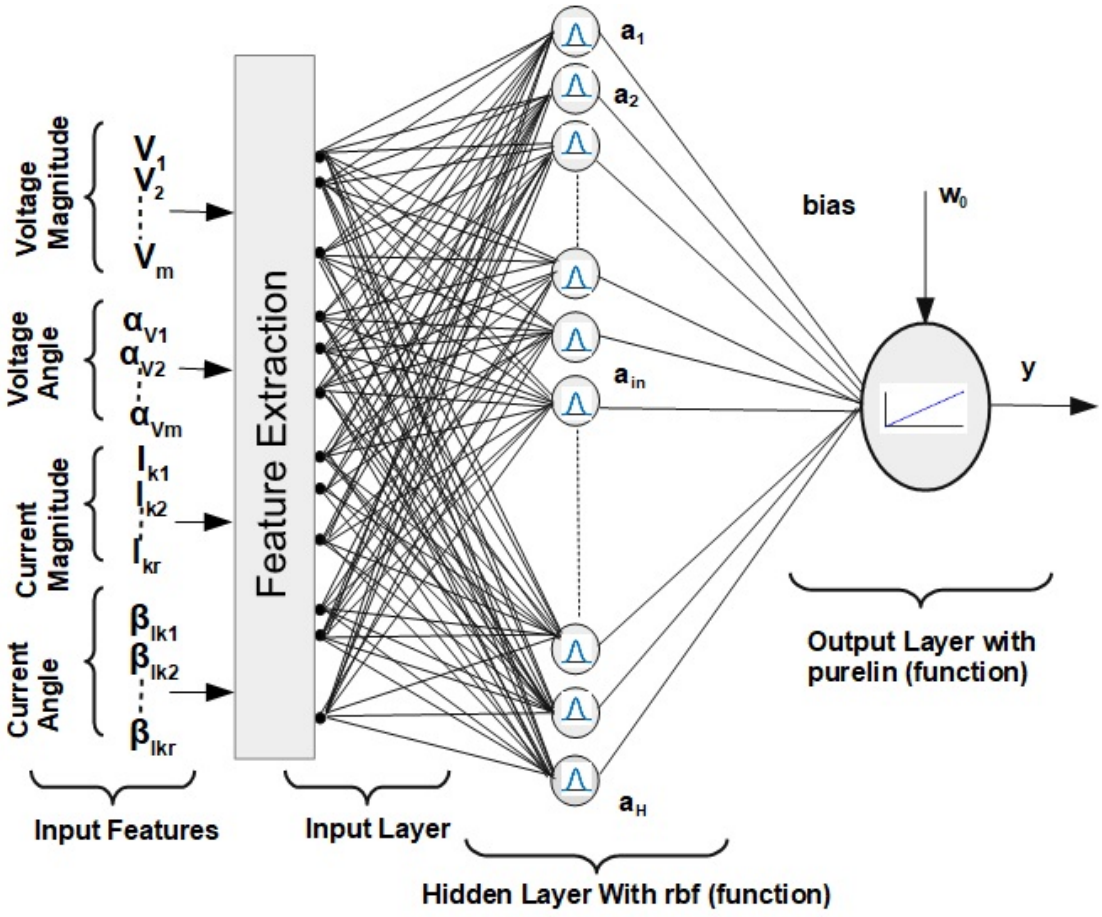


Figure 4.4 RBF architecture Proposed.

The output of the LSE, i.e., $|V|, \alpha_v, I, \beta_I$ is feature set $\mathcal{D}$ which comprises a large number of features. For efficient performance of ANN-based estimators, these large features set have to be reduced to a feature set $\mathcal{D}^r$ capable of adequately representing the complete set of Features. The D^r set of reduced features discussed in section 5.3 is used as input, and the corresponding ATC value obtained using the pattern search optimization is used as the target output. The training process involves the optimal determination of the center and width of the RBF neurons so as to meet the required criterion of error tolerance.

4.4.3 Feature Extraction

Feature extraction (FE) is an unavoidable process that is to be performed when a large number of data features are present. Feature extraction involves mapping of input features (large number) to new output features (fewer number). In this chapter, a Sparse Filtering Algorithm (SFA) extraction method is used for extracting the dominant features that could be used as input to the ANN. The sparse filtering algorithm begins with a data matrix $\mathcal{D}$ having n rows and p columns where they $(rows, columns)$ represent observations and measurements, respectively. The algorithm then takes either an initial random p-by-q (required number of features) weight matrix $\mathcal{W}$ and minimizes (36). The method shortens the initial feature set $\mathcal{D}$ to a smaller set $\mathcal{D}^r$ using the matrix $\mathcal{W}$ obtained by the SFA algorithm. (36).

$$Min \sum_{i=1}^{m} \left\| \widetilde{\mathcal{D}}^{r,i} \right\|_1 = \sum_{i=1}^{m} \left\| \frac{\widetilde{\mathcal{D}}^{r,i}}{\left\| \widetilde{\mathcal{D}}^{r,i} \right\|_2} \right\|_1 \qquad 4.12$$

In order to compute objective function (36), which depends on the n-by-p data matrix D and a weight matrix $\boldsymbol{W}$ the procedure given in Table 4.4 has been employed. The schematic representation SFA method has been illustrated in Figure 4.5.

4.4.4 Testing

A part of the data archived for training is extracted and kept aside for training and validation of the trained neural network. If the desired goal of tolerance is met in the testing phase, then the neural network is assumed to be trained, and final weights and biases are exported as a trained network that could be used for real-time online estimation.

Table 4.4 SFA Algorithm for Feature Extraction

Stage	Description
Step 1	Compute the n-by-q matrix $\mathcal{D} \times \mathcal{W}$ and using the approximate absolute value function $\psi(u) = \sqrt{u^2 + \epsilon}$, obtain the matrix D^r. ψ is the symmetric non-negative, and smooth function which approximates the absolute value function and ε has been taken as 10^{-8} [151].
Step 2	Define Normalized $\widetilde{\mathcal{D}}^{r,i}(i,j)$ as $\widetilde{\mathcal{D}}^{r,i}(i,j)=\frac{\widetilde{\mathcal{D}}^{r,i}(i,j)}{\|\widetilde{\mathcal{D}}^{r,i}(j)\|_2}$ where $\widetilde{\mathcal{D}}^{r,i}(j) = \sqrt{\sum_{i=1}^{n} \widetilde{\mathcal{D}}^{r,i}(i,j)^2 + \epsilon}$.
Step 3	Define Normalized $\widetilde{\mathcal{D}}^{r,i}(i,j)$ as $\widetilde{\mathcal{D}}^{r,i}(i,j)=\frac{\widetilde{\mathcal{D}}^{r,i}(i,j)}{\|\widetilde{\mathcal{D}}^{r,i}(i)\|_2}$ where $\widetilde{\mathcal{D}}^{r,i}(i) = \sqrt{\sum_{i=1}^{n} \widetilde{\mathcal{D}}^{r,i}(i,j)^2 + \epsilon}$.
Step 4	Compute the objective function $h(\mathcal{W})$ as the $l - norm$ of the matrix $\widetilde{\mathcal{D}}^{r,i}(i,j)$ using $h(\mathcal{W}) = \sum_{j=1}^{q} \sum_{i=1}^{n} \widetilde{\mathcal{D}}^{r,i}(i,j)$.
Step 5	Obtain the optimal value W using suitable techniques. Here, Broyden-Fletcher-Goldfarb-Shanno (LBFGS) quasi-Newton optimizer has been employed [152] with termination criterion as the maximum number of iterations and step tolerance (if the norm of the algorithm step at any time is less than the specified tolerance).

