Chapter 3

PSEUDO PMU FOR QUASI-STATIC ANALYSIS OF POWER SYSTEM AND NEAR REAL-TIME LOAD FORECASTING

3.1 INTRODUCTION

The Phasor Measurement Units (PMU) are being integrated into the power grids for enhanced monitoring and control of the power system. PMU's data are useful for post-mortem analysis and development of real-time control of the Power system. In this chapter, a Pseudo-PMU (PPMU) is proposed for the quasi-static analysis of the Power System. A methodology is presented for the emulation of PMU and the storage layout of data obtained from the emulation. Technique enabling the emulation of the PMU's specifically for Static Security Analysis will be of great use for analysis of the system by the researchers and scientists working in the power sector. For the optimal location of PMU's, pseudo-PMU is used, and quasi-static analysis of IEEE 24-bus RTS and IEEE 14 bus systems is carried out in MATLAB environment. The results illustrate the efficacy of the proposed method for quasi-static analysis of the power system.

This chapter also presents a methodology for near-real-time load forecasting of the power system. Near real-time load, forecasting is the prediction of power system load for the duration of the next few hours. It is becoming an important topic because of the process of deregulation and the introduction of competition in the wholesale electricity markets. Further spinning reserves, unit commitment, contingency analysis, and economic load dispatch functions of the energy management system rely heavily on near real-time load forecasts. Using Support Vector Regression (SVR) as a base, we present our other method for near-real-time forecasting. Different models are created for holidays

and weekends, which are compared with the SVR method. A self-learning weekly window is applied, which automatically trains the models for three weeks and predicts the load for the next week. Performance evaluation of the models has been accomplished by training and testing them for data provided by the Rajasthan state load dispatch center using evaluation metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). It has been found that the proposed models significantly outperform the traditional SVR model by a lesser MAPE.

3.2 PSEUDO PMU AND PHASOR MEASUREMENT UNITS

Synchrophasors are phasors taken at defined instants across the spectrum of existing phasor quantities in the universe of interest. From the view of power systems, the universe of interest would be the embodiment of voltage and current phasors of the power system network under consideration. Phasors are measured at a particular instant of time; therefore, they could be referred to as a scan of the system, providing an account of events in the first hand as a first information report. The availability of such a scan of the power system enhances the observability and adds to the controllability of the system. A precise clock using the Global Positioning System is used for defining the time instants for the phasor measurement.

Deployment of PMU units employed for synchrophasor measurements is being done at a large scale in power grids across the world. Synchrophasor technology is realized by using different components such as Phasor Measurement Units, Phasor Data Concentrator, Historian, Communication network, real-time visualization, and offline toolboxes [139]. The infrastructure can broadly classify into three different layers: 1) Measurement Layer 2) Data Collection Layer 3) Application Layer. The measurement

layer comprises PMU's deployed across the grid. Data Collection Layer is responsible for collecting synchrophasor data from PMU's deployed across the grid in Phasor Data Concentrator through the communication network. The application layer inculcates the tools and software required for analyzing the data obtained from the PMU units.

The Pseudo PMU presented here is concerned with the implementation of the measurement unit for the development and testing of different applications for the Application layer. For analyzing the system behavior, the Pseudo PMU takes the measurements for every instant as and when the scenarios are quasi-statically emulated. Therefore, the requirement for time synchronization vanishes as the measurements taken by the Pseudo PMU will be in synchronism with the quasi-static nature of the analysis.

3.3 METHODOLOGY FOR IMPLEMENTING PSEUDO PMU

A Pseudo-PMU (PPMU) emulation algorithm that captures the $voltage\ magnitude$, $voltage\ angle$, $branch\ current\ magnitude$, and $branch\ current\ angles$ at the PMU location is developed. For system analysis, the operating scenario under emulation would be a set of loading patterns, fault cases, and contingency cases. This set of operating scenarios will be used as input data for emulation. In this work, a set of loading patterns is considered as an example of an operating scenario. The loading pattern can be a forecasted loading pattern or predefined loading pattern for which the analysis has to be done. The overview of the procedure is illustrated in Figure 3.1. Different stages in the block diagram are explained in the following section.

3.3.1 Input Data

The set of operating scenarios are used as the input data. The data (loading and generating pattern) can be generated as per the requirements of the study for the power

system network under consideration. For instance, in ATC assessment, the loading pattern will be such that in the sink area, the loading will be increased from the base case, and in the source area, an equivalent increment of generation has to be done. The next stage input would be the data that has been defined, whereas the solution is carried out for the nonlinear load flow equation. The following subsection illustrates the different stages.

3.3.2 Solution of Power Flow Equations

The solution to the nonlinear power flow equations is obtained by using any suitable technique such as Fast Decoupled Load Flow (FDLF), DC load flow, Optimal Power Flow, etc.

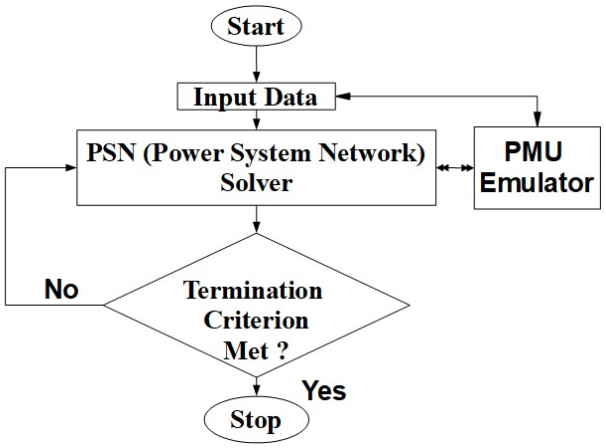


Figure 3.1 Block Diagram representation of proposed methodology.

Here in this work, the Newton-Raphson load flow algorithm has been used for solving the power flow equations. Nonlinear equations to be solved are as given:

$$P_i = \sum_{j=1}^{N} V_i V_j Y_{i,j} \cos(\theta_{ij} - \delta_i + \delta_j) \qquad 3.1$$

$$Q_i = -\sum_{j=1}^{N} V_i V_j Y_{i,j} \sin(\theta_{ij} - \delta_i + \delta_j) \quad 3.2$$

Subjected to the following constraints,

$$\sum_{i=1}^{N} Pg_i + \sum_{i=1}^{N} Pd_i + P_{loss} = 0 \quad 3.3$$

$$V_i^{min} \leq V_i \leq V_i^{max} \quad 3.4$$

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max} \quad 3.5$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \quad 3.6$$

$$P_{ij} \leq P_{ij}^{max} \quad 3.7$$

Here,

P_i, Q_i are the active and reactive power injections at bus i.

P_{gi}, Q_{gi} are power generated and load demand at bus i.

$V_i, V_i^{min}, V_i^{max}$ are voltage at i^{th} bus and it's minimum and maximum limits.

$Q_{gi}, Q_{gi}^{min}, Q_{gi}^{max}$ are reactive power generation of i^{th} generator and it's maximum and minimum limits.

P_{ij}, P_{ij}^{max} are the line flow through line i, j, and its maximum limit.

δ, θ_{ij} is the voltage angle at bus i, and impedance angle of line i, j.

Nonlinear equations (3.1) and (3.2) are solved by ensuring that certain equality and inequality conditions are met. These conditions are 1) Total power generated should be equal to the total power demand and total power loss (P_{loss}) in the system (3.3). 2) The voltage of all the buses should be maintained within their maximum and minimum limits

eq. (3.4). 3) The maximum and minimum reactive power and active power limits of the generators should be met (3.5) and (3.6). 4) The line flows should be within their limits (2.58). Newton-Raphson based load flow technique is widely used for solving these power flow equations. In the proposed method, any other suitable technique with some additional objective (such as optimal power flow) can also be used.

3.3.3 Pseudo-PMU Emulation

The Pseudo PMU emulation has been achieved by extracting the $voltage\ magnitude$, $voltage\ angle$, $current\ magnitude$, and $current\ angle$ as different input loading patterns are iteratively subjected to the load flow stage of the proposed method. The PMU emulation stage takes the data regarding the number of PMU installed and the optimal location of PMU's that is provided in the input data set. The stack of cells is formed from the PMU emulation stage, which is equal to the number of PMU installed in the system. Each cell is divided into four sub-cells. The '$From\ Bus\ data$' is the lines that are considered to be originating from the bus at which the PMU is placed is stored in the first sub-cell. The $'To\ Bus\ data'$ is the lines that are considered to be terminating at the bus at which the PMU is placed and are stored in the second sub-cell. The bus/node voltage magnitude and voltage angle are stored in the third sub-cell. The fourth subcell contains the data of $branch\ current\ magnitude$ and $branch\ current\ angles$.

