

Enhancing the Performance of Smart Grids via Applying Machine Learning Methods to Smart Meter Data

by

Ebrahim Pourjafari

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

© Ebrahim Pourjafari, 2019

Abstract

Conventional distribution systems need to undergo several updates to be able to meet modern energy requirements. While conventional grids have been able to satisfy the objectives they were designed for, lack of sensors and powerful communication systems, primitive control and management methods, weak protection strategies and inability to integrate Distributed Energy Resources (DERs) are some inadequacies of these grids that need to be rethought. Smart grid is a vision to address the shortcomings of conventional grids. To do this, the smart grid takes advantage of an advanced measurement and communication system known as Advanced Metering Infrastructure (AMI) to gather data across the grid. The goal is to leverage measurement data to enhance the operation of the system.

The objective of this thesis is to use the data provided by AMI to train machine learning models that can enhance the operation and management systems of smart grids. The first work in this thesis tries to improve the solution to the Volt-Var Optimization (VVO) problem by introducing an alternative data-driven approach for the current circuit-based VVO methods. Support Vector Regression (SVR) is used to train the data-driven model. Once the model is built, it is used as the internal model of Model Predictive Control (MPC) to optimally satisfy VVO objectives in a closed-loop control system. Since the model is built using the most recent data measurements, it can capture all features of the system. Therefore, in contrast to its circuit-based counterparts, it does not suffer from outdated circuit models.

Voltage sag readings of smart meters can be used to find the location of faults. The second work in this thesis tries to respond to a situation in which a limited number of AMI-enabled meters are available. The question is where to install those meters so that the observability of the grid is maximized. If locating faults is concerned, the question is where to install smart meters so that the location of faults can be predicted with the maximum accuracy. To find the best locations for installing smart meters, the proposed method in the second work states the fault locating problem as a classification problem. Then using an optimization procedure based on Simulated Annealing, the proposed method finds the optimal placement for smart meters for which the classification model can locate faults with the maximum accuracy. The classification models used in the second work are the Support Vector Machine (SVM) and Naive Bayes (NB) classifiers.

Preface

This thesis is an original work by Ebrahim Pourjafari. Prof. Marek Reformat was the supervisory author and was involved in developing the ideas for the research and manuscript composition. The third chapter of this thesis is published as an scholarly article in:

E. Pourjafari, M. Reformat, “A Support Vector Regression based Model Predictive Control for Volt-Var Optimization of Distribution Systems,” IEEE ACCESS, vol. 7, pp. 93352-93363, 2019.

To mom and dad

Acknowledgements

I would like to express my sincere gratitude and appreciation to Prof. Marek Reformat. This research and dissertation would not have been possible without his guidance and great supervision.

I would also like to thank the other members of my thesis examination committee, Dr. Petr Musilek, Dr. Hao Liang, Dr. Masoum Hossain, Dr. John Salmon and Dr. Hamed Mohsenian-Rad for spending time to review my thesis and providing invaluable comments to improve it.

Contents

1	Introduction	1
1.1	Smart Grid	2
1.2	Machine Learning	3
1.3	Motivation And Objectives of The Thesis	5
1.4	Outline of The Thesis	10
2	Background Material	11
2.1	Conventional Distribution Grids	11
2.1.1	Components of Distribution Grids	12
2.1.2	Configuration of Distribution Grids	15
2.2	Smart Grids: what is changing?	15
2.2.1	Key Components of A Smart Infrastructure	17
2.3	Support Vector Machine	19
2.3.1	SVM formulation	19
2.3.2	Kernel Trick	20
2.3.3	Multiclass Classification	21
2.3.4	Support Vector Regression	22
2.3.5	Training SVM Models	24
2.3.6	Validation and Hyperparameter Optimization	24
2.4	Naive Bayesian Classifier	25
2.4.1	Dealing with Continuous Data	27
2.4.2	Optimality of The Naive Bayes Classifier	28
3	Volt-Var Optimization of Distribution Systems	29
3.1	Literature Review	29
3.2	The Proposed VVO Scheme	33
3.3	The Framework of SVR-based MPC	36
3.3.1	Building An SVR Model of the Distribution Grid	36
3.3.2	Model Predictive Control	38
3.3.3	Need for A Fast Optimizer	42
3.3.4	GPU-based Optimizer	43
3.4	Simulation Results	50
3.4.1	Load Profile Simulation	52
3.4.2	Training the SVR Model	52
3.4.3	MPC Control	55
3.4.4	GPU-vs-CPU Speedup	65
3.5	Summary	67
4	Optimal Placement of Smart Meters for Locating Faults	70
4.1	Literature Review	71
4.2	Meter Placement Process: Overview	76
4.3	Meter Placement Algorithm: Details	80
4.3.1	Smart Meter Specification	81

4.3.2	Process of Merging Segments	82
4.3.3	Meter Placement Procedure: 1 st Stage	84
4.3.4	Meter Placement Procedure: 2 nd Stage	87
4.3.5	Fault Locating Procedure	89
4.3.6	An Illustrative Example	90
4.4	Simulation Results	95
4.4.1	Comparison between the NB and SVM	95
4.4.2	Simulation Results For IEEE34	98
4.4.3	Simulation Results For Extended IEEE34	104
4.5	Summary	106
5	Conclusion And Future Work	111
5.1	Volt-Var Optimization	111
5.2	Smart Meter Placement	113
	References	115
	Appendix A Simulated Annealing	122
	Appendix B Extended IEEE34 Test System	124

List of Tables

2.1	Comparison between conventional and smart grids	17
3.1	Installed DERs on IEEE123	52
3.2	Performance of trained SVR models	54
3.3	Simulation results of scenario 3	59
3.4	Simulation results of scenario 5	61
3.5	Computation time of GPU-based MPC and sequential MPC	65
4.1	Types of fault in distribution systems	77
4.2	Comparison between the NB and SVM	97
4.3	Comparison between different fitness functions for the first stage	98
4.4	Number of zones for each type of fault	99
4.5	Comparison in the number of zones if the fitness function changes	103
4.6	The accuracy of the proposed model in locating faults	104
4.7	Number of zones for each type of fault	106
B.1	Distributed loads 802-806	124
B.2	Distributed loads 808-810	126
B.3	Distributed loads 818-820	126
B.4	Distributed loads 820-822	126
B.5	Distributed loads 816-824	127
B.6	Distributed loads 824-826	127
B.7	Distributed loads 832-858	127
B.8	Distributed loads 858-834	127
B.9	Distributed loads 834-860	128
B.10	Distributed loads 860-836	128
B.11	Distributed loads 836-840	129
B.12	Distributed loads 862-838	129
B.13	Distributed loads 842-844	129
B.14	Distributed loads 846-848	129
B.15	Distributed loads 844-846	130
B.16	Added distributed loads 806-808	130
B.17	Added distributed loads 808-812	130
B.18	Added distributed loads 812-814	130
B.19	Added distributed loads 816-824	131
B.20	Added distributed loads 854-852	131
B.21	Added distributed loads 854-856	131
B.22	Added distributed loads 858-864	131

List of Figures

1.1	Main types of machine learning	4
1.2	The proposed data-driven VVO Approach	7
1.3	The proposed optimal smart meter placement	9
2.1	A conventional distribution system	12
2.2	Linear Support Vector Classifier	19
2.3	Linear Support Vector Regression	22
2.4	Effectiveness of kernel density estimation	27
3.1	Overall scheme of SVR-based MPC	34
3.2	Block diagram of MPC for the VVO of a distribution system	35
3.3	Inputs and outputs of the trained SVR model	38
3.4	The architecture of GPU	45
3.5	Parallel implementation of dot product on a GPU	47
3.6	Vector-matrix multiplication on GPU	48
3.7	GPU implementation of PSO	49
3.8	IEEE123 distribution test feeder	51
3.9	Daily profile of a bus feeding 6 households	53
3.10	5-day total residential load profile of IEEE123	54
3.11	Performance of controllers under scenario 1	56
3.12	Tap positions of OLTCs for Cont. 1 under scenario 1	58
3.13	KVAR injection of capacitors for Cont. 1 under scenario 1	59
3.14	Performance under scenario 1 for minimizing loss	60
3.15	Performance under scenario 1 for minimizing voltage deviation	61
3.16	Performance under scenario 2	62
3.17	Tap positions of OLTCs for Cont. 1 under scenario 3	63
3.18	KVAR injection of capacitors for Cont. 1 under scenario 3	64
3.19	Comparing the results of scenario 3 and scenario 4	65
3.20	Comparing the results of scenario 5 to scenario 6	65
3.21	Voltage profile under scenario 6	66
3.22	Voltage deviation under scenario 7	67
3.23	Power loss and average voltage deviation under scenario 8	68
3.24	GPU-vs-CPU speedup	68
3.25	Stream concurrency on GPU	69
4.1	Dividing a line into several segments	78
4.2	Flowchart of the proposed optimal placement method	79
4.3	Voltage sag caused by a fault	81
4.4	Example of acceptable merge of zones	83
4.5	Neighboring locations for smart meters	88
4.6	An illustrative example	90
4.7	First iteration of the first stage	92
4.8	Second iteration of the first stage	93
4.9	Second iteration of the first stage, continued	94