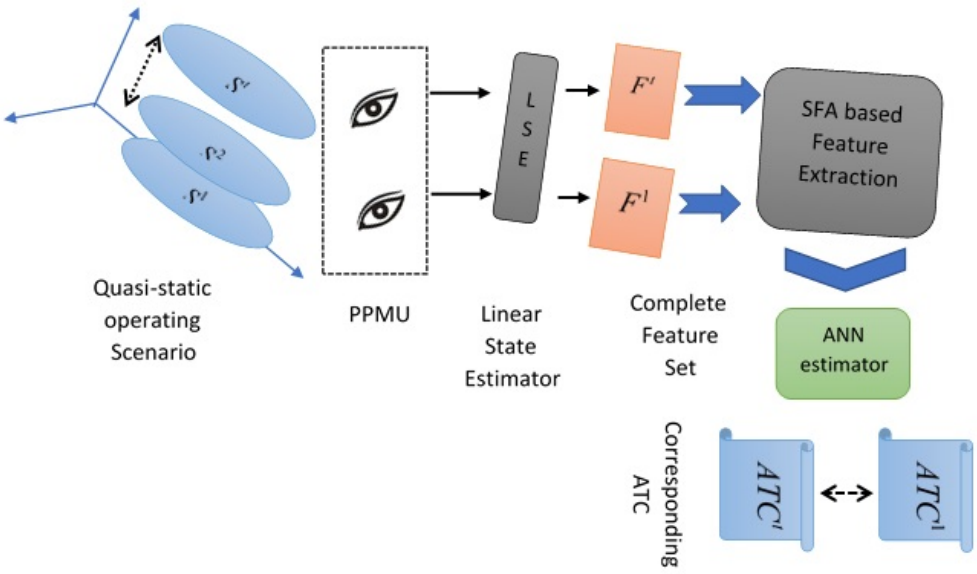


Figure 4.5 Schematic Representation for Feature Extraction.

The software-based implementation of the proposed scheme has been depicted in Figure 4.6, and practical hardware-based implementation has been shown in Figure 4.7. During software-based implementation, there are apparently two different stages of the scheme: - offline training and online implementation. The system scenario is inputted to the PSN solver, which is common to both online and offline stages. The PSN solver produces a solution corresponding to the input scenario and shares it with the Pseudo-PMU emulator (PPMU) and Optimization engine simultaneously. The optimization engine yields the optimal ATC values using the pattern search optimization method discussed in this chapter and stores it in ATC storage. In ATC storage, the ATC values are stored sequentially to be used as target output to the Neural Network Training block.

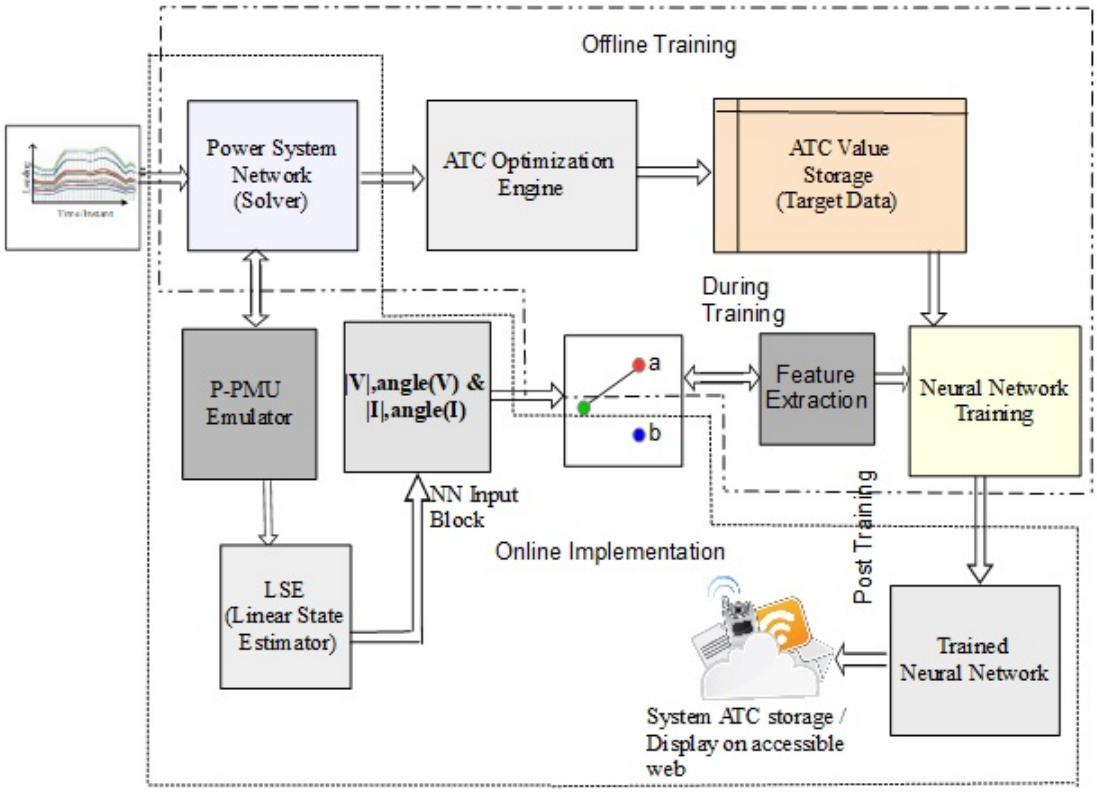


Figure 4.6 Software-based development of the proposed method.

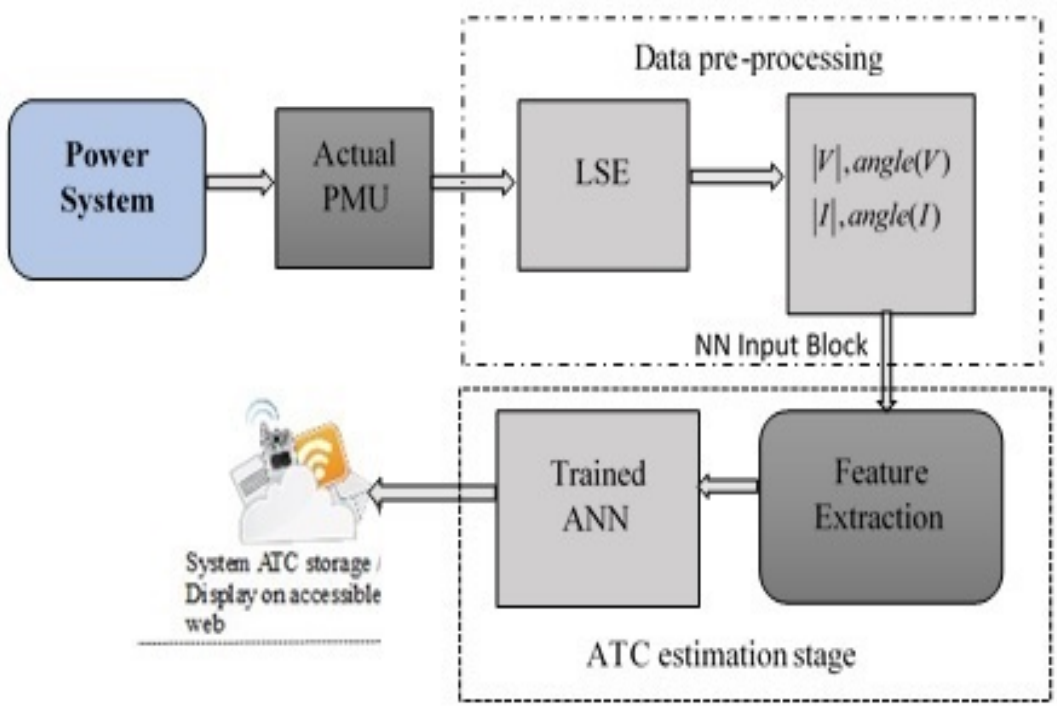


Figure 4.7 Practical Implementation layout of the proposed method.