3.3.4 Voltage and current Extraction

Extraction of voltage and current at any desired bus is quasi-statically achieved as the varying load is subjected to the system. The Newton-Raphson Jacobian matrix is given by the (3.8) [140]:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = J \begin{bmatrix} \Delta\delta \\ \frac{\Delta V}{V} \end{bmatrix} \quad 3.8$$

Where

$$j = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \cdots & \frac{\partial P}{\partial V} \\ \vdots & \ddots & \vdots \\ \frac{\partial Q}{\partial \delta} & \cdots & \frac{\partial Q}{\partial V} \end{bmatrix} \quad 3.9$$

Consider that a system is entirely observable after employing $'m'$ number of PMU's optimally placed at $'m'$ locations in the system. Now the voltage of those $'m'$ buses is taken and stored in a matrix of size $m \times n'$. Here n' is changing as the system is quasi-statically subjected to the given loading pattern. The voltages at the desired buses will be given by

$$\begin{bmatrix} \delta_i^k \\ V_i^k \end{bmatrix} = \begin{bmatrix} \delta_i^{k-1} \\ V_i^{k-1} \end{bmatrix} + \begin{bmatrix} \Delta\delta_i^k \\ \Delta V_i^k \end{bmatrix} \quad 3.10$$

Here k is the k^{th} loading to which the system under study is subjected, and i is the bus number. Now the system's connectivity can be represented in terms of the admittance matrix Y or impedance matrix Z. We will be using the admittance matrix for obtaining the connectivity as it is readily available since it has to be pre-computed for carrying out the load flow.

$$[Y] = \begin{bmatrix} Y_{i1} & \cdots & Y_n \\ \vdots & \ddots & \vdots \\ Y_{m1} & \cdots & Y_{mn} \end{bmatrix} \quad 3.11$$

Where Y_{ii} is the diagonal element of the admittance matrix and is equal to the sum of admittance connected at bus i and the sum of admittance of all the branches connected to bus i.

$$Y_{ii} = y_{ii} + y_{i1} + \cdots + y_{in} \quad 3.12$$

$Y_{ij}$is the off-diagonal element and is equal to the negative of the branch admittance connected between bus i and j.

$$Y_{ij} = -y_{ij} \quad 3.13$$

y_{ij} being the admittance of branch between i and j. And is given by the sum of the admittances at the bus i and half-line chargings of the lines connected to the bus i. The injected current at node i is given by

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{i1} & \cdots & Y_n \\ \vdots & \ddots & \vdots \\ Y_{n1} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \quad 3.14$$

$$[I] = [Y][V] \quad 3.15$$

The current injected by

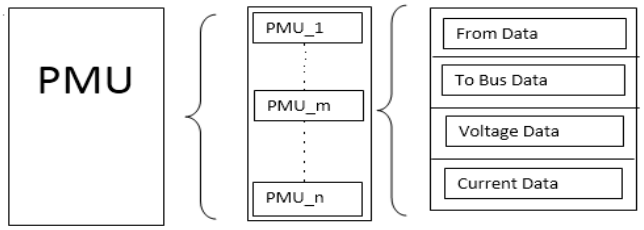
$$I_i = I_1 + I_2 + \cdots I_m \quad 3.16$$

Here m is the number of branches connected at bus i. Also

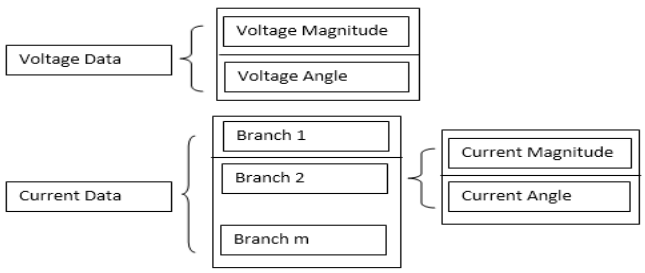
$$I_{ij} = y_{ij}^{l} V_i + y_{ij} V_j \qquad 3.17$$

Here y_{ij}^{l} component of y_{ij} on account of line $i \rightarrow j$.

PMU gives the current in the branches which directly connected to the nodes and node voltages at which the PMU's are placed. Diagrammatic representation of the data storage method for emulated PMU is given in Figure 3.2. The voltage data cell is illustrated to be divided into two cells: one containing voltage magnitude and the other containing the voltage angle in Fig.3. Similarly, $current$ data cell is further divided into cells depending upon the no of branches that are connected to the PMU bus. These sub-cells are further divided into two cells, one containing the $branch\ current\ angle$ and the other containing $branch\ current\ magnitude$.

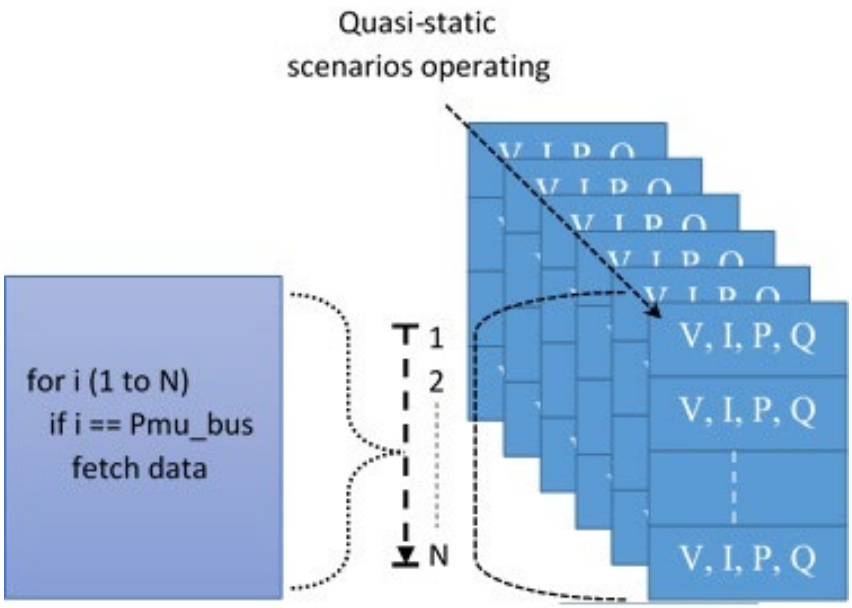


(a)

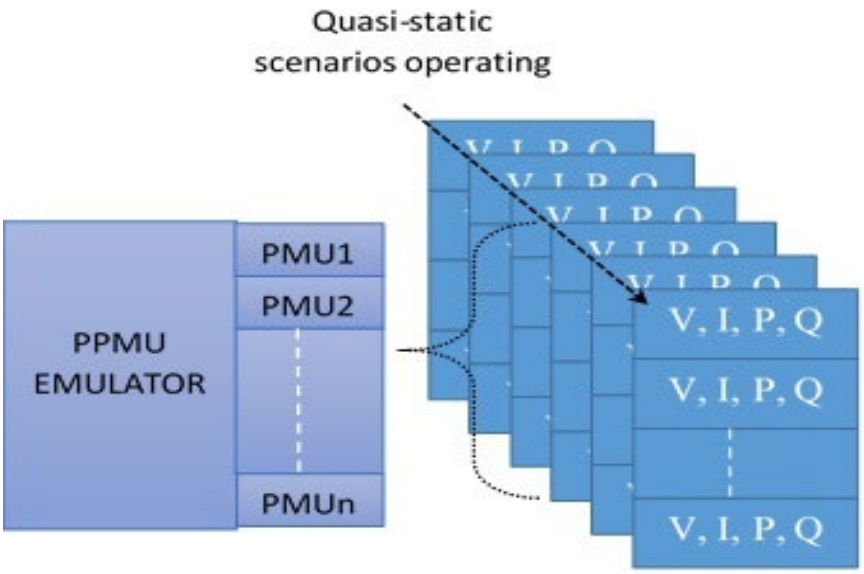


(b)

Figure 3.2 PMU DATA STORAGE LAYOUT



(a) Iterative Algorithm



(b) PPMU Algorithm

Figure 3.3 Schematic comparison of iterative and PPMU algorithm.

Further, if PPMU is not employed, then an iterative algorithm that sweeps across the operating situations (probable/feasible solutions) and fetches the data if the $bus\ id$ is

the same as the PMU location bus would have to be utilized. This happens to be a time-consuming process, and the time taken would increase with the increase in complexity (number of buses) of the system. The schematic comparison of the Iterative method and PPMU algorithm is shown in Figure 3.3.