4.10	Third iteration of the first stage	95
4.11	Final result of optimal placement	95
4.12	IEEE34 test bus system	96
4.13	Zone map for single-phase and three-phase faults	100
4.14	Zone map for LL faults	101
4.15	Zone map for LLG	102
4.16	Confusion matrix for the fault-type-detection model	103
4.17	Zoning map for LGa	107
4.18	Zoning map for LLL	108
4.19	Confusion matrix for the fault-type-detection model for extended IEEE34	109
B.1	Extended IEEE34 system	125

List of Abbreviations

AMI	Advanced Metering Infrastructure
DER	Distributed Energy Resources
GPU	Graphics Processing Unit
MPC	Model Predictive Control
NB	Naive Bayes
OLTC	On-Load Tap Changer
PMU	Phasor Measurement Unit
PSO	Particle Swarm Optimization
SIMD	Single-Instruction Multiple-Data
SMX	Streaming Multiprocessor
SVM	Support Vector Machine
SVR	Support Vector Regression
VVO	Volt-Var Optimization

List of Symbols

α_i, α_i^*	Dual parameters in a SVM model
b	Bias in a SVM model
B_{ij}	Susceptance between buses i and j
β	Error compensation coefficient of MPC
C	Box constraint of SVM
CAP_j	Value of the j^{th} capacitor
CAP_i^{max}	Maximum kVAR of the i^{th} capacitor
Cap_{max}	Maximum allowed change in kVAR of a capacitor
Cap_i^{step}	kVAR of each capacitor in the i^{th} capacitor bank
c_1, c_2	Accelerating coefficients of PSO
ΔCap_i	Change in the value of the i^{th} capacitor
ΔTap_i	Change in the ratio of the i^{th} OLTC
$\langle \cdot, \cdot \rangle$	Dot product
E_{loss}	Loss prediction error
E_{volt}	Voltage prediction error
e_i	Voltage prediction error for the i^{th} bus
e_p	Power loss prediction error
ϵ	The length of insensitive zone in an SVR model
$\ \cdot, \cdot \ $	Euclidean norm
$F()$	Fitness function for meter optimal placement
F_i	The i^{th} feature in a dataset
G	Vector representation of Gaussian kernel
G_{best}	Best position for all particles of PSO
$g()$	Gaussian distribution function
G_{ij}	Permittance between buses i and j
γ	Coefficient of Gaussian kernel
k	Boltzmann constant in Simulated Annealing
$k(\cdot, \cdot)$	Kernel for SVM
L_{SM}	Location of smart meters
Λ	The class of a sample

I	current
$J()$	primal objective function of SVM
μ	Mean value of a Gaussian distribution
N_{cap}	Number of capacitors
N_L	Number of loads
N_{tap}	Number of OLTCs
$NZ()$	Number of zones
ω	Weight vector in SVM
$p()$	Probability of a variable
P_{best}	Best position of the i^{th} particle of PSO
P_{feeder}	Active power measured at feeder
P_i	Active power injection at the i^{th} bus
P_{loss}	Power loss
Q_i	Reactive power injection at the i^{th} bus
R_{line}	Resistance of a line
SV	Support vector matrix
σ	Variance of a Gaussian distribution
T	Temperatre in Simulated Annealing
TAP_i^{max}	Maximum ratio of the i^{th} OLTC
Tap_{max}	Maximum allowed ratio change in an OLTC
T_c	Control horizon of MPC
T_{cap}	Number of control steps the kVAR of a capacitor cannot change
T_p	Prediction horizon of MPC
T_s	Length of the control step of MPC
T_{tap}	Number of control steps the ratio of an OLTC cannot change
TAP_n	Tap ratio of the n^{th} OLTC
θ_{ij}	Phase angle between buses i and j
V_{max}	Maximum allowed voltage
V_{min}	Minimum allowed voltage
V_{sp}	Setpoint voltage for MPC
w_j^i	Weight coefficient in MPC optimization objective
X_i	Position vector for the i^{th} particle
x_i	Training data input
ξ_i	Slack variable in SVM formulation
χ	Inertia weight of PSO
y_i	Training data output

Chapter 1

Introduction

Modern energy requirements pose multiple challenges to conventional distribution systems. In its original design, the distribution grid was meant to distribute electricity between customers, by bridging between them and the bulk part of the power grid. While the conventional grid can reliably serve this purpose, its original design is incapable of satisfying modern energy needs. The conventional grid suffers from the minimal usage of sensors and measuring instruments and a lack of powerful electronic communication, so severely that there is almost no measuring device with communication capabilities installed downstream a distribution feeder. The absence of sophisticated control methods is prominent. Though the major portion of power loss in a power grid happens across its distribution system, the conventional solution to lowering down power loss has been the implementation of primitive rule-based approaches that can never guaranty the optimal reduction of losses.

In addition, the one-way electricity flow design of conventional grids only allows for the transfer of electricity from the bulk power system to customers. The one-way power transfer design prohibits the installation of Distributed Energy Resources (DERs) inside the distribution network, although the distribution grid is the main section of the power grid for integrating such energy resources. Moreover, the protection is poor, and the system restoration has to be done manually. There is no automatic fault locating method that could pinpoint the location of fault in a distribution system. Following the occurrence of a fault, power utilities rely on customer calls to locate the outage area

and send their crew to visually investigate that area and find the cause and location of the fault. Also, the crew has to manually reset protection devices after the fault is cleared.

1.1 Smart Grid

The smart grid is a concept aimed to address numerous inadequacies of conventional distribution systems. It is an effort to upgrade the distribution grid in order to meet modern energy needs, integrate DERs, reduce greenhouse gas emissions, improve power quality indices, and enhance protection and security strategies. To do this, the smart grid takes advantage of advanced measurement and communication technologies to improve the observability of the grid, sophisticated control and optimization approaches to enhance power quality and provide necessities for the integration of DERs, and smart protection and restoration mechanisms. A smart grid model can be divided into three subsystems from a technical point of view [1]:

Smart infrastructure system: This system provides the capability for two-way flow of electricity and information. Two-way flow of electricity means that besides the conventional flow of power from the generation side to customers, DERs can be installed in the smart grid and inject their generated electricity into the grid. Customers of a smart grid are also able to produce electricity and put it back into the network. The two-way flow of information implies that customers' information can now be collected and sent to operation centres through communication technologies. Utility providers can also send information and commands to end-user metering devices [1]. The infrastructure that has provided the two-way flow of data in smart grids is called Advanced Metering Infrastructure (AMI). AMI is a combination of several technologies including metering devices and communication systems, which was originally designed to collect energy consumption data of customers. However, it was soon proved based on several studies the data collected via AMI could reveal invaluable operational information about the distribution grid. The measurement devices used in this infrastructure are called smart meters. Basic types

of these devices can measure voltage amplitude and power consumption, while advanced smart meters have extra capabilities such as recording the load pattern, recording time-based voltage sags, outage count and duration, *etc.*

Smart management system: This system is responsible for providing sophisticated control and management functionalities to the grid. The smart management system exploits new capabilities of the smart infrastructure, mainly the data provided by metering and communication technologies to pursue advanced control objectives such as Volt-Var Optimization (VVO), emission control, utility maximization, *etc.*

Smart protection system: The smart protection system leverages the advanced features of the smart infrastructure to provide smarter solutions to reliability analysis, fault protection, and cyber security.

1.2 Machine Learning

Machine learning is a branch of computer science in which a set of training data is used to build a mathematical model for recognizing, predicting, or making decisions about future data. The process of building a mathematical model is called training, in which no explicit learning instruction is involved. Instead, the machine learning algorithm tries to learn from the embedded characteristics of training data. There are three main paradigms of machine learning:

Supervised Learning: In this learning method, the training dataset contains both inputs and desired outputs. The machine learning model tries to learn the functional relationship between training inputs and outputs. This is typically performed by expressing the input-output relationship as an optimization problem and trying to optimize it so that the prediction error between the predicted outputs and desired outputs are minimized. Classification and regression are two examples of supervised learning. In classification, each training sample that belongs to a class, is labelled as a member of that class. The objective is to train a mathematical model that can predict the class of each sample. In regression, the goal is to estimate the output of the system

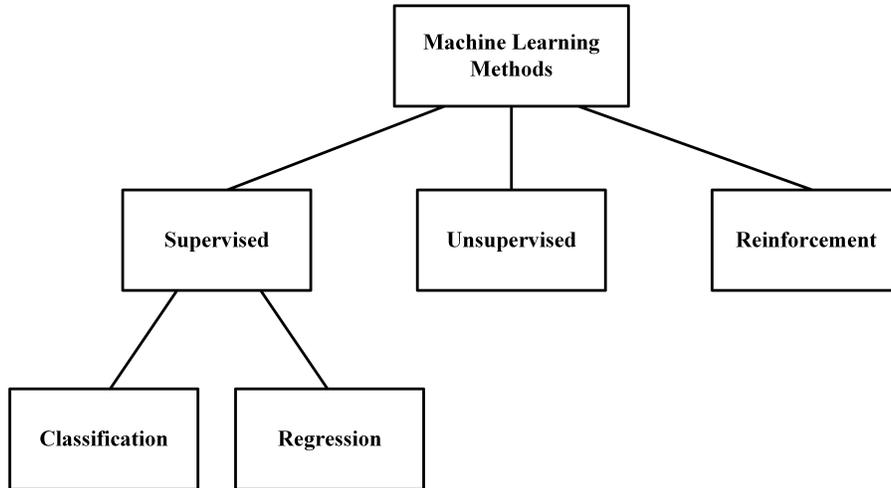


Figure 1.1: Three main types of machine learning

based on a set of input data. It can be seen as the process of finding the relationship between a dependent variable (output) and independent variables (inputs).