Concurrently, the PPMU emulator interacts with the PSN solver to archive the voltages and currents of the PMU location buses and sends them to the LSE block. In the linear state estimator block, all the states of the network are obtained. The LSE block sends the linearly estimated states to the ANN input block. It is to be clarified that the values sent by the ATC storage as target and the input sent by the $NNinput$ block to the ANN are quasi-statically synchronized. The ANN input block feeds the ANN training block till it is trained. This is depicted by placing the switch at position $'a'$ of the switch block shown in Figure 4.7. The switch being in position $'b'$ the trained ANN would produce ATC value for any novice input received from the ANN block. Once the network is trained, the final weights and biases are exported to the Trained ANN block. This can be accomplished by diverting the switch from position $'b'$. The ATC value so produced can be sent to the ATC storage for further applications such as power transaction management, forecasting, system analysis, display on the publicly accessible web, etc. The above-described methodology can be practically implemented using trained ANN by a utility, as shown in Figure 4.7. The PSN solver and the input scenario have been replaced by the actual power grid, whereas the PPMU emulator has been replaced by the actual PMU in this figure. The PMU measurements have been directly sent to the pre-processing data stage (i.e., LSE and Feature Extraction) for ATC estimation by the trained ANN.

4.6 AUTHENTICATION IN REAL-TIME SIMULATION USING RTDS

The functioning of the ANN estimator has been verified in real-time using RTDS (Real-Time Digital Simulator) platform. The test system has been modeled on RSCAD software and compiled on RTDS hardware. The different loading scenarios have been sent to the RTDS simulator from the monitoring and control center that has been built in MATLAB.

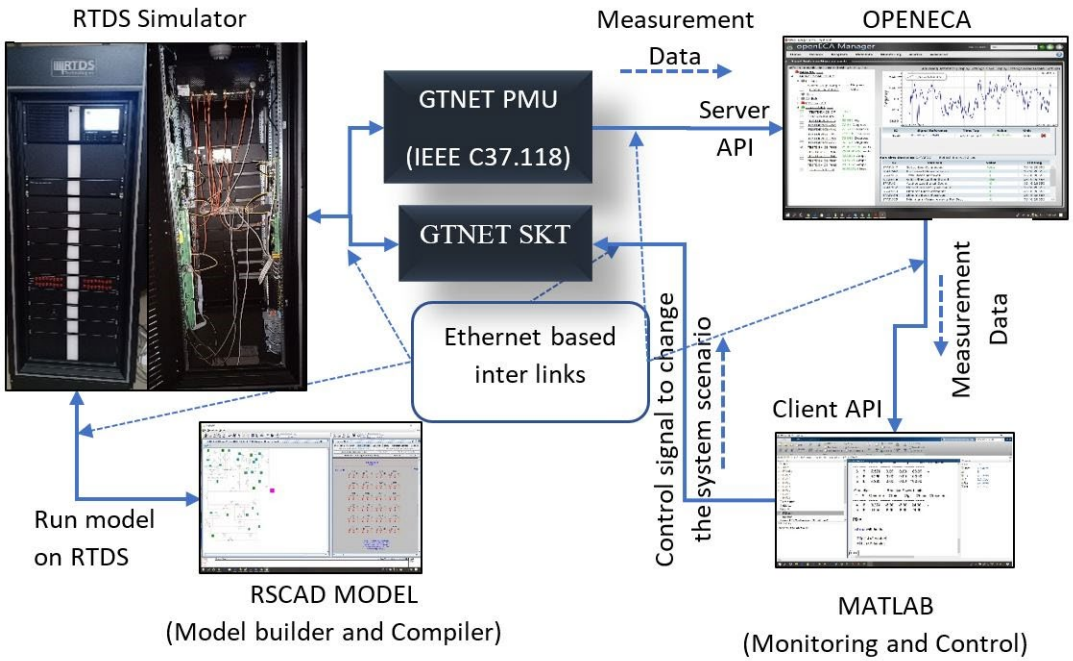


Figure 4.8 Pictorial illustration of the process used in authentication in RTDS simulation.

The control center issues the appropriate command signals corresponding to any change in the operating scenario of the system. This communication is established through the 'GTNET SKT' protocol. The 'GTNET SKT' protocol is capable of handling data streams with an update frequency of 2 kHz (maximum). A schematic representation of the process has been shown in Figure 4.8. The GTNET PMU has been placed at the optimal PMU locations in the power system model using RSCAD software. The RTDS uses the GTSYNC card to time stamp the measurement taken by the GTNET PMU's. These PMU's could publish data in the IEEE C37.118 protocol and send the data to other devices and applications such as PDC. An open-source PMU connection tester provided by GPA (Grid Protection Alliance) [153] has been used to testing and generating PMU configuration files. These configuration files are used by the openECA to establish a connection with GTNET PMU and acquire measurement data using server API. The $OPENECA$ sends the data to the monitoring and control center using the client API. The

ATC estimator running in the monitoring and control center has been used to estimate the ATC in real-time.

4.7 CASE STUDY

The methods proposed in this chapter have been simulated on intel core i7 @ 2.24 GHz computer using MATLAB 2017b and MAT- POWER 6.01 software. The description of the test systems on which the proposed methods have been illustrated is delineated in the following subsystem.

4.7.1 Description of Test System

The performance of the method has been manifested on IEEE 24 Bus, 30 Bus, and IEEE 118 Bus systems. The data used for modified IEEE 24, 30, and IEEE 118 Bus test systems have been taken from [23], [154], and [155], respectively. Table 4.5 contains the information pertinent to the area wise classification of buses (both generator and load buses), optimal location of PMU's, the number of branches linked with the corresponding PMU buses (given in small bracket alongside the PMU bus location), along with the area wise generation capacity, load and the available generation margin. The generation margin is an indicator that reflects the amount of power which an area can additionally supply to its own area without importing from other areas (while fulfilling the system criterion). The tie lines interconnecting the different areas are given in Table 2.6.

Table 4.5 Description of the test system.