3.3.5 PMU PLACEMENT

The optimal location of the PMU's can be obtained by using any well-established PMU placement method. Here an ILP based PMU placement technique has been utilized for determining the optimal location of the PMU's in the system. The optimal PMU placement formulation based topological observability method finds an optimal set of PMUs such that a bus in the power system must be reached at least once by the PMUs.

The optimal placement of PMUs for an N bus system is formulated as follows [141]:

$$Min \sum_{i=1}^{N} w_i x_i \tag{3.18}$$

$$AX \geq b \tag{3.19}$$

$$X = [x_1, x_2, \dots, x_N]^T \tag{3.20}$$

Where w_i is weight factor accounting for the cost of installed PMU at the bus i; N is the total number of system buses, A is the connectivity matrix of the power system, X is a binary variable vector having elements x_i define the possibility of PMUs on a bus i whose entries are defined by Equation (3.20), and AX is a vector such that its entries are non-zero if the corresponding bus voltage is observable using the given measurement set and according to observability rules mentioned above. It provides full network

observability while minimizing the total installation cost of the PMUs; otherwise, its entries are zero.

$$x_i = \begin{cases} 1 & if\ a\ PMU\ is\ needed\ at\ bus\ i \\ 0 & otherwise \end{cases} \quad 3.21$$

The elements of A are defined as follows:

$$a_{ij} = \begin{cases} 1 & if\ i = j \\ 1 & if\ i\ and\ j\ are\ connected \\ 0 & otherwise \end{cases} \quad 3.22$$

b is a vector whose entries are all ones, as shown in Equation (2.58).

$$b = [1\ 1\ 1\ \ldots 1]^T \quad 3.23$$

Eq. (3.24) represents the expression of maximum observability from which we can check the observability (O_{total}) of each bus of the system [141]:

$$O_{total} = \sum_{k=1}^{p} A_{l_{oc}}(k) \quad 3.24$$

Where p is the total optimal number of PMUs, A is the connectivity matrix, and the location of PMUs at the power system buses indicated by l_{oc}. The concept of zero injection buses (ZIB) has been used to reduce the optimal number of PMUs further. ZIB is that bus at which neither load nor generation is connected. Detail expressions of the ZIB have been discussed in [142]. The selections of the buses in normal case and the selections of the neighbouring buses of the ZIB have been updated to get the multiple results via ILP.

3.3.6 CASE STUDY

The method is tested on IEEE 24 bus RTS system, IEEE 30 bus test system, and IEEE 118 bus test system. The hourly loading pattern is taken and given as input for illustration of the proposed Pseudo PMU emulation. The ILP method discussed in section 3.3.5 is used to obtain the candid solution for the placement of the Pseudo PMU's. The proposed method attempts to optimally place the PMU by maximizing the observability of the system. The results obtained of PMU placement in IEEE 24 bus RTS system are given in Table 3.1, and the results are compared with those obtained in [141].

Table 3.1 OPTIMAL PLACEMENT OF PMU IN IEEE 24-BUS RTS SYSTEM.

Method	**Without ZIB**			**With ZIB**		
	No. of PMUs	**Location of PMUs**	**Obs.**	**No. of PMUs**	**Location of PMUs**	**Obs.**
Proposed	7	2, 3, 7, 10, 16, 21, 23	29	6	1, 2, 8, 14, 20, 21	22
		2, 3, 8, 10, 16, 21, 23	**31**		1, 2, 8, 14, 20, 21	25
					2, 8, 10, 15, 20, 21	25
Ref. [141]	7	2, 3, 7, 10, 16, 21, 23	29	6	1, 2, 8, 16, 18, 23	24

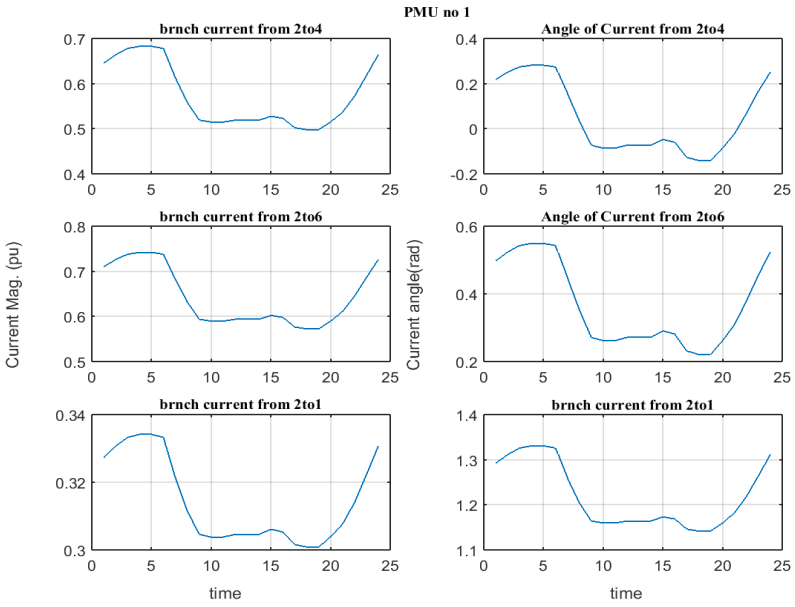


Figure 3.4 Plot of current obtained by PMU 1 at bus 2.

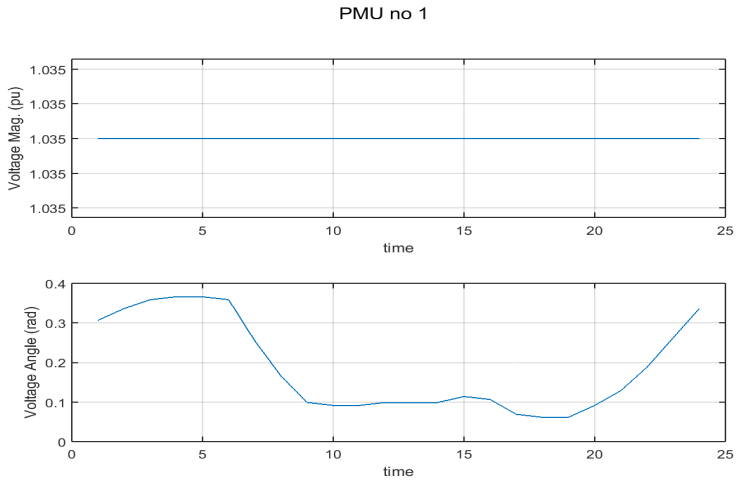


Figure 3.5 Plot of Voltage obtained by PMU 1 at bus 2.

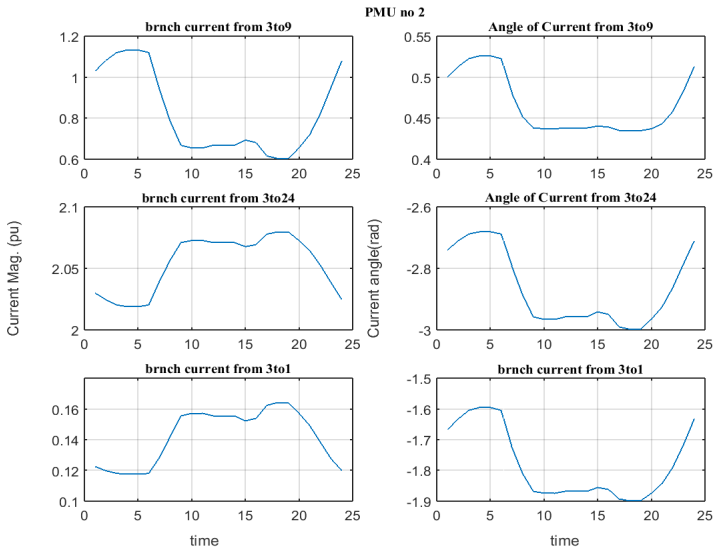


Figure 3.6 Plot of current obtained by PMU 2 at bus 3.