Unsupervised Learning: In this type of learning, there is no clear output for training data. So, the machine learning model has to learn based on similarities between input data and make actions accordingly. Clustering is an example of unsupervised learning.

Reinforcement Learning: This method of learning is between supervised and unsupervised learning, in which no output is provided, yet the machine learning model tries to learn the input-output relationship of the data [2]. Instead of using labelled outputs, reinforcement learning rewards an agent based on the decisions the agent makes, in order to improve the ability of the agent to make the right decision. Fig. 1.1 shows the main three types of machine learning paradigms.

The output of the training process is a machine learning model. This mathematical model is able to make predictions based on new input data. Machine learning models are different in many aspects, however the main difference is the application they are designed for, whether it is classification, regression, clustering, *etc.* For example neural networks, the Support Vector Machine (SVM) and the Naive Bayes (NB) network can be used for both classification

and regression, while K-means can be applied to clustering problems.

1.3 Motivation And Objectives of The Thesis

As mentioned earlier, a smart grid can be divided into the smart infrastructure, smart management, and smart protection systems. AMI, as a part of the smart infrastructure, provides crucial operational data about the grid. The main motivation of this thesis is to apply machine learning methods to measurement data from AMI, to develop data-driven approaches that can enhance the performance of management and protection systems of smart grids. The measurement units considered in this research are relatively low-cost smart meters that can perform power and voltage amplitude measurements and are commonly used in distribution grids. This thesis does not consider advanced high-cost meters like Phasor Measurement Units (PMUs) that are mainly installed in transmission systems, not distribution grids.

To do this, the thesis considers two different scenarios for AMI implementation. The first scenario assumes that all customers across the grid are equipped with AMI-enabled smart meters. Therefore, there is full observability of the grid. The smart meters in this scenario are supposed to be performing basic measurements including power consumption and RMS voltage measurements. The objective here is to leverage the full observability provided by AMI to raise the efficiency of the Volt-Var Optimization. The data collected from smart meters is used to develop a data-driven model of the system which can improve the Volt-Var Optimization of distribution systems. The objective of VVO study, as a part of the smart management system, is to minimize power loss, and to keep voltage profiles within acceptable levels. Current VVO approaches are either rule-based or circuit-based. Rule-based methods follow simple rules to perform the VVO, although relying on primitive decision-making rules causes sub-optimal results. Circuit-based methods can provide optimal solutions to the VVO problem, provided that their circuit models are accurate. But this is not usually the case, as power utilities do not often keep track of changes and modifications in their distribution grids. So, their circuit models could be

outdated.

In this thesis, the Support Vector Regression (SVR) is applied to AMI measurements to build a nonlinear model of the distribution system. The regression model is then served as the internal model of Model Predictive Control (MPC) to optimally control the distribution grid and enforce VVO objectives. MPC takes advantage of an internal optimizer to generate its control decisions. It seems a perfect controller for the VVO problem, since the VVO objectives can be incorporated into the internal optimization procedure of MPC. Besides, the feedback control feature of MPC allows for compensation of prediction errors of the SVR model or errors caused by topology changes in the system. A Graphics Processing Unit (GPU) based PSO optimizer is employed for solving the internal optimization problem of MPC. This parallel optimizer ensures the VVO controller is suitable for real-time applications. The followings are the main features of the proposed data-driven VVO method:

- Building a model using measurement data allows the proposed method to capture the latest operational conditions of the distribution grid. Therefore, it can provide a superior response to the VVO problem compared to circuit-based methods, in case the circuit model is inaccurate.
- It is hard for circuit models to accurately model voltage dependency of loads. In contrast, the SVR model in our proposed method can easily capture such dependency from measurement data and model it.
- DERs can be implicitly detected and modelled.
- The nonlinear SVR model can adequately map nonlinear powerflow equations of the system. Thus, it can provide a more accurate solution to the VVO problem compared to other data-driven methods that use linear machine learning methods like K-nearest neighbor.
- The closed-loop feature of MPC allows for compensation of prediction errors of the SVR model. This means that even if the regression model is to some extent inaccurate, MPC can still provide close to optimal solutions to the VVO problem.

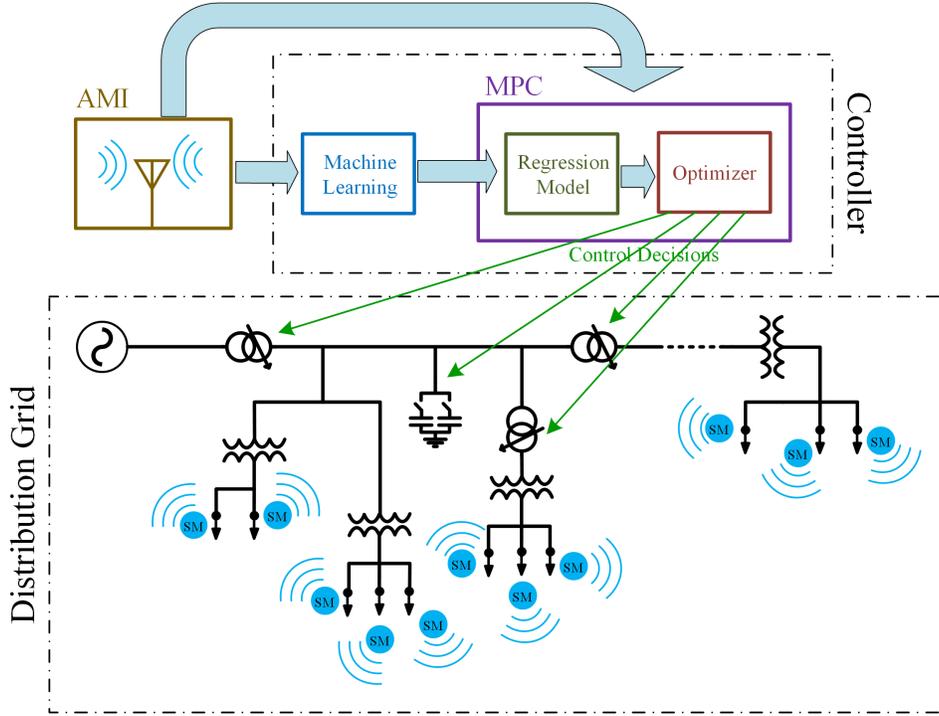


Figure 1.2: The proposed data-driven VVO Approach

- MPC takes advantage of an internal optimization process to generate its control decisions. VVO objective and operational constraints can be effectively integrated to this optimization process.
- The GPU-based optimizer ensures real-time operation of MPC.
- In order to reduce computational burden, many circuit based VVO methods use simplified versions of powerflow equations that can be only used in radial distribution systems. In contrast, the SVR model in our proposed approach does not make any assumption about the topology of the grid. Therefore, it can be applied to meshed grids as well.

Fig. 1.2 describes the overall scheme for the proposed data-driven VVO approach.

The second scenario investigated in this thesis considers a case that the number of available smart meters is limited, so it is not possible to install a smart meter for each customer. The objective is to find a solution to the question that where to install this limited number of smart meters so that

the visibility of the system is maximized. The smart meters considered for this scenario are more advanced than basic meters. They can record voltage sags in a time-based manner, and have backup power that ensures their uninterrupted operation during outages. Considering this measurement scenario, this thesis introduces a method that will enhance the performance of protection systems of smart grids. More specifically, this thesis proposes a method for the optimal placement of smart meters across the system so that the location of fault can be predicted with the maximum accuracy.

There are several methods for locating the outage area or finding the location of fault. Some of these methods even use smart meter measurements to locate fault. Yet, there has not been a study for determining the best locations for placing smart meters which could result in the maximum fault locating accuracy. The proposed method can locate the best places for installing smart meters through an optimization procedure performed by Simulated Annealing. To be more precise, the proposed method performs two optimizations at once. One is to find the best places for installing smart meters. The other is to divide the distribution grid into the maximum possible number of zones, in which the zone that contains a fault can be pinpointed with the absolute accuracy.

To perform these two optimizations, the fault locating problem is stated as a classification problem. First, the distribution system is divided into small segments. A segment is the maximum accuracy the proposed method can offer as the location of a fault. At each iteration of the optimization procedure for locating meters a potential set of locations for placing meters is chosen. Then a classification model (either the SVM or NB classification) is trained, aiming to classify all segments accurately. If the confusion matrix shows some segments are misclassified, a merging routine will merge those segments into a new zone. The merging procedure continues until there is no more misclassified segment left. Therefore, the maximum number of zones that ensures the absolute fault locating accuracy is achieved. The contributions of the proposed method in the second study are:

- The proposed method determines the best locations for installing smart meters, so that the maximum capability of a limited number of smart meters for locating faults can be exploited.
- Zoning in the distribution grid is determined in such a way that the accuracy of locating faults is maximized.