Case	Area	Generator Bus	Load Bus	PMU location bus (Branch)	Gen Cap MW	Load (MW)	Margin (MW)
IEEE 24 BUS Test System	1	[14,15,16,17, 18,19,21]	[5,17,19]	[16 (4), 21 (5)]	1170	1125	45
	2	[13,22,23]	[6,8,9,10,11, 12,20]	[8 (3), 10 (5), 23 (4)]	1551	1141	410
	3	[1,2,7]	[3,4,24]	[2 (3), 3 (3)]	684	584	100
IEEE 30 BUS Test System	1	[1,2]	[3,4,5,6,7,8,9 ,11,28]	[1 (2), 2 (4), 6 (7), 9 (3)]	121.94	84.5	37.44
	2	[13,23]	[12,14,15,16, 17,18,19,20]	[12 (5), 15 (4), 19 (2)]	40	76.2	-36.2
	3	[22,27]	[10,21,24,25, 26,29,30]	[10 (6), 25 (3), 27 (4)]	105	48.5	56.5
IEEE 118 Bus Test System	1	[61,62,65,66, 69,70,72,73,7 4,76]	[60,67,75,78, 7947]	Taken from [156]	2468.2	599	2049.2
	2	[99,100,103,1 04,105,107,11 0,112,113,116]	[98,101,102, 106,108,109, 114,115,117, 118]	Taken from [156]	1229.2	715	577

Table 4.6 Details of Test Cases Under Consideration (Base Case).

Area	**Tie Line**	
	24 BUS System	**30 BUS System**
1 to 2	21-22,17-22,19-20(2),14-11	4-12,
1 to 3	15-24	9-10,6-10,28-27
2 to 3	3-9,4-9,1-5,2-6,7-8	17-10,20-10,23-24

4.7.2 Result and Discussion

For every input scenario (loads, generation, network topology, and PMU location), different values of TTC (total transfer capability) corresponding to various contingencies have been obtained. These TTC values correspond to the minimum of the voltage, stability, and thermal limits. The TTC value corresponding to the severe-most contingency would be the lowest. For the defined credible set of contingencies, a TTC ranking has been obtained, and the top-ranking TTC value has been used to obtain ATC. Such a selection of (TTC rank 1) for ATC calculation would guarantee the safer operation of the power system as any other single outage contingency would result in a TTC/ATC higher than the rank one selection.

4.7.2.1 *Feature Extraction:*

The result of feature extraction using the method discussed in section 4.4.3 has been obtained for all the three test cases under study as tabulated in Table 4.7. The SFA has been used to reduce the size of the input features. The size of the reduced feature set is equal to the number of optimal PMU's that would be required for ensuring complete observability of the system.

Table 4.7 Reduced Features for different test cases using SFA.

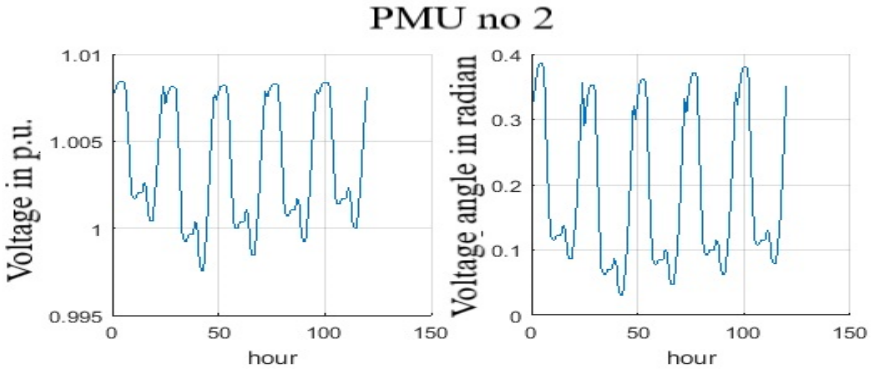
S. N	Test System	After PPMU		After LSE		Reduced Feature
		$[V, \alpha, I, \beta]$	$[V \& I]$	$[V, \alpha, I, \beta]$	$V \& I$	
1	24	68	34	124	62	7
2	30	120	60	140	70	10
3	118	342	171	608	304	33

4.7.2.2 *Modified IEEE 24-bus RTS system:*

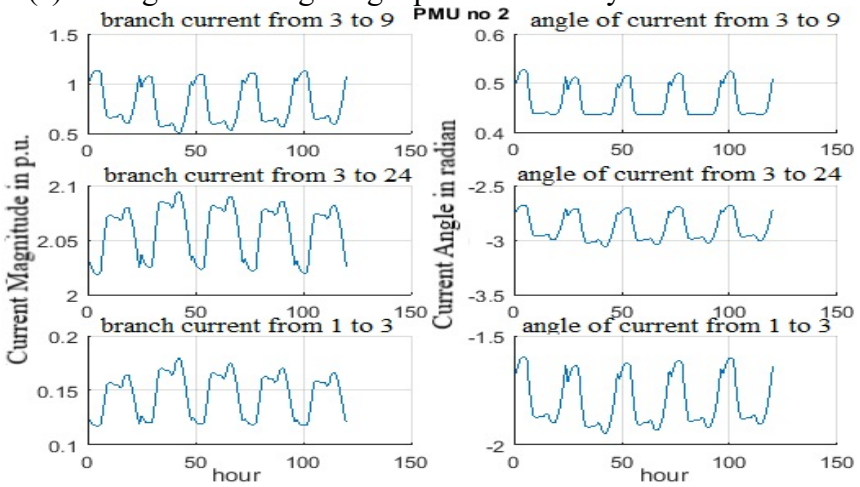
The entire network has been divided into three areas. The ATC for the transfer of power from area 1 to area 2 has been evaluated using the proposed method. The CBM and TRM have been taken as 5 percent of the existing transmission commitments. The results obtained by the proposed method have been tabulated in Table 4.8, along with results reported in [23] for comparison. The TTC value corresponding to the contingency that yields the lowest TTC has been considered for ATC evaluation. Such TTC value corresponds to an outage of the largest outage generator in area 2. It can be noticed from this table that the TTC corresponding to the above contingency produced by the proposed method is higher compared to the method reported in [23]. In other cases, also the TTC is higher in the proposed method. The ATC has been calculated by taking CBM to be 60 MW and tabulated in Table 4.8.

Table 4.8 Comparison of results obtained from TSCOPF and Proposed Method.

Case Description	Proposed Method			TSCOPF		
Outage	TTC (MW)	CBM (MW)	ATC (MW)	TTC (MW)	CBM (MW)	ATC (MW)
Largest gen in area 1	465	-	-	350	-	-
Largest gen in area 2	358.636	60	298.636	262.8	60	200.8
Line 21-22	465	-	-	350	-	-
Line 17-22	465	-	-	350	-	-
Line 19-20	465	-	-	350	-	-
Line 14-1	400	-	-	306.6	-	-



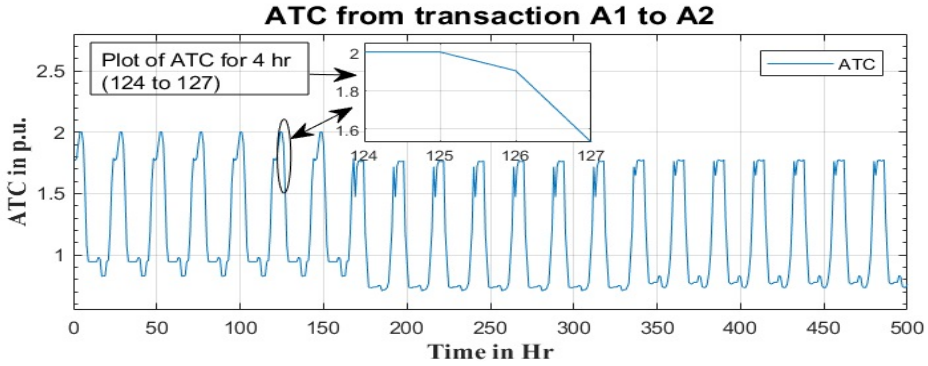
(a) Voltage and voltage angle plot obtained by PMU 1 at bus 2