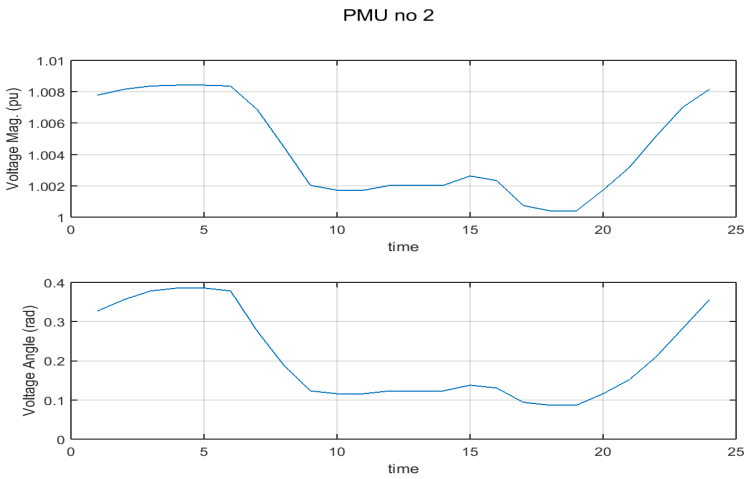


Figure 3.7 Plot of Voltage obtained by PMU 2 at bus 3

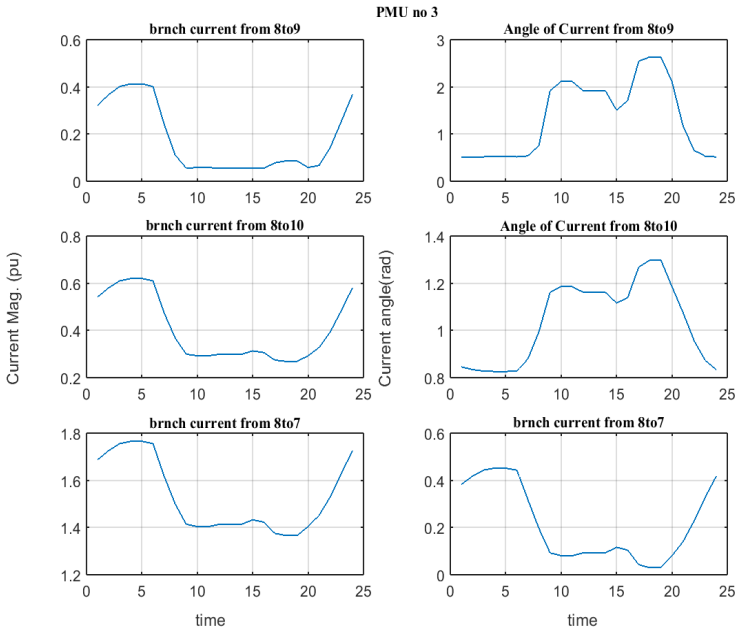


Figure 3.8 Plot of current obtained by PMU 3 at bus 8

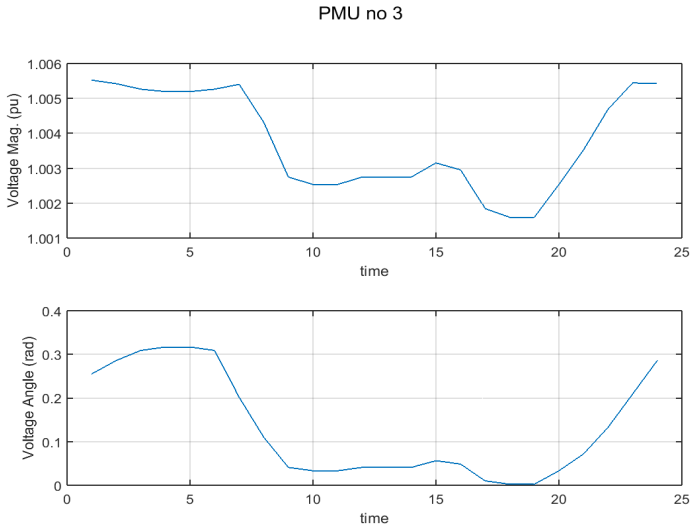


Figure 3.9 Plot of Voltage obtained by PMU 3 at bus 8.

The location for which the observability is maximum has been considered as the appropriate location for PMU placement, and the Pseudo PMU's are placed at this location. The data for the RTS test system has been taken from [143].

The plots obtained by using this method are given in Figure 3.4-Figure 3.9. The figures have $PMU\ number$ as their title; this title corresponds to the PMU whose plot is being shown in the figure. Figure 3.4, Figure 3.6, Figure 3.8 gives the plot of the branch currents, whereas Figure 3.5, Figure 3.7, Figure 3.9 gives the node voltage plot. The plots are obtained for 24 hours of load variation.

Also, the comparative results for the iterative algorithm method and PPMU based emulation method for different test cases using MATLAB 17b; running on an intel core i7 processor, has been shown in Table 3.2.

Table 3.2 Comparison of Iterative and PPMU algorithm using MATLAB 17b on intel i7 processor.

S. N	Case	Time Taken	
		Iterative Algorithm	**PPMU Algorithm**
1	IEEE 24 BUS RTS	0.0011	0.0007
2	IEEE 30 BUS System	0.0012	0.0006
3	IEEE 118 BUS System	0.0043	0.0015

3.4 NEAR REAL-TIME LOAD FORECASTING

Near real-time load forecasting can be considered as a more precise form of short-term load forecasting that too for a smaller duration. In this world of commercialization, it is a major tool used by power utilities to maximize profit. It is becoming an increasingly important topic not only from the point of view of power generation but also from the financial perspective. It is an important process that can increase the efficiency and revenues for the electrical generating and distribution companies [144]. Since Electrical energy is an energy that cannot be stored at a large scale economically and efficiently, an equilibrium has to be always maintained between load demand and supply. Disturbance in this equilibrium can adversely affect equipment at the generation side as well as household equipment and grid stability. Near real-time load forecasting is used to supply the necessary information for system management of day to day operation, unit commitment, and estimation of load flow, grid operation, and power plant scheduling. With proper load forecasting, we can have optimum use of spinning reserve and online generators. The main contribution of this section is:

- To develop an algorithm that reduces the differences between actual and forecasted values with MAPE not more than 1.5%.
- To implement a moving window on all the developed models that automatically trains itself as soon as the data is available.
- An attempt has been made to find the most unstable time of the day using.

Here three different techniques are proposed and compared with a standard SVR (Support Vector Regression-based) based method. Among all the three developed models, the one with optimized parameters gives remarkable accuracy and outperforms

the SVR model. All the models are trained with varying sets of historical data and tested for different conditions. The performance of the SVR model is directly proportional to the amount of data available for training [145]. The accuracy of the prediction model is measured by RMSE and MAPE. But to measure the error for each instant evaluation parameter MAE (Mean absolute error) is used to get the better inside of the picture.

3.4.1 Data conditioning and pre-processing

The data required for near-real-time load forecasting would comprise historical load data, weather data (typically temperature), holidays, and festivals. The raw data comprises invalid entries, missing entries, and non-synchronized data.

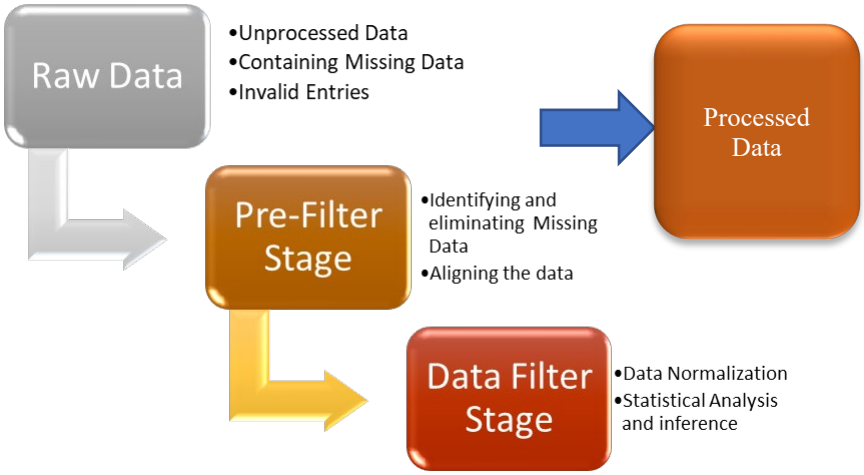


Figure 3.10 Schematic representation of data pre-processing stage.

3.4.2 SVR based method for load forecasting.

The raw data thus needs to be conditioned and pre-processed to make it apt for the near real-time load forecasting. The schematic representation of the overall process has been inked in Figure 3.10. Support Vector Machine was formulated by Vapnik in 1995. It works on the principle of Structural Risk Minimization (SRM) instead of Empirical Risk Minimization (ERM), which is far better than the later one. Initially, SVM

was developed for classification problems, but very soon, it was extended for the regression domain [146]. SVR uses the soft margin technique to obtain the best generalization ability. The aim of SVR is to find the hyperplane in space [147]. It shows good performance despite insufficient training samples. Regression problems are solved with the help of an alternative loss function called $(\epsilon - loss)$ function and two slack variables.

Consider the problem of approximating the set of data,

$$A = \{(x1, y1), \ldots, (xl, yl)\}, \qquad x_i \in Rn, y_i \in R \tag{3.25}$$

With a linear function,

$$f(x) = < w, x > + b. \tag{3.26}$$

Where w is a weight vector, and b is called the bias term.