Fig. 1.3 represents the overall scheme of the proposed method. As the figure shows, the output of the optimization procedure is the best locations for installing smart meters, as well as fault locating and type classification models and zoning maps for each type of fault. After the smart meters are installed at designated locations, their voltage readings can be fed to the fault locator module to locating the zone of the fault.

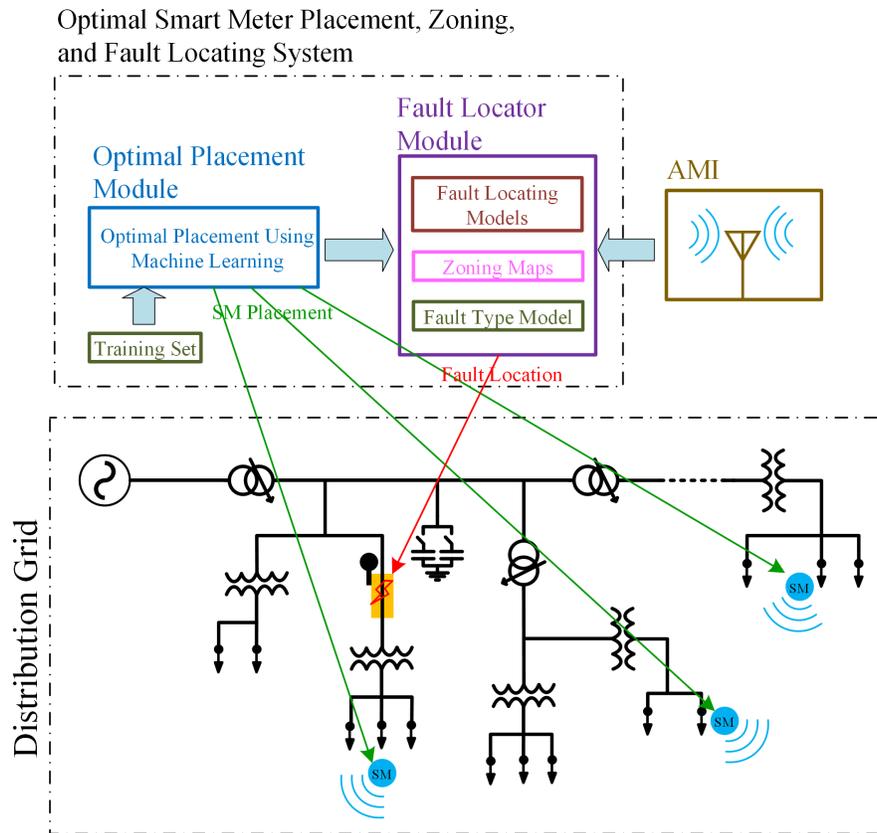


Figure 1.3: The proposed optimal smart meter placement: Optimal placement module finds the best locations for installing smart meters. Fault locator module locate the zone of fault using smart meter readings.

1.4 Outline of The Thesis

The rest of this thesis is organized as follows:

Chapter 2 Provides background knowledge about distribution systems and machine learning. It describes several components of distribution grids which are involved in the power quality management and protection of the grid. It then describes deficiencies of conventional grids and how smart grids aim to tackle those deficiencies. Different parts of a smart infrastructure are explained as well. The chapter then focuses on machine learning models. Details about the SVM for both classification and regression tasks are provided, and training and hyperparameter optimization procedures are explained. The final section of this chapter describes the NB classification model.

Chapter 3 describes the proposed SVM-based MPC approach for the VVO problem. After literature review, the overall scheme of the method is explained, followed by implementation details of the proposed method at the next section. The simulation results of applying the method to IEEE123 test system shown and finally, a summary is provided about the strengths of the method and observations about it.

Chapter 4 is dedicated to the novel method for optimal placement of smart meters. A comprehensive literature review is presented at the first section, followed by explaining the overall scheme of the method. The next section explains details of the proposed method and provides an illustrative example that shows how the optimization and merging procedures are done. The simulation results of applying the proposed method to IEEE34 test system and its extended version are provided, while a summary about the proposed method concludes this chapter.

Chapter 5: This chapter is dedicated to concluding the thesis and providing some suggestions about future work on the subject of using machine learning in smart grids.

Chapter 2

Background Material

This chapter of the thesis provides background information about distribution systems and machine learning methods. The first section of the chapter describes a conventional distribution system and its components, while the second section talks about smart grids, why these grids are considered smart, and the components of a smart grid infrastructure. Section 2.3 describes the Support Vector Classification as well as Support Vector Regression. Finally, section 2.4 illustrates the Naive Bayes classifier.

2.1 Conventional Distribution Grids

The power system consists of three subsystems: the generation system, the transmission system, and the distribution system. Each of these subsystems has a distinguished function. The generation system is responsible for generating electric power. The transmission grid connects the generation system to load centres and allows for transmitting power through its high-voltage lines. Finally, the distribution system links customers to the rest of the power grid.

Fig. 2.1 shows the one-line diagram of a conventional distribution grid. The transformer in the distribution substation steps down transmission-level high voltage (35 to 230 kv) to the medium voltage of primary distribution circuits (600 volts to 35 kv) [3]. Primary distribution circuits consist of feeder lines, distributor conductors and lateral cables. Feeders are distribution lines that carry large amounts of currents. A distribution transformer can feed several feeders. The flow of current in a distribution system is from the feeder

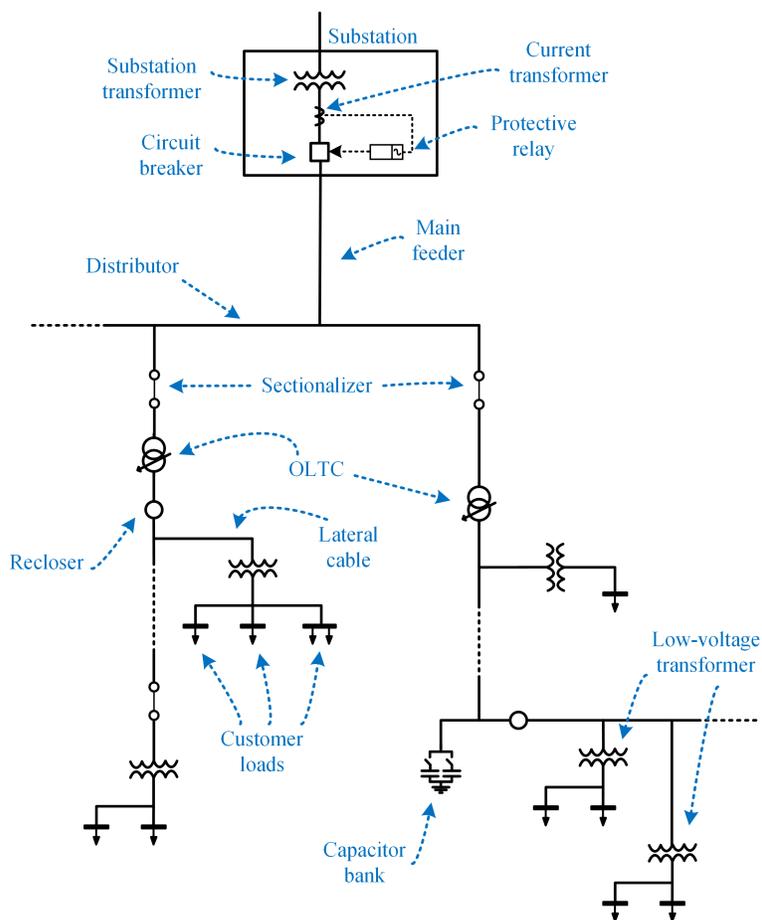


Figure 2.1: One-lin diagram of a classical distribution system and its components

to distributor conductors, then to lateral cables. Lateral cables are connected to low-voltage distribution transformers that reduce the primary distribution voltage to the low voltage level of secondary distribution circuits (usually 240 volts in North America). The secondary distribution grid connects customers to the rest of the distribution system.

2.1.1 Components of Distribution Grids

In addition to conductors and transformers, some other devices are installed in a distribution system to improve its performance and ensure its safe operation. While capacitor Banks and OLTCs can improve the performance of a distribution system, devices like protective relays and circuit breakers pro-

tect it against faults. The functionality of these devices is described in the following:

Capacitor: Capacitors can improve the performance of a distribution system in the following ways [3]:

- Reducing power loss: they can inject a portion of the reactive power consumed by motors and other loads and hence, reduce the line current. A decrease in the line current significantly lowers the power loss on that line ($P_{loss} = I^2 R_{line}$).
- Increasing loading capacity: capacitors reduce the current in the grid. This means the freed loading capacity of the distribution system can be utilized to serve more customers.
- Improving voltage profile: injecting reactive power to a system can correct power factor of that system. A corrected power factor, in turn, will improve the voltage profile of that power grid.