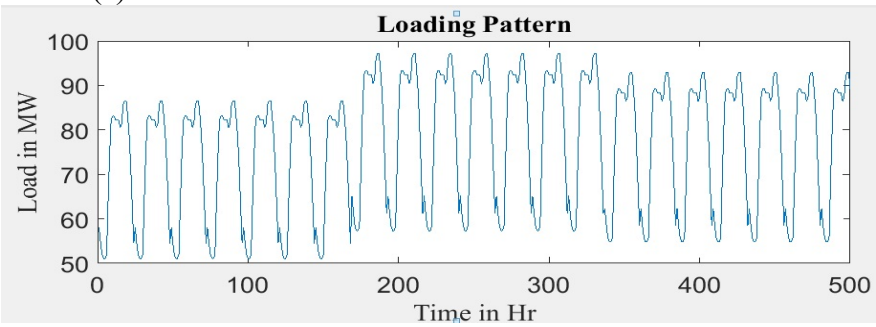
(b) Voltage and voltage angle plot obtained by PMU 1 at bus 2

Figure 4.9 Data Archived by PMU at bus 2.

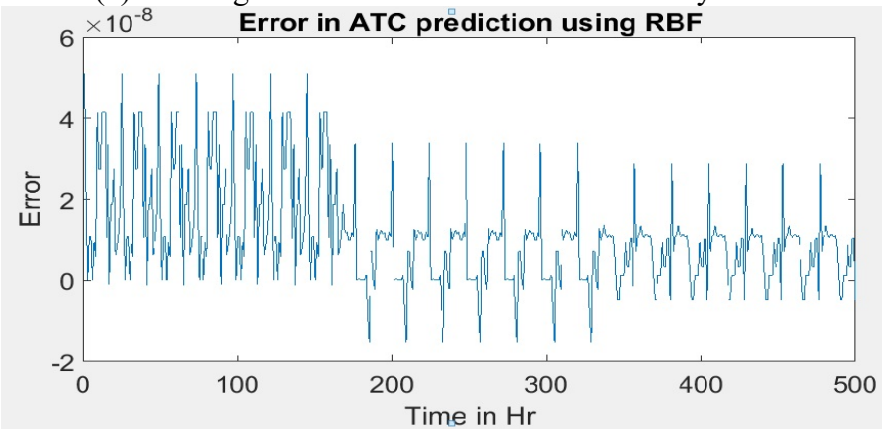
The TRM is neglected for comparison of the proposed method with TSCOPF. It can be observed that the proposed method results in a higher value of ATC than that of the TSCOPF method. The PMU emulation for modified IEEE 24-bus system has also been done and the results obtained by PMU 2 (for example) are given in the Figure 4.9(a) and Figure 4.9(b). The figures show the voltage and current magnitudes and angles recorded by the PMU's as the system is subjected to the predefined loading pattern.



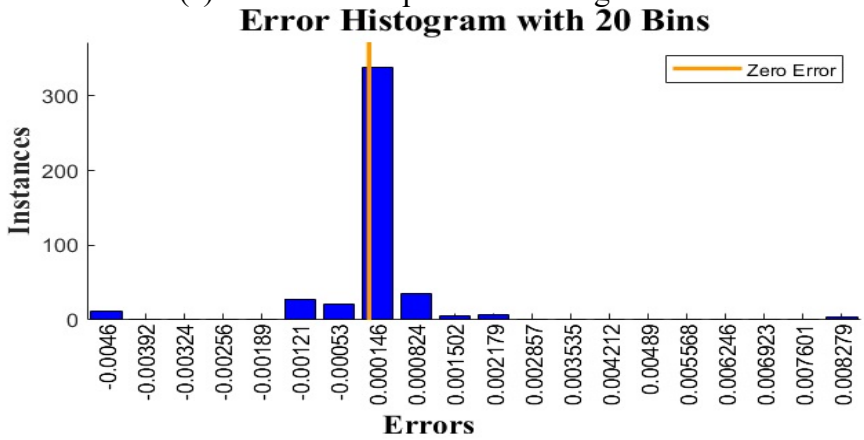
(c) ATC from area 1 to 2 for modified IEEE 24 bus test



(d) Loading Pattern of modified IEEE 24 bus system



(c) Error in ATC prediction using ANN



(d) Error Histogram of ATC estimator using reduced features for IEEE 24 Bus System

Figure 4.10 IEEE 24 bus test system Emulation Results.

PMU's are assumed to be placed at optimal locations with maximum observability as the objective of their placement. The optimal locations of the PMU in this work are taken from reference [157] and are given in Table 4.5. It can be observed from Figure 4.9(a) and Figure 4.9(b) that the hourly load pattern has been quasi-statically simulated, and the data archived by the PPMU emulator for 120 hours are shown. The variations in the voltage and currents obtained are in phase with the variations observed in the loading for the same duration. During the offline simulation stage, the ATC optimization engine evaluated the ATC value corresponding to each loading scenario, and the corresponding value has been stored in the system. ATC storage.

Figure 4.10(a) depicts the plot of ATC in the system from area 1 to 2 without considering any outage in the system, and Figure 4.10(b) depicts the corresponding loading pattern. The data archived from the ATC storage system and PMU emulator has been used for training of RBF and LMNN based neural networks. There are 68 inputs and a single output (ATC values) being used for training purposes; the inputs are obtained from the PMU emulator and comprise of (V $[Magnitude\ and\ Angle]$, Branch Currents $[Magnitude\ and\ Angles]$). The emulation has been done for hourly load model given in reference [157]. The performance of the RBF neural network based estimator for the considered test system has been given in Figure 4.10(c). It can be observed that the error in ATC prediction is less in the case of the RBF neural network, and therefore RBF based neural network ATC estimator has been proposed. The trained estimator has been used in the real-time estimation of ATC. The input to the estimator is the voltages and current measured by the PPMU's across the system.

4.7.3 Modified IEEE 30-bus test system:

The IEEE 30-bus system has been divided into three areas, and ATC for the transaction of power between the areas has been evaluated. The input data has been randomly generated using uniform distribution between 80 to 120 percentages of base case loading by employing the equations given hereunder.

$$P_L^i = P_L^0(1 + \zeta) \quad 4.13$$

$$Q_L^i = Q_L^0(1 + \zeta) \quad 4.14$$

Here, $\zeta = a + rand(b - a), b = 0.2\ and\ a = -0.2.$

The data generation is done by ensuring a fixed loading characteristic that is achieved by maintaining a constant power factor. The condition and convergence of the ATC optimization engine for transfer from area 1 to area 2 have been given hereunder in Table 4.9 and in Figure 4.11, respectively. It could be observed that PS converged in five iterations with -0.70677 as the best function value and 816 function evaluations. In the case of the IEEE 30 bus test system, the input scenarios have been generated by employing (4.13) and (2.58).

Table 4.9 Condition of source and sink areas of the 30-bus test system.