The optimal regression function is given by the minimum of the functional,

$$\begin{aligned} &Min\big(\Phi(w,\xi)\big) \\ &\phi(w,\xi) = 0.5\big||w|\big|^2 + \sigma \sum_{i=1}^{l} (\xi_i^+ + \xi_i^-) \end{aligned} \tag{3.27}$$

$$subjected\ to: \begin{cases} y_i - w^T\phi(x_i) - b \le \epsilon + \xi^+ \\ w^T\phi(x_i) + b - y_i \le \epsilon + \xi^- \\ \xi_i^+, \xi_i^- \ge 0; i = 1,2, \cdots l, \epsilon > 0 \end{cases} \tag{3.28}$$

where σ is a pre-specified value, and the slack variables representing upper and lower bounds on the outputs of the system are ξ^+, and ξ^-. Slack variables are the potential violations of the margin constraints [148]. $\phi(x)$ is the high-dimensional feature mapping to the feature space. The constrained minimization problem of (3.27) can be solved using any apt optimization tool.

3.4.3 Proposed Models

In the proposed method, the load data is arranged in a time-series format. As we can observe from Figure 3.11 that through time-series data format, we can get various feature vectors such as load at the previous hour of the day, previous day's current hour, previous week's current hour, etc., just by giving proper shifting in data set. Binary variable one is used for denoting workdays and weekends by zero. Holidays are also denoted by binary variable 1. Load at the weekend is generally lesser than load at working days.

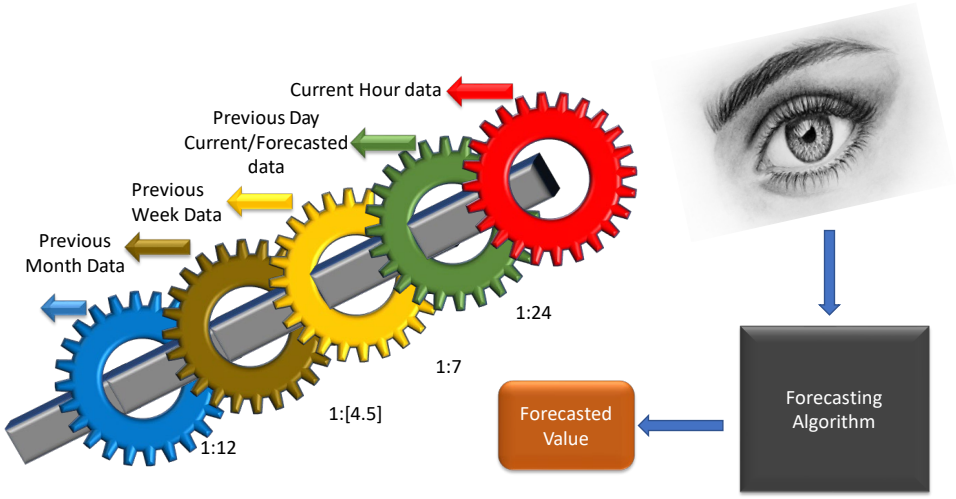


Figure 3.11 Schematic representation of input features.

The observer reads the data at an instant of time and feeds the same to the forecasting algorithm where the forecasted value is estimated. The estimation formulation (3.31) has been developed through the development of two developmental models (2.58) and (3.30); these are given as under

$$M3(f) = CHD + CHD * \frac{PHD - CHD}{\frac{PHD + CHD}{2}} \qquad 3.29$$

$$M2(f) = \text{CHD} + \left(0.75 * \left(\text{CHD} * \frac{\text{PHD} - \text{CHD}}{\frac{\text{PHD} + \text{CHD}}{2}}\right) + 0.125 * \left(\text{PDCHD} * \frac{\text{PDFHD} - \text{PDCHD}}{\frac{\text{PDFHD} + \text{PDCHD}}{2}}\right) + 0.125 * \left(\text{PWCHD} * \frac{\text{PWFHD} - \text{PWCHD}}{\frac{\text{PWFHD} + \text{PWCHD}}{2}}\right)\right) + \text{E} * \text{CHD} \quad 3.30$$

$$M1(f) = \text{CHD} + \left(\alpha * \left(\text{CHD} * \frac{\text{PHD} - \text{CHD}}{\frac{\text{PHD} + \text{CHD}}{2}}\right) + \beta * \left(\text{PDCHD} * \frac{\text{PDFHD} - \text{PDCHD}}{\frac{\text{PDFHD} + \text{PDCHD}}{2}}\right) + \gamma * \left(\text{PWCHD} * \frac{\text{PWFHD} - \text{PWCHD}}{\frac{\text{PWFHD} + \text{PWCHD}}{2}}\right)\right) + \text{E} * \text{CHD} \quad 3.31$$

In the above equations,

Where,

CHD is current hour data

PHD is previous fifteen-minute data

$PDCHD$ is previous day current hour data

$PDFHD$ is previous day forecasted hour data

$PWCHD$ is previous week current hour data

$PWFHD$ is previous week forecasted hour data

E is error

α, β and γ are parameters to be optimized through training.

Load forecasting is a nonlinear process. Hence Model 1 ($M1$) uses $fmincon$ as the optimization technique. It is an optimization function used for nonlinear programming.

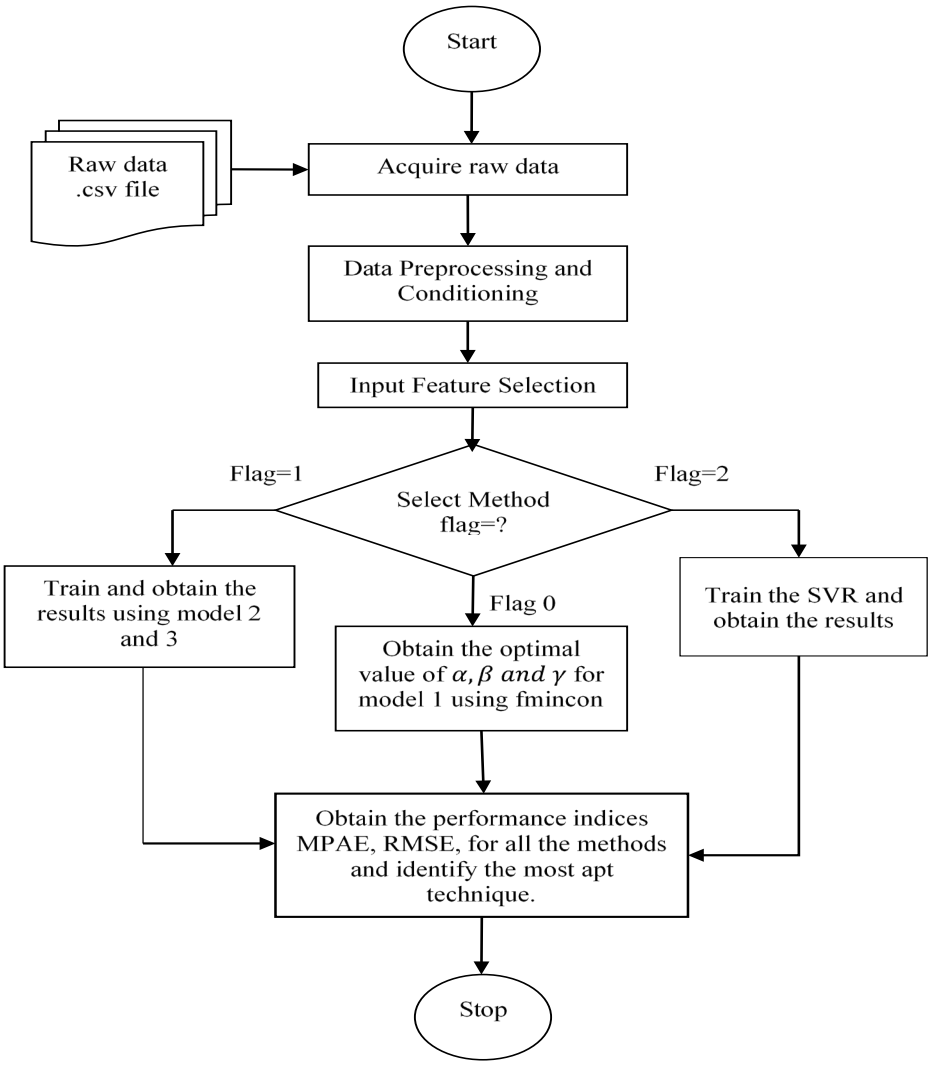


Figure 3.12 Flow chart showing the implementation of the proposed method.

Fmincon finds a constrained minimum of a scalar function of several variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming. *Fmincon* uses (Sequential Quadratic Programming) SQP algorithm for the optimization. SQP is an iterative method for constrained nonlinear optimization. The flowchart representing the various steps involved in the forecasting process has been illustrated in Figure 3.12.