Capacitor banks are made of a parallel combination of several capacitor units. Different ratings for capacitor units from 50 kVAR to 500 kVAR are available [3]. Capacitor banks can be installed either in the substation or on the primary distribution grid. The capacitor banks installed on primary grids can reduce power loss and voltage drop more efficiently, although the procedure for the optimal control of these capacitors is more complicated.

Capacitor banks can be controlled locally or remotely, while each capacitor unit in the bank can be switched separately. It is very important to switch the right amount of capacitance on, in order to minimize power loss and voltage drop. Improper control of capacitor banks can have adverse effects. Injecting kVAR more than the need of loads results in a leading power factor. This will increase line currents and power loss, and reduce the loading capacity of the system. Also, injecting too much reactive current to the system increases voltage profiles to dangerous levels. Proper control and switching of capacitor banks is an objective of the VVO study.

OLTC: In addition to capacitors, On-Load Tap Changers (OLTCs) can be used to improve the voltage profile. An OLTC is a transformer with two windings in which the ratio of the secondary winding is adjustable. The alteration of the secondary ratio is done using a mechanical arm changing the physical connection point to the secondary. OLTCs are often designed so that the secondary tap can change up to 32 steps, changing the secondary voltage $\pm 10\%$ of the primary voltage level. As the name suggests, the tap-changing procedure is performed so that the load current is never interrupted. OLTCs can be controlled remotely or by local controlling devices. The proper control of OLTCs is an objective of the VVO study.

Overcurrent relay: Besides the aforementioned devices that enhance the performance of the distribution system, another set of apparatus are employed to protect the grid against faults and malfunctions. Such devices include protective relays, circuit breakers, sectionalizers and reclosers. Several types of relays are connected throughout the power system to detect undesirable conditions and isolate defective apparatus. The distribution system is subject to faults that cause high short-circuit currents. High load demand levels can also result in overload currents. Therefore, the most commonly used protective relay in distribution grids is the overcurrent relay.

Overcurrent relays for the distribution grid are installed in distribution substations and send tripping commands to circuit breakers to de-energize the downstream distribution circuit in case a fault happens. A typical overcurrent relay considers a time delay before initiating a trip command, hoping the fault gets cleared out or other downstream overcurrent relays closer to the location of the fault act first. The duration of this delay has an inverse relationship with the amplitude of the short-circuit current. The bigger the fault current, the sooner the overcurrent relay trips the circuit.

Circuit breaker: A circuit breaker is a switching device in the substation that can be used to de-energize a distribution feeder in case a short-circuit in its downstream happens. Circuit breaker are controlled by protective relays such as the overcurrent relay and unlike fuses, can be reset to restore power to the feeder after the fault is cleared out.

Recloser: An automatic circuit recloser is a switchgear installed on overhead distribution lines to shut off the distribution circuit if momentary faults such as lightning and windblown tree branches occur. The recloser closes the circuit an instant later to check if the fault is cleared or not. If the fault was temporary, the closer keeps the circuit close to restore power. If the fault is not removed, the recloser shuts off the circuit again. This open/close cycle is tried a few times and if the fault persists, the recloser remains off until the fault is cleared out by the power utility.

Sectionalizer: The sectionalizer is another protecting device used in distribution systems. It is a switching device that automatically de-energizes faulty sections of a distribution system. The sectionalizer is used in conjunction with other switching devices such as circuit breakers and reclosers, aiming to confine permanent faults to a smaller section of the distribution grid.

2.1.2 Configuration of Distribution Grids

Distribution grids can have different length and configurations. The configuration can be radial or meshed (networked). Most distribution systems are radial. A radial distribution system is a grid in which there is only one path for the flow of current from the feeder to a load. In contrast, there could be multiple paths that connect two points of the circuit in a meshed grid. Radial systems are more popular as they are cheaper to implement and fault protection of them can be performed easier than networked grids. On the other hand, meshed grids are more reliable as there are multiple paths to transfer electric power to downstream circuits. The shape and length of distribution systems can extremely vary from one grid to another and depend on several factors including the geographical topology of the area covered by the circuit, natural obstacles, street layouts and the location of big loads [3].

2.2 Smart Grids: what is changing?

as mentioned in the previous chapter, the conventional distribution systems suffers from the lack of sophisticated control and management system, inade-

quate measurement and communication devices, as well poor fault protection strategies. The main reason why the power grid could not keep pace with the modernization other industries experienced might be due to the fact that the power grid had got so massively expanded that it was impractical to upgrade it from the technical and economical points of view.

However, this trend has changed, and the distribution network has been undergoing significant modernization during the last two decades. Advances in electronic communication, the emergence of electric vehicles, the appearance of machine learning and computational intelligence, the push for reducing greenhouse gas emissions, and the need for a more efficient and reliable power grid have all contributed to the advent of intelligent or smart grids. The smart grid integrates power, communications, and information technologies for an improved power grid infrastructure that serves loads while preparing the infrastructure for an ongoing evolution of end-use applications [4].

A comparison between conventional and smart grids is provided in Table 2.1 [5]. Smart grids take advantage of advanced communication systems, sensors, measuring units, feedback control and intelligent systems, aiming to improve the performance of the grid. The benefits smart grids can contribute are [6]:

- Improving power reliability and quality
- Optimizing facility utilization and avoiding construction of peak load power plants
- Enhancing capacity and efficiency of existing electric power networks
- Improving resilience to disruption
- Enabling predictive maintenance and self-healing responses to system disturbances and faults
- Facilitating expanded deployment of renewable energy sources
- Integrating DERs to the grid

Feature	Conventional grid	Smart grid
Metering	mechanical	digital
Communication	one-way	two-way
Generation	centralized	distributed
Restoration	manual	self-healing
Monitoring	manual check	remote check
Control	limited	pervasive

Table 2.1: Comparison between conventional and smart grids

- Automating maintenance and operation
- Reducing greenhouse gas emissions by enabling electric vehicles and new power sources
- Reducing oil consumption by reducing the need for inefficient generation during peak usage periods

2.2.1 Key Components of A Smart Infrastructure

There are several components that make a grid infrastructure smart. Each of these components adds a smart functionality such as smart metering, communication, storage and generation to the grid. The components of a smart infrastructure are described in the following:

Smart Meter: A smart meter is a modernized version of the traditional mechanical meter. A smart meter can perform several types of real-time measurements and is able to communicate with the meter data management system to deliver measured data. The communication is bidirectional, which means the smart meter can receive commands from that centre as well. Beside energy measurements, some smart meters are able to perform measurements that are valuable for power system operation, performance, and reliability studies. For example, the residential smart meter I-120+c from General Electric can perform voltage measurements (min, max, average), time-based voltage sag/swell measurements, outage count and duration recording, as well as load profile recording [7].

A variety of technologies are currently used for smart meter communication. Cellular phone networks (2G/3G/4G), landline, radio frequency, satellite communications, Wi-Fi, and powerline communication are possible technologies utilized for smart meter communication [8]. Powerline communication uses the physical electric distribution system to send/receive data.

AMI: AMI is one of the most important features of a smart grid that provides a configured structure for collecting and transferring data throughout the grid. This infrastructure is achieved by integrating smart metering technology, communication network, meter data management systems, and means to integrate the collected data into software packages and interfaces [9]. The data management system is a system located at the utility provider end, which is responsible for restoring and analyzing data received from smart meters, as well as taking real-time actions for changes and emergencies happen in the grid [9]. AMI collects and transfers data in periodic intervals about 15 minutes [9].

DER: Installing DERs close to load tapping points has been a growing trend in recent years. DERs reinforce the operation of the distribution grid by meeting a portion of load demands in that grid, lowering down power loss, and improving voltage profile. In addition, because of the nature of such sources which are usually micro-scale renewables like wind and solar, DERs can help reducing greenhouse gas emissions. They allow for two-way flow of power in the grid, which is in contrast to the way conventional distribution systems are designed and operate. Hence, the methods used for controlling and operating distribution systems must be updated to consider DERs and handle the two-way flow of power.

Energy Storage System: Customers' load profiles change dramatically during the day. Also, renewable energy resources have a variable nature. An energy storage system can be seen as a tool for moderating variable power generation and consumption. Although energy storage systems are not widely available yet, they are believed to be a key asset in the evolving smart grids [10].

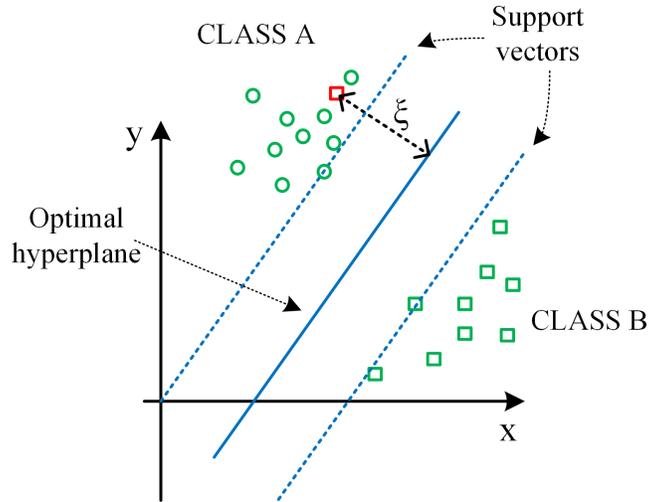


Figure 2.2: A linear Support Vector Machine classifying the data of two classes

2.3 Support Vector Machine

The Support Vector Machine is a supervised machine learning model, developed by Vapnic and co-workers [11] at AT&T Bell laboratories. The SVM is widely used in classification and regression. Fig. 2.2 shows how a linear SVM classifier sets the boundary between the sample data of two linearly separable classes. The SVM works based on maximal margin classifier strategy [12]. This strategy implies that among all possible hyperplanes that can linearly separate two classes, the classifier seeks the hyperplane that maximizes the bound between those two classes. For example, in Fig. 2.2, the SVM classifier finds a hyperplane (the solid blue line) so that the distance between the hyperplane and the nearest data samples of class A and class B are maximized. The two parallel dashed lines on two sides of the maximal margin hyperplane represent the shortest distance between the members of each class to the maximal margin (optimal) hyperplane. These two parallel lines are called support vectors.