S. No	**Parameter**	**Area**	
		Area 1	Area 2
1	Initial Generation	84.51	56.2
2	Max Generation	160	70
3	Available Generation Capacity	75.49	13.8
4	Contingency Considered	Outage of Generator at bus 23	

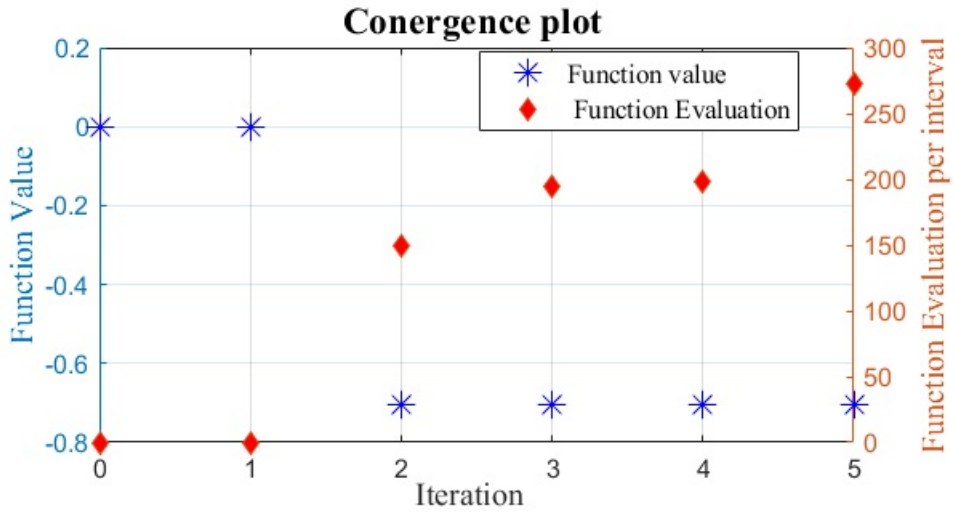
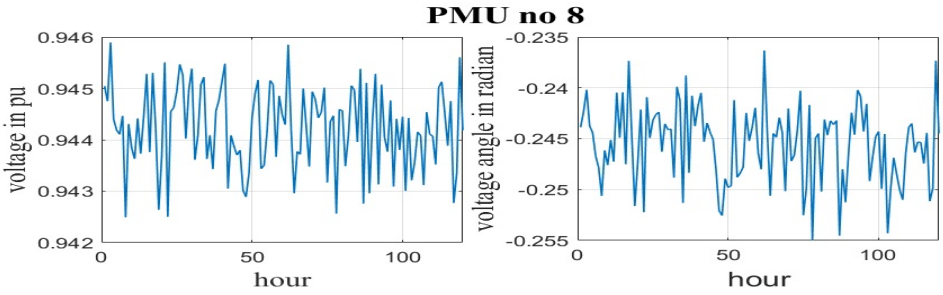
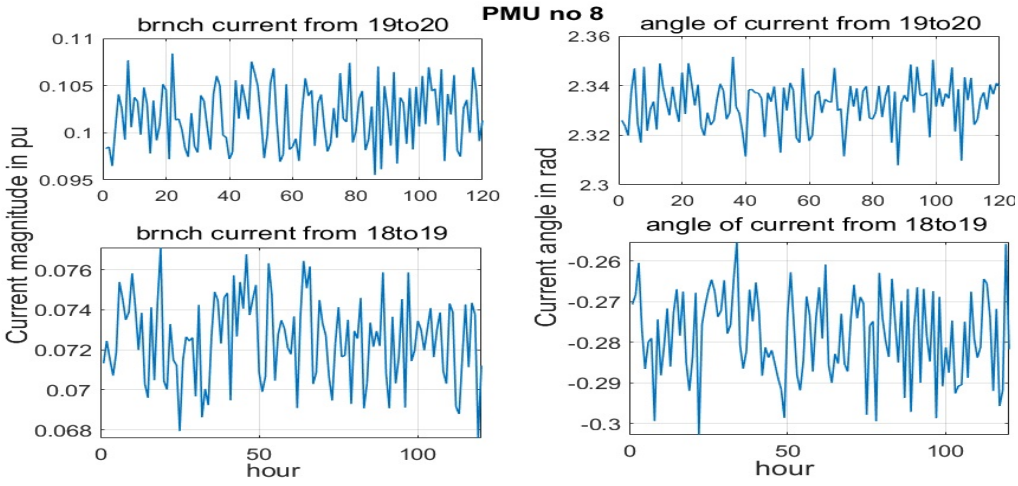


Figure 4.11 Plot of convergence for ATC evaluation of IEEE 30 bus test system from area 1 to area 2.



(a) Voltage and voltage angle plot obtained by PMU 8 at bus 19

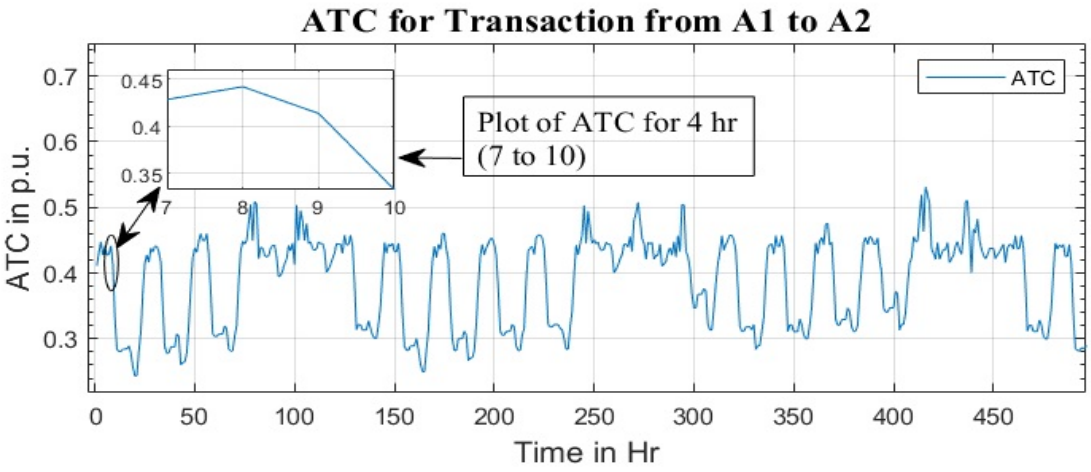


(b) Branch current and angle plot obtained by PMU 8 at bus 19

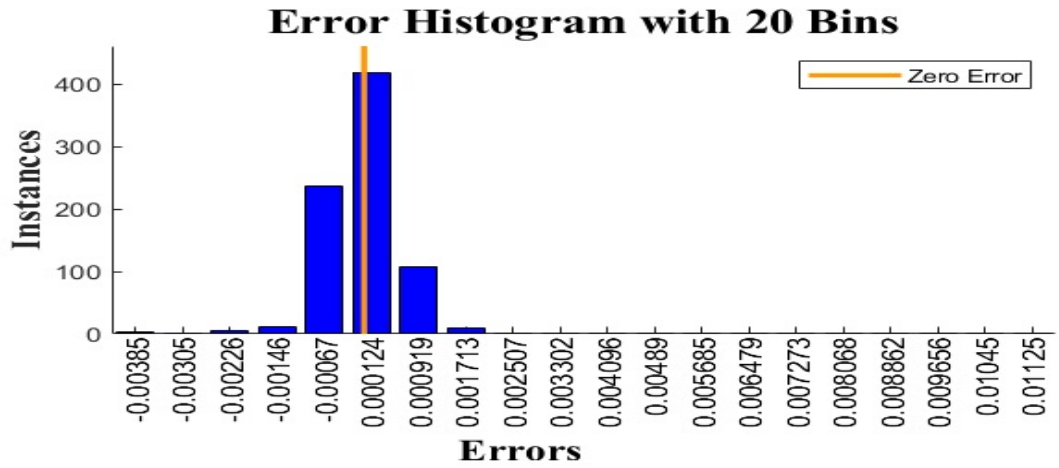
Figure 4.12 PMU emulation for IEEE 30 bus Test System.