3.4.4 Evaluation and Test Results.

The proposed models and the SVR model are trained using two months of data from Jan 21 to Mar 21 and tested for the next upcoming weekday, weekend, and holiday. The plot of results obtained using various models has been inked in figures Figure 3.13 to Figure 3.19. It has been observed that the error for the weekends is higher than the weekdays. Further, the developed models give better prediction accuracy than the SVR model in the case of holidays. Overview of the model accuracy for four different days is presented in Table 3.3, and corresponding plots are drawn.

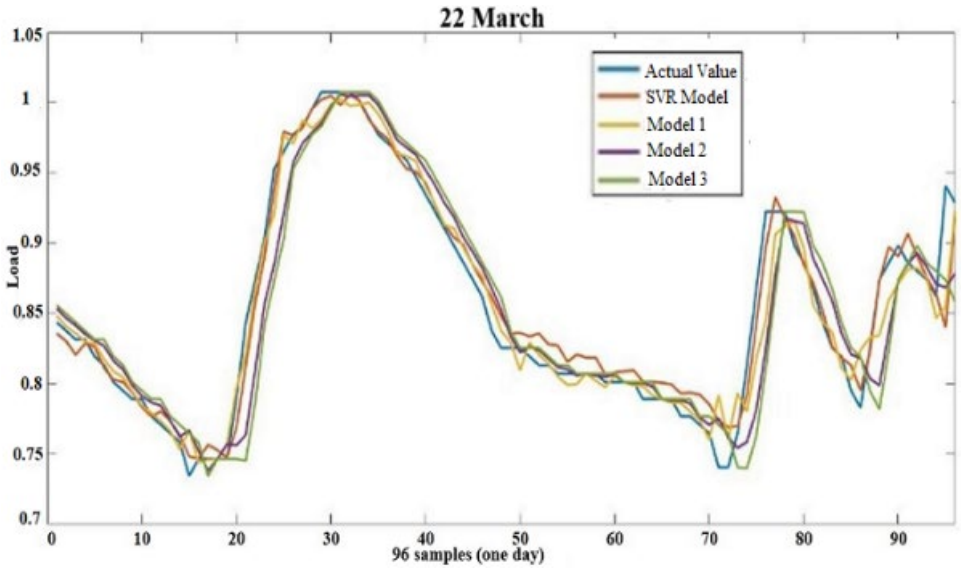


Figure 3.13 Plot for Mar 22, 2017 (weekday)

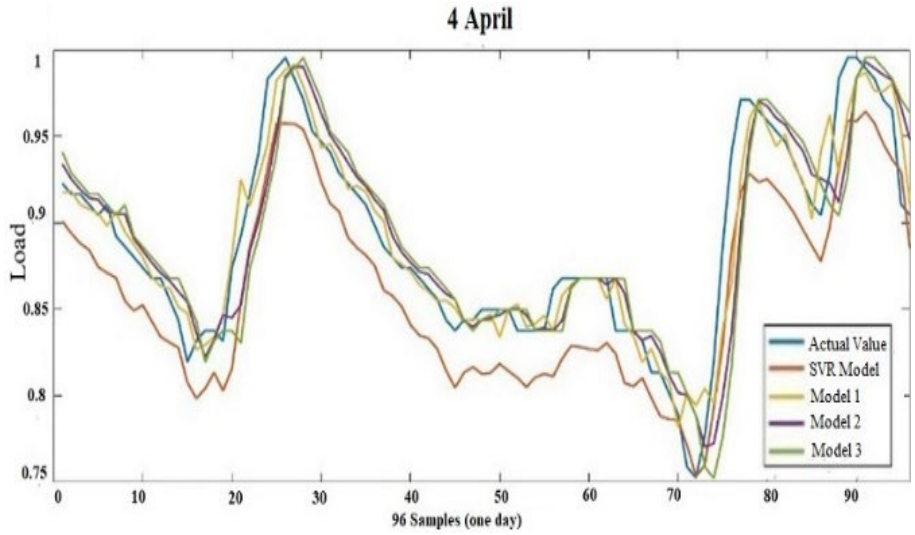


Figure 3.14 Plot for Apr 4 (Holiday).

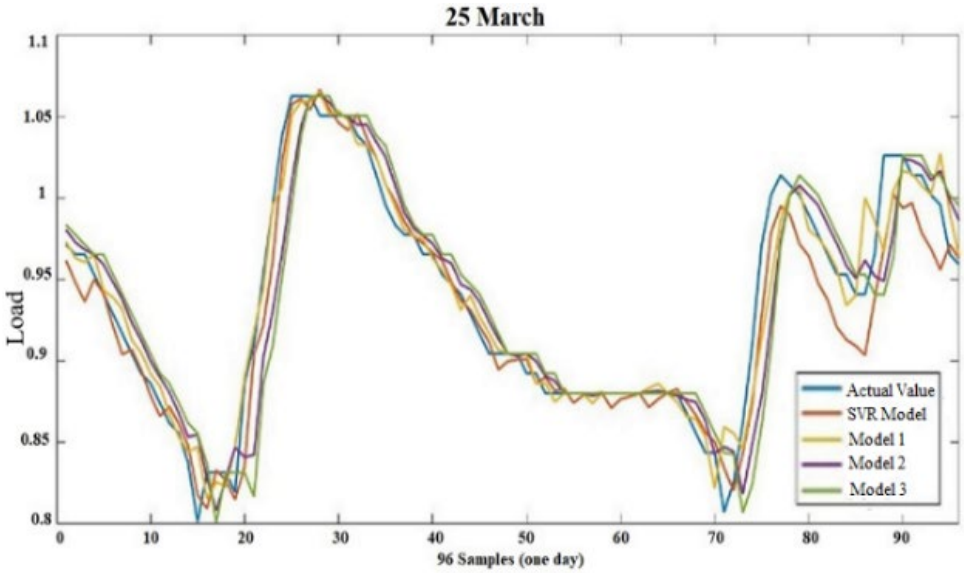


Figure 3.15 Plot for Mar 25, 2017 (Saturday).

The load trends of different festivals are different, and in most cases, festivals' load is lower than workdays' load [149]. To find the most disturbing time of a day, the SVR model is tested for four different time slots of a day, as shown in Table 3.4. By finding the most disturbing slot, we can improve integrity and security by paying special attention to that particular time of the day. The model is tested for different time slots of six days.

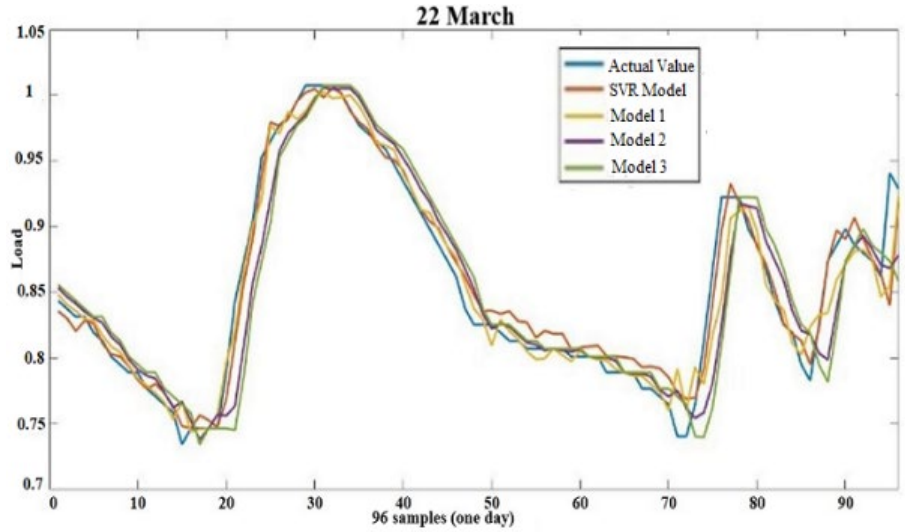


Figure 3.16 Plot for Mar 22, 2017 (weekday).