2.3.1 SVM formulation

Suppose data samples in Fig. 2.2 are represented by $(x_i, y_i), i = 1, \dots, l$. We can label the members of class A as $+1$ ($y_i = +1$) and the members of class B

as -1 ($y_i = -1$). The optimal hyperplane can be stated as [13]:

$$\langle \omega, x \rangle + b = 0 \quad (2.1)$$

where $\langle \cdot, \cdot \rangle$ denotes dot product, ω is a weight vector, and b is the bias term. Therefore, $\langle \omega, x \rangle + b \geq +1$ for all members of class A and $\langle \omega, x \rangle + b \leq -1$ for the members of class B. We obtain the following inequality by incorporating y_i into (2.1):

$$y_i[\langle \omega, x \rangle + b] \geq 1 \quad (2.2)$$

To find the optimal hyperplane, one has to solve the following optimization problem [13]:

$$\text{minimize } J(\omega) = \frac{1}{2} \|\omega\|^2, \quad (2.3a)$$

$$\text{s.t. } y_i[\langle \omega, x \rangle + b] \geq 1, i = 1, \dots, l \quad (2.3b)$$

Consider the case that the classes are not linearly separable as a result of existing outliers like the red square in Fig. 2.2. This problem can be solved by allowing the outliers to be misclassified. This is done by introducing the slack variable ξ_i to (2.3):

$$\text{minimize } J(\omega) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l \xi_i, \quad (2.4a)$$

$$\text{s.t. } \begin{cases} y_i[\langle \omega, x \rangle + b] \geq 1 - \xi_i, \\ \xi_i \geq 0, \end{cases} \quad (2.4b)$$

where C is a positive constant. Introducing the slack variable to the equation adds some error to the classification, but it makes the classes linearly separable.

2.3.2 Kernel Trick

A linear classifier may not be adequate for many real-life applications. In such cases, a nonlinear optimal hyperplane may be more appropriate. Kernel trick is a preprocessing technique that maps input data to a higher dimensional space. If the mapping function is appropriate, the transferred feature space can be linearly separable. There are several kernel functions that can map the feature space to higher dimensions. The key to define a kernel is that

the function has to be in the form of a dot product between training samples and support vectors. Polynomial and Gaussian functions are two well-known examples of nonlinear kernels. The polynomial kernel can be expressed as [13]:

$$k(x_i, x) = (1 + x_i x)^P \quad (2.5)$$

in which x is a support vector and P is the order of the polynomial function. The following is the definition of Gaussian kernel [13]:

$$k(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \quad (2.6)$$

where γ is a positive constant.

It is usually preferred to solve the dual of the optimization problem (2.4) rather than the primal. Considering the kernel function, the dual problem of (2.4) can be expressed as:

$$\text{maximize } L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j k(x_i, x_j), \quad (2.7a)$$

$$\text{s.t. } \begin{cases} \sum_{i=1}^l \alpha_i y_i = 0, \\ \alpha_i \in [0, C], \end{cases} \quad (2.7b)$$

in the above equation, α_i is Lagrange multiplier. Some of Lagrange multipliers can be zero, while the others can be greater than zero. The non-zero multipliers determine the support vectors. The optimal hyperplane can be expressed as:

$$\sum_{i=1}^l \alpha_i k(x_i, x) + b \quad (2.8)$$

2.3.3 Multiclass Classification

Thus far, the SVM can be only used for binary classification. However, there are several methods that can extend the application of the SVM to multiclass classification. One-versus-all [14] is one implementation of a multiclass SVM. If n is the total number of classes, the one-versus-all SVM constructs n SVM models, for each model the training samples of one class are labelled +1, while the label assigned to all other classes is -1. If $\sum_{i=1}^l \alpha_i^j k(x_i^j, x) + b^j$ is the

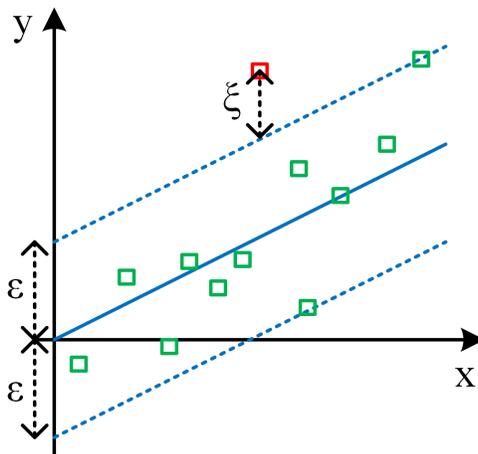


Figure 2.3: Support Vector Regression for estimating a linear function

j^{th} trained SVM model, the most probable hypothesis for the input x can be chosen by:

$$\Lambda = \arg \max_j \left(\sum_{i=1}^l \alpha_i^j k(x_i^j, x) + b^j \right) \quad (2.9)$$

in which Λ is the predicted class for the variable x .

One-versus-one [15] is another multiclass SVM method in which the method trains several models, while each model is trained using the data of only two classes. Therefore, each model can classify only two specific classes. The total number of trained models is $n(n-1)/2$. The voting decision for finding the class of a variable is the same as one-versus-all.

DAGSVM [16] is another multiclass SVM that trains the models similar to one-versus-one. However, the winning class is decided based on Directed Acyclic Graph (DAG) theory, not using the arg max function. This helps the method to be faster than one-versus-one. Besides, it provides some insight about the generalization error of the model where the other mentioned multiclass methods lack such capability.

2.3.4 Support Vector Regression

SVR is a modified version of the SVM that can be used for regression. Fig. 2.3 illustrates the fundamental idea behind the linear SVR in which each square represents one pair of training data set $(x_i, y_i), i = 1, \dots, l$. The objective is

to find a regression function $f(x)$ as smooth as possible, while the maximum deviation of the function from all training targets remains below ε . Therefore, a maximum error ε is allowed, and the area between two dashed lines in Fig. 2.3 is called ε -insensitive zone. This, however, might be infeasible (depending on the chosen value for ε) and some outliers (the red square in Fig. 2.3) might exist. This problem can be solved by allowing for some error in order to make the problem feasible. The objective function (2.4) can be modified in the following way to be used for regression [17]:

$$\text{minimize } J(\omega) = \frac{1}{2}\|\omega\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*), \quad (2.10a)$$

$$\text{s.t. } \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i, \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (2.10b)$$

and the linear regression model can be stated as:

$$f(x) = \langle \omega, x \rangle + b \quad (2.11)$$

Training samples within the ε -insensitive zone add no error to the regression model ($\xi_i = \xi_i^* = 0$). In contrast, ξ_i and ξ_i^* are greater than zero for training data located outside of ε -insensitive zone. These outliers are called Support Vectors. Considering the nonlinear kernel, the dual problem of (2.10) can be written as:

$$\text{maximize } L(\alpha) = -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i(\alpha_i - \alpha_i^*) \quad (2.12a)$$

$$\text{s.t. } \begin{cases} \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \\ \alpha_i, \alpha_i^* \in [0, C], \end{cases} \quad (2.12b)$$

Finally, we can express the regression model as:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*)k(x_i, x) + b \quad (2.13)$$

in which $\alpha_i - \alpha_i^*$ is nonzero if x_i is a support vector. The polynomial and Gaussian kernels that described for classification can be used for nonlinear regression as well.

2.3.5 Training SVM Models

One needs to solve the quadratic programming problem (2.7) for classification and (2.12) for regression to train a support vector model. Depending on the size of training data, the size of this quadratic programming problem and memory requirements for saving it on a computer could become so immense that prohibits the use of standard quadratic solvers. To overcome this problem, several support vector learner algorithms are proposed to solve (2.7) and (2.12). One of these algorithms is SVM^{light} [18] which decomposes the original quadratic problem into a series of smaller working sets and solves those working sets sequentially. Therefore, memory consumption is curtailed considerably.

Sequential minimal optimization [19] is another support vector trainer that avoids numerical quadratic optimization by decomposing the original problem into several small quadratic programming sub-problems. Each of these small quadratic programming sub-problems involves optimizing only two Lagrange multipliers. Therefore, it can be solved analytically. The sub-problems are solved sequentially, and the algorithm continues until all Lagrange multipliers are optimized.