The data archived by the PPMU emulator when the generated scenarios are quasi-statically simulated are given in Figure 4.12. The sub-figure Figure 4.12(a) depicts the

voltage plot, and sub-figure Figure 4.12(b) depicts the branch current plot. The ATC estimator for IEEE 30 bus test system has been trained for quasi-static yearly data generated using loading factors given in [143]. The results of ATC estimation and corresponding error are given in Figure 4.13.



(a) Estimated ATC

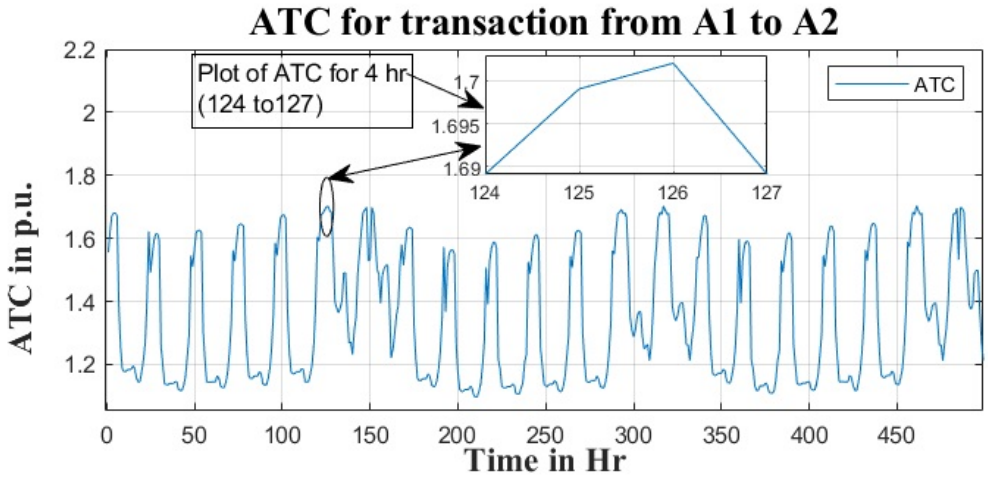


(b) Error Histogram

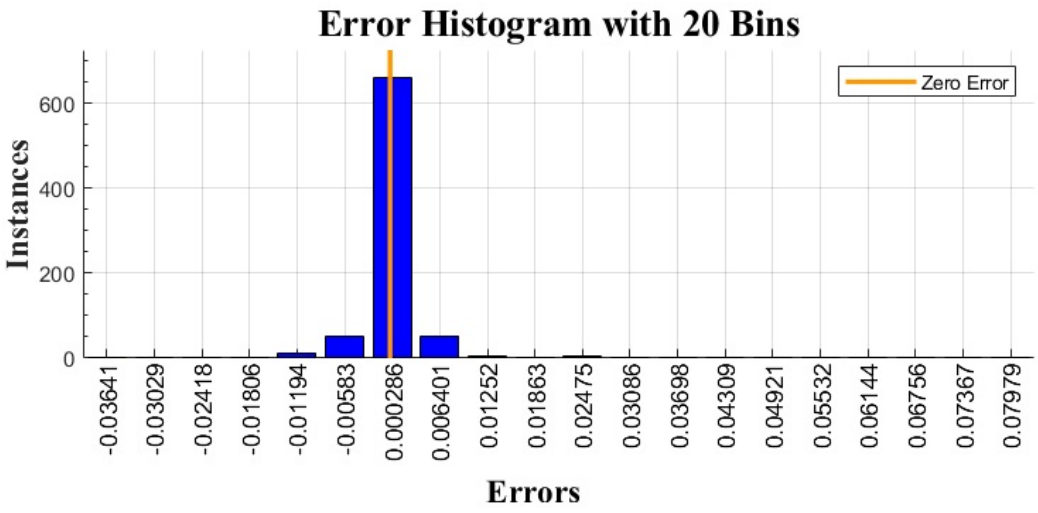
Figure 4.13 a. ATC estimated using ANN estimator for Transaction from Area 1 to Area 2 of IEEE 30-Bus system; b. Error Histogram of ATC estimator using reduced features for IEEE 30 Bus System

4.7.4 Modified IEEE 118 bus test system:

The IEEE 118 bus test has been modified by taking the long-term line limit as (2×base case power flow) and emergency rating as (5× base case power flow). Further, for the purpose of analysis, two different areas (Area1) and (Area2) comprising of the buses given in Table 4.5 are formed. The ATC estimation for transactions from Area1 as source and Area2 as sink have been performed using the proposed method.



(a) Estimated ATC



(b) Error Histogram

Figure 4.14 a. ATC estimated using ANN estimator for Transaction from Area 1 to Area 2 of IEEE 118-Bus system; b. Error Histogram of ATC estimator using reduced features for IEEE 118 Bus System.

The yearly loading factors given in [143] have been used to generate quasi-static data for the IEEE 118 BUS TEST system. The 118 Bus system being a large test case with a larger number of PMU's (n_p is 33). The size of the input to the ANN increased significantly. Therefore, a feature extraction and selection method has been adopted so as to adequately train the ANN estimator. The input provided by the PPMU through LSE is (608×1), which is reduced to (33×1). The estimated ATC by the ANN estimator and its performance in terms of error has been given in Figure 4.14(a) and Figure 4.14(b).

4.7.5 Performance of ANN based estimator:

Comparative analysis of the developed estimator for different test system have been illustrated in Table 4.10. Here the error has been expressed in terms of Mean Absolute Error(MAE), Mean Squared Error (MSE), Sum of Absolute Errors (SAE) and Sum of Squared Errors (SSE). The sample size for obtaining the performance has been taken as 1000.

Table 4.10 Performance of ANN estimator.

S. N	Indices	Test Case					
		IEEE 24 Bus		IEEE 30 Bus		IEEE 118 Bus	
		Before FE	After FE	Before FE	After FE	Before FE	After FE
1	MPAE	$2.10e^{-8}$	$2.54e^{-4}$	$7.42e^{-4}$	$4.20e^{-3}$	$1.76e^{-6}$	$2.4e^{-3}$
2	MSE	$1.51e^{-15}$	$9.98e^{-7}$	$3.87e^{-6}$	$5.47e^{-5}$	$1.21e^{-7}$	$3.7e^{-5}$
3	SAE	$1.55e^{-3}$	0.11	0.59	3.33	0.14	1.9
4	SSE	$6.81e^{13}$	$4.94e^{-4}$	$3.10e^{-3}$	$4.38e^{-2}$	$9.65e^{-5}$	$2.90e^{-2}$

Table 4.11 Training Time and Data Generation Time

Test System	Data Generation Time (sec)		ANN Training Time (sec)	
	Duration	Time	Without FE	With FE
24 Bus	8760	43800.51	13.21	9.54
30 Bus	8760	70080.43	68.03	47.5
118 Bus	8760	91980.29	94.47	72.7

Table 4.12 Offline Training and Testing Time.