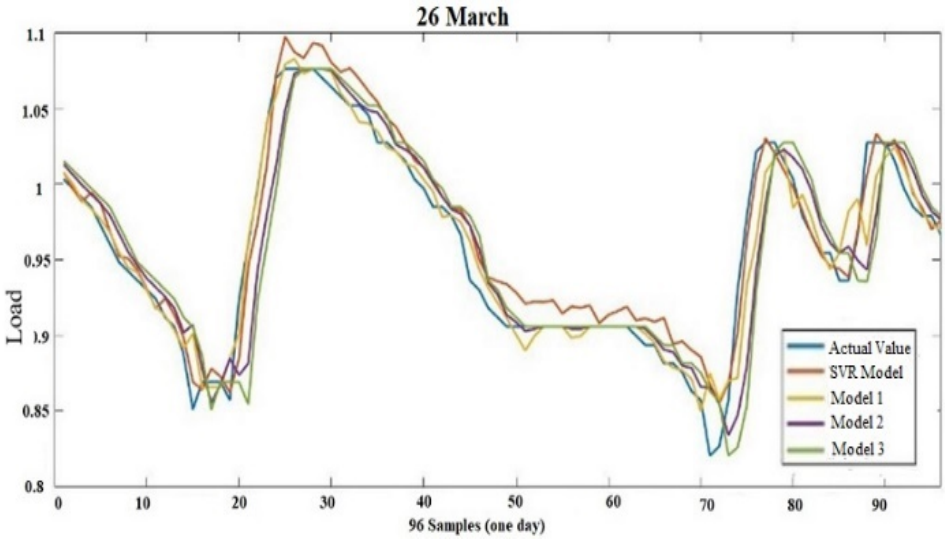


Figure 3.17 Plot for Mar 25, 2017 (Sunday).

It is observed from Table 3.4 that the most unstable time of the day is from 6 pm to 9 pm. The evening time is most unstable, maybe because for many people it is the recreational time of the day, closing off some organizations as well as a decrease in sunlight.

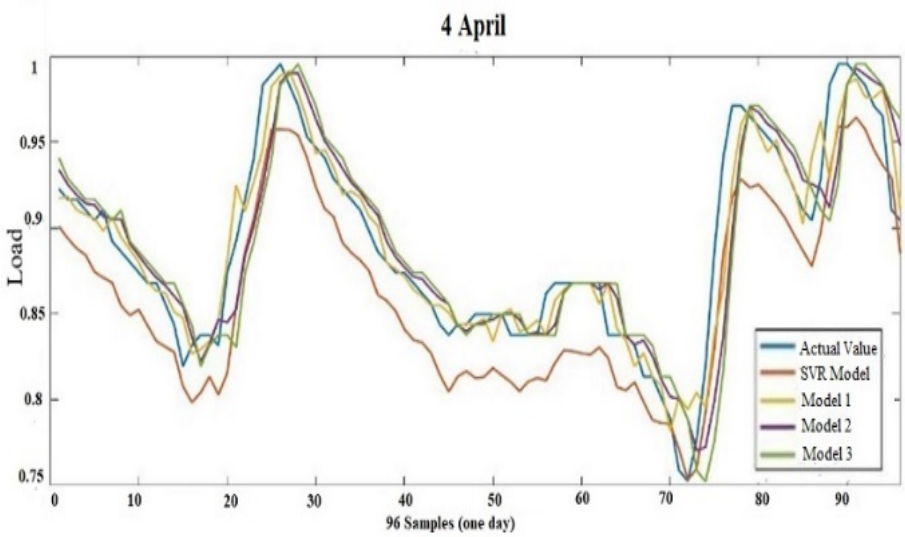


Figure 3.18 Plot for Apr 4 (Holiday)

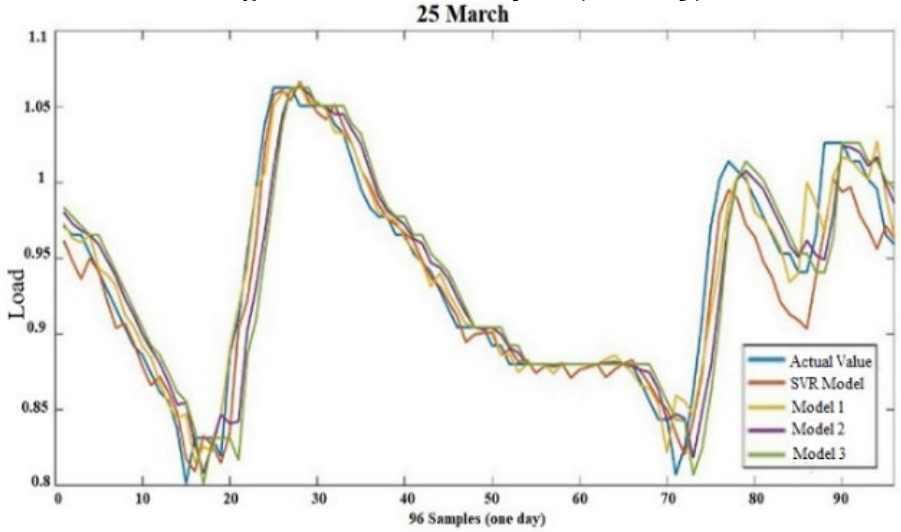


Figure 3.19 Plot for Mar 25, 2017 (Saturday).

Table 3.3 COMPARISION OF ERRORS FOR DIFFERENT TYPES OF DAYS.

Day	Date	SVR Model		Model 1		Model 2		Model 3	
		RMSE (%)	MAPE	RMSE (%)	MAPE	RMSE (%)	MAPE	RMSE (%)	MAPE
Week day	22nd March	1.22	1.159	1.87	1.3646	2.84	2.2724	3.47	2.7999
Weekend	25th March	1.98	1.495	1.76	1.178	2.72	1.9484	3.34	2.3945
Weekend	26th March	1.62	1.502	1.76	1.1741	2.76	1.976	3.4	2.4554
Holiday	4th April	3.16	3.327	1.85	1.3509	2.62	1.9828	3.19	2.4888

As this time of the day experiences, maximum fluctuations so special attention should be given. For increasing the reliability and efficiency of the SVR model for weekends and holidays, separate models have been designed to predict load for these special or non-working days. The holiday model is trained for 21 different holidays of Rajasthan and is tested randomly for three holidays.

Table 3.4 Finding the most unstable time of the day.

Date	Time	RMSE (%)	MAPE
22 March	12 am to 3 am	0.61	0.6517
	8 am to 11 am	0.47	0.4033
	1 pm to 4 pm	1.30	1.4876
	6 pm to 9 pm	1.79	1.7514
Mar 23	12 am to 3 am	3.27	2.2927
	8 am to 11 am	0.96	0.7770
	1 pm to 4 pm	2.17	2.4761
	6 pm to 9 pm	2.93	2.4993
24 March	12 am to 3 am	0.71	0.6770
	8 am to 11 am	0.99	0.7847
	1 pm to 4 pm	1.44	1.5513
	6 pm to 9 pm	2.55	2.1507
Mar 25	12 am to 3 am	2.15	1.4366
	8 am to 11 am	0.89	0.7636
	1 pm to 4 pm	0.48	0.4016
	6 pm to 9 pm	2.69	2.6629
26 March	12 am to 3 am	0.70	0.6263
	8 am to 11 am	1.71	1.6744
	1 pm to 4 pm	1.29	1.4292
	6 pm to 9 pm	1.93	1.6690
Mar 27	12 am to 3 am	1.18	1.2065
	8 am to 11 am	1.31	1.2617
	1 pm to 4 pm	0.39	0.3543
	6 pm to 9 pm	2.97	2.8080

The results are compared with a simple SVR model, which is also trained for 21 days. Table 3.5 presents the error given by the special holiday model and the simple SVR model. On the above-mentioned days, the Holiday model has less MAPE when compared to the simple SVR model. Similarly, here we present a separate model for the weekend to increase the prediction accuracy. The Weekend model is also trained for 21 Saturdays and Sundays of the year and is tested for three random weekends in the upcoming months. Table 3.6 clearly explains the need and the importance of a separate weekend model. A weekly moving window is applied to the training data, which automatically trains the models for three weeks and predicts the result for the next week. Thus, the simple SVR model and the developed models can be used for automatically predicting the load for the next week as soon as it gets the load data of the current week. Table 3.7 shows the MAPE for 42 weeks of the year by the SVR model. Higher error in some weeks indicates changed load in this week due to factors that were not in the previous three weeks and were not incorporated in the forecast. With the help of Table 3.8, we can easily evaluate the accuracy of the moving window model.

Table 3.5 Comparison of Holiday model with simple SVR model.

Date	Holiday	Simple Model		Holiday Model	
		RMSE (%)	MAPE	RMSE (%)	MAPE
15-Aug	Independence Day	3.08	2.2612	1.01	0.7743
2-Oct	Gandhi Jayanti	5.89	4.6422	1.99	1.5418
25-Dec	Christmas	10	6.4803	5.01	3.4055

Table 3.6 Comparison of weekend model with simple SVR model.