2.3.6 Validation and Hyperparameter Optimization

K-fold cross-validation is a common procedure for validating machine learning models, in which the training dataset is partitioned into k randomly chosen subsets or folds. A single fold is held out for testing the accuracy while the rest of the folds are used for training. This procedure is repeated for all other folds and the overall accuracy of the model is calculated by averaging the accuracy of k folds. There are different choices for the value of k, but 10-fold cross-validation is more common.

Every machine learning model has a set of parameters that their values influence the accuracy of the trained model. These parameters are called hy-

perparameters. For example, C , γ , and ε are the hyperparameters of Support Vector Machine. Finding the optimal values for hyperparameters so that the trained model is the most accurate is called hyperparameter optimization or tuning. One method for hyperparameter optimization is grid search. In grid search, a set of values is assigned to each hyperparameter (C, γ, ε) , and the support vector model is trained multiple times, each time using one combination of the values of parameters. Finally, the combination that results in the highest k-fold cross-validation accuracy is chosen.

Random search is another hyperparameter optimization that instead of enumeration which is used in grid search, randomly selects hyperparameter values and evaluates the output. If the number of hyperparameters in a learning algorithm is small, random search can be faster than grid search [20].

2.4 Naive Bayesian Classifier

The Naive Bayesian classification is a probabilistic supervised machine learning model, designed based on Bayes' theorem of conditional probability. The reason the classifier is called naive is that the classifier assumes predictors are independent, so there is no correlation between them. However, the independence assumption is fairly strong and does not hold for many real-life or artificial applications, as features could be correlated. The independence assumption makes the NB model easy to build as it does not require costly iterative learning algorithms. As a result, the NB classifier is highly scalable and can be applied to large datasets.

Suppose Λ_k is a class in a set of classes $k = 1, \dots, K$. Based on Bayes' theorem, the posterior probability of class Λ_k with respect to the feature F_i , $i = 1, \dots, n$ can be expressed as:

$$p(\Lambda_k|F_i) = \frac{p(F_i|\Lambda_k)p(\Lambda_k)}{p(F_i)} \quad (2.14)$$

in which $P(\Lambda_k|F_i)$ is the posterior probability, $p(F_i|\Lambda_k)$ is likelihood, $p(\Lambda_k)$ is the prior probability of class Λ_k , and $p(F_i)$ is the prior probability of the

predictor. The joint probability model for Λ_k considering all predictors is [21]:

$$p(\Lambda_k|F_1, \dots, F_n) = \frac{p(F_1, \dots, F_n|\Lambda_k)p(\Lambda_k)}{p(F_1, \dots, F_n)} \quad (2.15)$$

Since the denominator doesn't depend on Λ_k and the values of predictors F_i are given, the probability model can be rewritten and expanded to [21]:

$$\begin{aligned} P(\Lambda_k, F_1, \dots, F_n) &= p(\Lambda_k)p(F_1, \dots, F_n|\Lambda_k) \\ &= p(\Lambda_k)p(F_1|\Lambda_k)p(F_2, \dots, F_n|\Lambda_k, F_1) \\ &= p(\Lambda_k)p(F_1|\Lambda_k)p(F_2|\Lambda_k, F_1)p(F_3, \dots, F_n|\Lambda_k, F_1, F_2) \\ &= p(\Lambda_k)p(F_1|\Lambda_k)p(F_2|\Lambda_k, F_1)p(F_3|\Lambda_k, F_1, F_2)p(F_4, \dots, F_n|\Lambda_k, F_1, F_2, F_3) \end{aligned} \quad (2.16)$$

and so on. Features can be mutually correlated in the above equation. Considering the conditional independence assumption:

$$p(F_i|\Lambda_k, F_j) = p(F_i|\Lambda_k), \quad i \neq j \quad (2.17)$$

Therefore, (2.16) can be rewritten as:

$$\begin{aligned} P(\Lambda_k, F_1, \dots, F_n) &= p(\Lambda_k)p(F_1|\Lambda_k)p(F_2|\Lambda_k)\dots p(F_n|\Lambda_k) \\ &= p(\Lambda_k) \prod_{i=1}^n p(F_i|\Lambda_k) \end{aligned} \quad (2.18)$$

Hence, the conditional distribution over the class Λ_k is:

$$P(\Lambda_k|F_1, \dots, F_n) = \frac{1}{Z} p(\Lambda_k) \prod_{i=1}^n p(F_i|\Lambda_k) \quad (2.19)$$

where Z is a constant scaling factor, dependent only on F_1, \dots, F_n . The naive Bayes classifier combines the Bayes conditional probability model (2.19) with a decision rule to construct a classifier. A common decision rule for the NB classifier is $\arg \max$. Using this decision rule, the NB classifier can be expressed as [21]:

$$CLASS(f_1, \dots, f_n) = \arg \max_{k=1, \dots, K} p(\Lambda_k) \prod_{i=1}^n p(F_i = f_i|\Lambda_k) \quad (2.20)$$

where f_i is an observation.

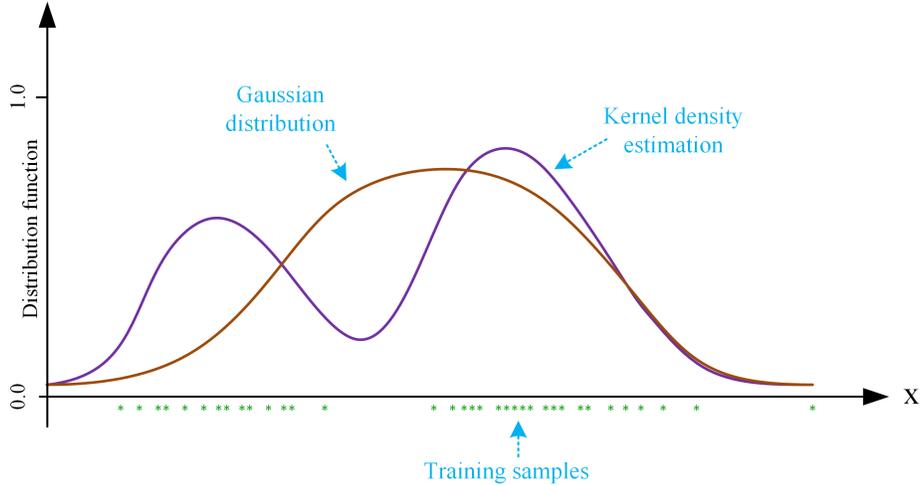


Figure 2.4: Comparing the effectiveness of kernel density estimation and Gaussian distribution in estimating the probability density of training data

2.4.1 Dealing with Continuous Data

The Naive Bayes deals with binary and numeric data differently. If f_i is binary (discrete), the output of $p(F_i = f_i|\Lambda_k)$ is a real number between 0 and 1. However, if f_i is numeric (continuous), it is modeled by a continuous probability distribution. A typical assumption is that the values associated with each class have Gaussian distribution as follows:

$$p(F_i = f_i|\Lambda_k) = g(f_i; \mu_{\Lambda_k}, \sigma_{\Lambda_k}) \quad (2.21)$$

in which μ_{Λ_k} and σ_{Λ_k} are the mean and variance of Gaussian distribution, respectively. This, however, could result in poor performance for the NB classifier, since the data of many applications is not normally distributed.

One solution to this problem is to use kernel density estimation [22] instead of Gaussian distribution. In this method the probability density of training data is estimated by averaging over a large set of Gaussian kernels as follows [22]:

$$p(F_i = f_i|\Lambda_k) = \frac{1}{n} \sum_{i=1}^n g(f_i; \mu_i, \sigma_{\Lambda_k}), \quad \mu_i = f_i \quad (2.22)$$

this provides more accurate estimation of the probability distribution of training data. Fig. 2.4 shows the effectiveness of kernel density estimation in estimating the probability distribution of some training samples.

2.4.2 Optimality of The Naive Bayes Classifier

Despite assuming conditional independence of predictors, the NB classifier can surprisingly represent performance competitive to more advanced machine learning models in many cases. A good explanation for such unexpected good performance is provided in [23] where it is explained that the NB classifier can perform well if the dependences are distributed evenly or if dependences cancel each other out.

Chapter 3

Volt-Var Optimization of Distribution Systems

This chapter presents a Support Vector Regression based Model Predictive Control for Volt-Var Optimization of distribution systems. The proposed technique is not rule-based, neither it uses circuit model simulations. The proposed VVO method builds an SVR model of the distribution system using measurement data from AMI. The trained regression model is then employed by MPC in a closed-loop control scheme to control capacitor banks and tap changers of the distribution system so that the power loss is minimized, and voltage profiles are maintained within a specific range. The simulation results of applying the proposed SVR-based MPC to IEEE123 bus test feeder proves that despite its measurement-based feature, the proposed approach is capable of providing close to optimal solutions to the VVO problem.

3.1 Literature Review

The main objectives of the Volt-Var Optimization (VVO) are to minimize power loss in the distribution system and keep voltage profiles at acceptable levels under different load conditions. The VVO has become an important feature of modern smart grids in which the VVO scheme monitors daily operation of the distribution system and generates proper control decisions. Those control decisions are then sent to capacitors and On-Load Tap Changing (OLTC) transformers in order to lower power loss and improve voltage levels.