Without Feature Extraction					
Test System	**Training Time**	**Estimator Time**	**LSE Time**	**Total Time**	
24 Bus	13.21	0.00685	0.000198	0.00705	
30 Bus	68.03	0.0078	0.000299	0.0081	
118 Bus	94.47	0.00886	0.000301	0.00916	
With Feature Extraction					
Test System	**Training Time**	**FE time**	**Estimator Time**	**LSE Time**	**Total Time**
24 Bus	9.54	0.000471	0.0058	0.000198	0.006
30 Bus	47.5	0.000438	0.0077	0.000299	0.008
118 Bus	72.7	0.00046	0.00718	0.000301	0.00748

It can be observed that after using the reduced feature set as input to the estimator, the error metrics deteriorated significantly but they are still under the acceptable limits. It can be observed that after using the reduced feature set as input to the estimator, the error metrics deteriorated significantly, but they are still under the acceptable limits. It has been noted that for IEEE 118 bus test system, the error is more than the other two cases; thus, it can be said that the overall performance of the ATC estimator degrades with an increase in the complexity of the system. Nevertheless, the error falls well within the range of $\mathbf{10^{-3}}\ \boldsymbol{to}\ \mathbf{10^{-5}}$ which can be very well accepted for ATC estimation application. The overall time elapsed in data generation for ANN training, in ANN

training, and ATC estimation has been given in Table 4.11 and Table 4.12. The time elapsed has been acquired for the solution obtained using the proposed method on MATLAB 2017b using the intel core i7 processor.

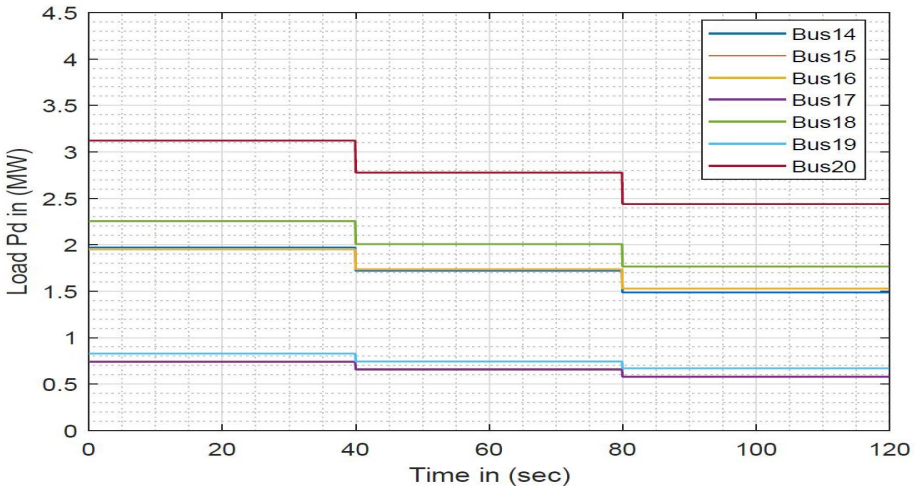
4.7.6 Real Time Authentication of IEEE 30 bus test system using RTDS:

The IEEE 30 bus test system has been modeled in RTDS software. The system has been subjected to different loading scenarios using the command signals issued from the control center through the GTNET SKT. The 'GTNET PMU's have been used to measure the relevant parameters. These PMU measurements have been acquired using IEEE C.37.118 protocol by '$OPENECA$' software. These measurements have been sent to the control center, where the proposed method has been utilized to estimate the ATC.

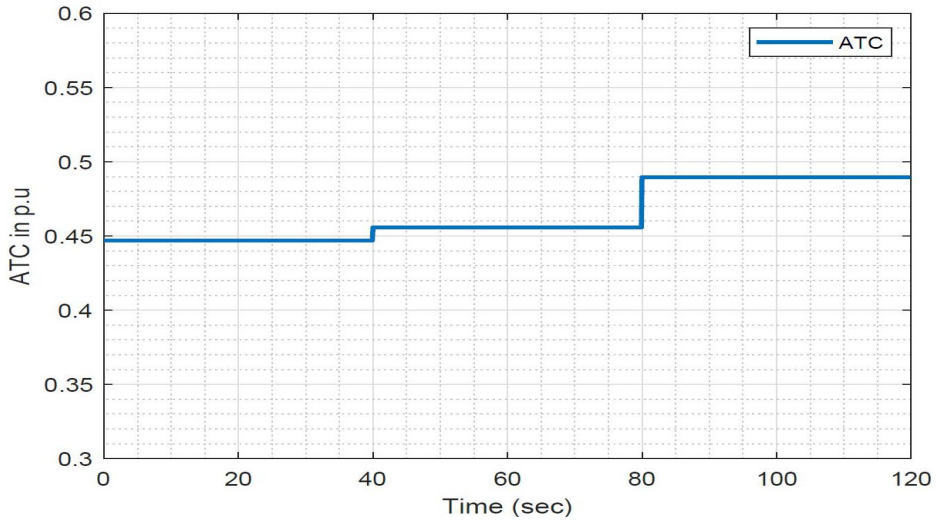
The overall time elapsed in various processes during the ATC estimation has been given in Table 4.13. The result of estimated ATC values and corresponding loading scenario has been shown in figures Figure 4.15(a) and Figure 4.15(b). The loading pattern has been changed at an interval of 40 seconds. The corresponding ATC values have been shown in Figure 4.15(b). It can be observed from these results that the ATC values increase with the decrease in load.

Table 4.13 Real Time Implementation on RTDS.

Test System	Communication time (sec)	LSE Time (sec)	FE Time (sec)	Estimation Time (sec)	Total Time (sec)
IEEE 30 Bus	0.05	0.0078	0.000299	0.0077	0.0658



(a) Loading of buses in area A2



(b) ATC estimated using proposed ATC estimator

Figure 4.15 Estimated ATC and loading of buses in area A2 during RTDS simulation.

Table 4.14 Comparison of Proposed Method with the Referred Methods

System	Source	Sink	Referred Method		Proposed Method
			SATC(MW)	SATC(MW)	ATC (MW)
30 Bus [158]	2	18	3.7	0.95	3.86
	2	23	3.6	1.15	3.95
	2	30	2.09	0.63	2.0974
39 [159]	36	21	-	7.3	20
145 Bus [158]	141-143	34,35	6.5	1.82	6.5478

4.7.7 ATC and Dynamic ATC:

The dynamic ATC for system for any transaction can be considered as an approximation of the static ATC [158]. A comparative analysis of static and dynamic ATC has been done for different cases and reported in Table 4.14. It can be observed that the ATC obtained using the proposed method is three to four times the dynamic ATC.

4.8 CONCLUSION

In this chapter, a framework for real-time ATC estimation using ANN has been presented with its application on IEEE 24-bus, IEEE 30-bus, and IEEE 118-bus test systems. An ATC optimization engine employing the method proposed for estimation of ATC has been used for obtaining the ATC training data in the offline stage of real-time estimator development. PMU emulation algorithm for quasi-static analysis of power system (using PPMU) has been used to obtain the voltage and current states of the system quasi-statically. The information obtained from the PPMU's has been used as an input of the real-time ATC estimator. The results have been authenticated using a real-time digital simulator on IEEE 30 bus test system.