Weekend (Sat & Sun)	Simple Model		Weekend Model	
	RMSE (%)	MAPE	RMSE (%)	MAPE
25-26 Feb	6.39	5.8197	1.38	0.8981
20-21 May	4.99	4.6374	2.04	1.6458
17-18 June	4.43	3.4756	2.17	1.7364

Table 3.7 Error Range for SVR Model.

Error range	Number of times RMSE fall in this range	Number of times MAPE fall in this range
0 - 1	0	0
1 – 1.5	4	10
1.5 – 2	7	10
2 – 2.5	8	8
2.5 – 3	8	6
3 – 4	8	4
4 – 6	7	4

Table 3.8 SVR MODEL ABSOLUTE ERROR PERCENTAGE FOR 10 WEEKS.

Error %	Feb 13– Feb 22	Feb 23-Mar 1	Mar.2–Mar 9	Mar 10- Mar 17	Mar 18-Mar 25	Mar 26-Apr 1	Apr 2 –Apr 8	Apr 9-Apr 15	Apr 16– Apr 22	Apr 23-Apr 29
<1	48.9583	34.9702	43.6012	21.2798	37.7698	22.7679	21.5774	25.8929	51.1905	34.9702
1≤e<2	29.1667	23.2143	35.8631	11.7559	26.6369	28.2738	22.9167	25.2976	30.0953	24.4047
2≤e<3	13.0954	11.0112	11.1607	8.33333	14.8809	16.8155	19.6429	19.4941	10.2679	15.2977
3 ≤e<5	6.84533	12.2024	8.63095	16.3691	18.6012	18.8988	29.1667	15.0928	6.25021	15.1786
5≤e	1.93452	18.6012	0.74048	42.2619	6.10119	13.2441	6.09643	14.2587	2.23214	10.4167

The model is not able to reduce the MAPE below 1%, but for ten predicted weeks, the error is less than 1.5%. For four weeks, the error is very high, which is not desirable. This puts forward the need for a better model because near real-time load forecasts play an important role in the successful operation of the Power System. All three proposed models are tested for 18 weeks. Table 3.9 compares the performance of all the three models on the basis of RMSE and MAPE. Among all the models, Model 1 shows remarkable performance with the least error.

Model 1 accurately predicts the load for week 18, which is the most disturbing week of the year, as seen in Table 5. Model 2 accuracy is far better than Model 3.

Table 3.9 SVR weekly window errors for 42 weeks consecutively.

Week	RMSE	MAPE	Week	RMSE	MAPE
s	1.80	1.3680	20	2.17	1.7094
2	3.55	2.4629	21	1.35	1.1109
3	1.73	1.3572	22	2.91	2.6238
4	6.36	5.6361	23	2.92	2.5604
5	2.41	2.0260	24	1.45	1.1200
6	3.23	2.5818	25	4.75	3.1932
7	2.63	2.4466	26	3.79	2.9880
8	3.31	2.8840	27	4.49	3.6231
9	1.76	1.2939	28	1.59	1.1890
10	2.63	2.1493	29	3.08	1.8788
11	5.31	4.6396	30	2.26	1.7386
12	2.01	1.5663	31	2.39	1.8077
13	1.89	1.4741	32	2.72	2.0950
14	3.98	2.7042	33	3.80	3.1037
15	3.09	2.3825	35	2.83	2.0122
16	1.42	1.0434	36	1.83	1.3281
17	1.39	1.0197	37	2.11	1.6232
18	6.67	5.9912	38	2.99	2.0758
19	4.27	4.2463	39	5.20	3.7167
20	2.17	1.7094	40	1.97	1.6383
21	1.35	1.1109	41	2.27	1.8916
22	2.91	2.6238	42	2.92	1.7630

Table 3.10 Comparison of prediction performance of all the three proposed models.

WEEK	RMSE			MAPE		
	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
1	1.8134	2.9433	3.6416	1.0719	1.8408	2.3017
2	1.8026	2.9066	3.5396	1.0489	1.8651	2.2823
3	1.7967	3.0004	3.7025	1.1299	1.9747	2.4523
4	1.7553	2.7852	3.3883	1.2366	2.0801	2.5346
5	1.7052	2.7545	3.3858	1.2226	2.0811	2.5678
6	1.7209	2.7076	3.3407	1.1084	1.8715	2.3326
7	1.5048	2.3714	2.9237	1.082	1.7945	2.226
8	1.4453	2.2836	2.8353	1.1002	1.831	2.2783
9	1.4485	2.2576	2.7785	1.0048	1.6382	2.0308
10	1.5075	2.2997	2.8403	0.9865	1.6348	2.0179
11	1.44	2.2166	2.731	0.9366	1.5687	1.9481
12	1.6129	2.5335	3.0988	1.0319	1.6978	2.1017
13	1.6931	2.527	3.0911	0.9849	1.6059	1.987
14	1.6576	3.0911	3.1101	1.0314	1.6894	2.1065
15	1.6315	2.4871	3.0595	1.0551	1.7406	2.1505
16	1.4291	2.1277	2.6414	0.9342	1.5093	1.8825
17	1.5235	2.2928	2.8194	1.0088	1.6378	2.0455
18	1.5345	2.8194	2.7139	1.034	1.6518	2.0505

If all the models discussed here are evaluated on the basis of MAPE and RMSE, then the proposed Model 1 is the most reliable with the highest accuracy. The proposed models are more precise than the SVR model Table 3.10, Table 3.11, and

Table 3.12 compares all the developed models for ten weeks on the basis of absolute percentage error. For this table, an absolute error has been calculated at a sampling rate of 15 minutes and grouped together for a week.

Table 3.11 MODEL 1 ABSOLUTE ERROR PERCENTAGE FOR 10 WEEKS.

Error %	Feb 13– Feb 22	Feb 23- Mar 1	Mar. 2 – Mar 9	Mar 10- Mar 17	Mar 18- Mar 25	Mar 26- Apr 1	Apr 2 – Apr 8	Apr 9- Apr 15	Apr 16– Apr 22	Apr 23- Apr 29
<1	67.45914	70.13373	63.44725	61.81278	58.54383	66.27043	64.93314	61.36071	64.63596	64.33878
1≤e<2	17.38484	16.04755	20.05944	19.01932	23.62556	17.68202	18.57355	24.96285	22.13967	22.13967
2≤e<3	6.092125	5.640461	8.172363	8.023774	8.172363	6.686478	8.320951	5.794948	6.686478	6.686478
3 ≤e<5	5.794948	4.60143	4.606241	7.875176	6.092125	5.200594	5.497771	5.794948	4.606241	4.457652
5≤e	3.268945	3.566122	3.71471	3.268945	3.566122	4.160475	2.674591	2.080238	1.931649	2.377415

Table 3.12 MODEL 2 ABSOLUTE ERROR PERCENTAGE FOR 10 WEEKS.

Error %	Feb 13– Feb 22	Feb 23- Mar 1	Mar. 2 – Mar 9	Mar 10- Mar 17	Mar 18- Mar 25	Mar 26- Apr 1	Apr 2 – Apr 8	Apr 9- Apr 15	Apr 16– Apr 22	Apr 23- Apr 29
<1	50.66865	45.17088	42.64487	42.05052	37.74146	44.13076	44.57652	40.56464	44.8737	49.92571
1≤e<2	19.46508	23.47697	24.66568	23.47697	25.11144	26.59733	26.00297	28.82615	29.27192	21.39673
2≤e<3	9.212481	11.29272	10.99554	11.14413	15.75037	10.54978	12.03566	14.26449	11.29272	13.37296
3 ≤e<5	10.10401	1054978	11.14413	11.88707	11.14413	8.618128	8.320951	9.509658	9.063893	8.172363
5≤e	10.54978	9.5096	10.54978	11.44131	10.2526	10.10401	9.063893	6.835067	5.497771	7.132244

3.5 CONCLUSION

In this chapter, at first, a PMU emulation technique has been presented for Static Security Analysis of the power system. Emulation of PMU for IEEE 14 bus system by using this method has been demonstrated, and the results are analyzed. The method presented can be used to prepare offline data for training and testing of neural network-based estimators/controllers for the power system. Any varying state can be quasi-statically represented and used in the method to generate relevant data for the above purpose. The presented methodology can also be used to develop and validate different control algorithms requiring PMU data in an offline environment. In general, the presented technique can be used to develop the static security assessment algorithms for real-time application in an offline environment.

Secondly, a near-real-time load forecaster has been developed. Experimental results confirm the efficacy of the proposed models over the SVR model. Our experimental results show's that Model 1, in terms of accuracy, is better than Model 2, which is better than Model 3. Further, the most unstable time of the day has been identified from 6 pm to 9 pm. In the upcoming chapter, we discuss the particularities related to the development of a real-time ATC estimator.