Most of the existing VVO methods are either rule-based techniques or model-based approaches. Rule-based methods provide sub-optimal solutions to the VVO problem. In these methods, simple sets of rules are employed to decide the ON/OFF status of capacitors and tap ratios of OLTCs. For instance, the controller sends ON/OFF signals to capacitors or increment/decrement signals to OLTCs if for a certain amount of time, the measured voltage at the substation falls out of a predefined bandwidth. A conventional rule-based technique for maintaining the voltage profile of the distribution system using OLTCs is line-drop compensation. During heavy load situations, a voltage regulator boosts the voltage by changing the tap ratio of the OLTC, while voltage is boosted the least during light load conditions. The voltage regulator uses Thevenin equivalent impedance of the whole distribution system or a part of it to decide how much voltage compensation should be done. Line-drop compensation can be performed in two ways: load centre and voltage spread. In the load centre method the impedance is chosen to regulate the voltage at a given point downstream of the regulator, while in the other method the impedance settings are chosen to keep the voltage within a predefined range when operating from light load to full load [3].

In [24], the sets of rules are improved; If the change in the measured current of the feeder is more than a pre-set threshold, a new voltage bandwidth is calculated based on current load levels, and the settings of OLTCs are updated according to the new voltage bandwidth. Also, a simple control strategy governs the switching of capacitors; they are turned on if reactive power consumption measured at the substation exceeds a certain level and switched out if the consumed reactive power falls below a certain threshold.

Thanks to the simple rules involved and minimum measurement requirements, the implementation of rule-based schemes is relatively easy. This, however, could result in control decisions far from optimal as these methods are not equipped with precise optimization strategies. Rule-based methods cannot effectively minimize power loss and most of them are only suitable for controlling radial distribution systems. Besides, these approaches are not designed to deal with the integration of Distributed Energy Resources into distribution

grids.

Model-based VVO techniques rely on a model of the distribution system to make control decisions. Here, "Model" refers to computer simulations based on power flow equations or their approximate versions. For example, simulations based on linearized power flow or using DistFlow [24] which can describe power flow equations for radial systems.

The model-based VVO has been extensively studied in literature. The VVO in [25]-[26] is described as a mixed-integer programming problem and solved using analytical optimization algorithms. A modified genetic algorithm is used in [27] to solve the VVO problem. Particle Swarm Optimization is another example of a metaheuristic algorithm that is applied in [28]-[29] to solve the Volt-Var problem. Satisfactory results have been reported in both papers while [29] employs high-performance computing to parallelize PSO in order to speed up the optimization procedure and in turn, improve the scalability of the algorithm. Model Predictive Control is an advanced closed-loop control method that predicts the future response of the system under control using an explicit model, and makes its control decisions by solving a constrained optimization problem [30]. The application of MPC in Volt-Var optimization is studied in [31]-[32]. Both mentioned papers use DistFlow for prediction, so it limits their application to only radial distribution grids. Both studies, however, have reported promising results in applying MPC to the VVO problem in the presence of renewable energy resources.

Difficulties in modelling unbalanced and voltage dependent loads are common in model-based VVO algorithms [33]. Some of these methods are not designed to control systems retrofitted with DERs. Moreover, most of these methods (for example [25], [31] and [32]) can only solve the VVO problem for radial distribution grids. However, the main obstacle for actual implementation of model-based VVO techniques is the unavailability of precise power flow models of distribution systems. The power flow model is built based on physical configuration of the distribution grid and its parameters with the assumption that such data is accurate. In many primary and secondary distribution grids this is not the case [34], due to the fact that power utilities

have limited information about their distribution networks. Vast numbers of distribution feeders, limited data on secondary networks and unbalanced loads are factors that contribute to such limited knowledge of distribution data [33].

Advanced Metering Infrastructure is a key component of smart grid that gathers different types of measurements including voltage and power measurements of customer across the grid. The availability of such data allows for another type of power system analytics called data-driven approaches in which measurement data is fed to statistical or machine learning algorithms to build a data-driven model of the distribution system. Data-driven models do not suffer from inaccuracy of the power flow model or limited data availability of grid topology as they are entirely measurement-based and do not require circuit-based computer simulations.

An example of a data-driven approach is [35] where linear regression is used to build a linearized model of the distribution network. The effectiveness of the SVR in modelling power flow equations of distribution systems is proven in [34]. The nonlinear kernel used in the mentioned study allows the SVR model to handle the nonlinearity of system power flow and to incorporate active controllers of DERs in the regression model. Data-driven models presented in [35] and [34] have a strong potential to be employed in the VVO problem. Such potential has been comprehensively reviewed in [36] from technical and security points of view.

The potential of data-driven models in modelling the distribution system is exploited in this chapter to introduce a novel data-driven VVO approach. The proposed method is based on MPC in which the predictive control predicts the future behavior of the distribution system employing an SVR model. The regression model is built using measurement data of a few days of operation of the system, collected from AMI. Once the model is trained, MPC employs it to control the system.

3.2 The Proposed VVO Scheme

The power flow equations of a power system can be described as follows:

$$P_i = V_i \sum_{k=1}^N V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad (3.1)$$

$$Q_i = V_i \sum_{k=1}^N V_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad (3.2)$$

where N is the number of buses in the system and $i = 1, \dots, N$. P_i and Q_i are the active and reactive power injections at bus i . V_i represents the amplitude of the voltage of bus i , and θ_{ik} is the phase angle difference between buses i and k . $G_{ik} + jB_{ik}$ is the element at the i^{th} row and k^{th} column of the admittance matrix. Knowing the amount of loads consumed in the grid at a given time t , along with having the total active power measured at the main feeder, one can calculate the total loss of the distribution system as follows:

$$P_{\text{loss}}(t) = P_{\text{substation}}(t) - \sum_{i=1}^{N_L} P_i(t) \quad (3.3)$$

in which $P_{\text{feeder}}(t)$ is the active power measured at the feeder and N_L is the total number of phases with installed loads. The reason behind measuring each phase of a multi-phase bus separately is the presence of unbalanced loads in the system.

Circuit-based methods employ (3.1)-(3.3) to optimize Volt-Var operation of distribution systems. Due to the reasons mentioned in the previous section, such circuit models could be inaccurate. Using inaccurate models results in poor solutions, far from optimal, for the VVO problem. This paper proposes an SVR-based MPC approach as an alternative for circuit-based VVO algorithms, in which a data-driven model replaces the circuit model (3.1)-(3.3). The SVR model is trained using measurement data from AMI. Building a model using online measurements allows the proposed algorithm to capture the most recent operational conditions of the distribution grid. Hence, it does not suffer from the inaccuracy caused by an outdated circuit model. The collected data from AMI includes the amplitude of voltages as well as real and reactive powers of

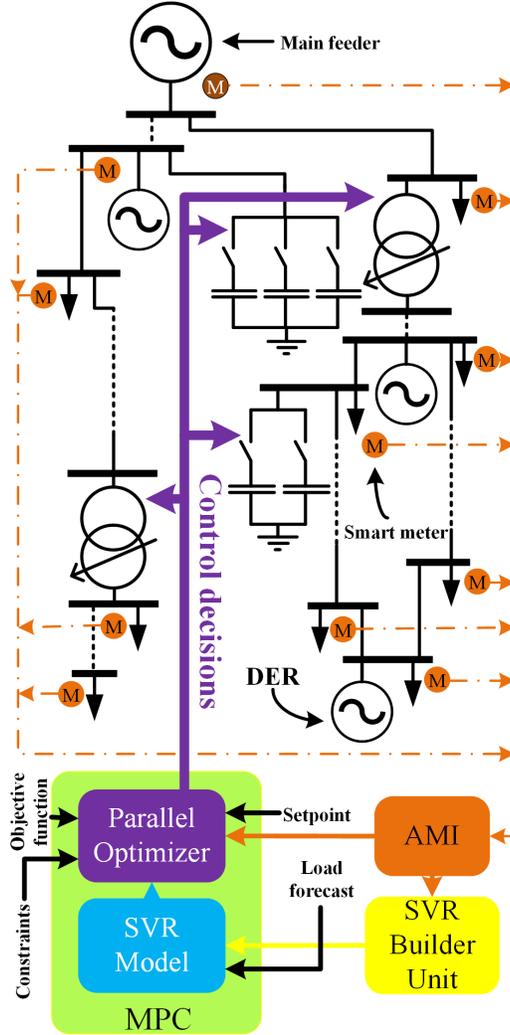


Figure 3.1: The overall scheme of the proposed SVR-based MPC

loads. This paper assumes that each load in the system is equipped with a smart meter, so the voltage amplitude and consumed power of all loads are sent to the proposed control algorithm at 15-minute intervals through AMI. AMI measurements are used for training an SVR model, as well as the operation of the SVR-based predictive controller.

Fig. 3.1 represents the general scheme of the proposed SVR-based MPC. The first step in the proposed control scheme is to build an SVR model of the system. This is performed by the SVR builder unit in which the measurement data from AMI, tap positions of OLTCs and reactive power injections of capacitor banks are fed to the unit as the inputs. The output of the SVR

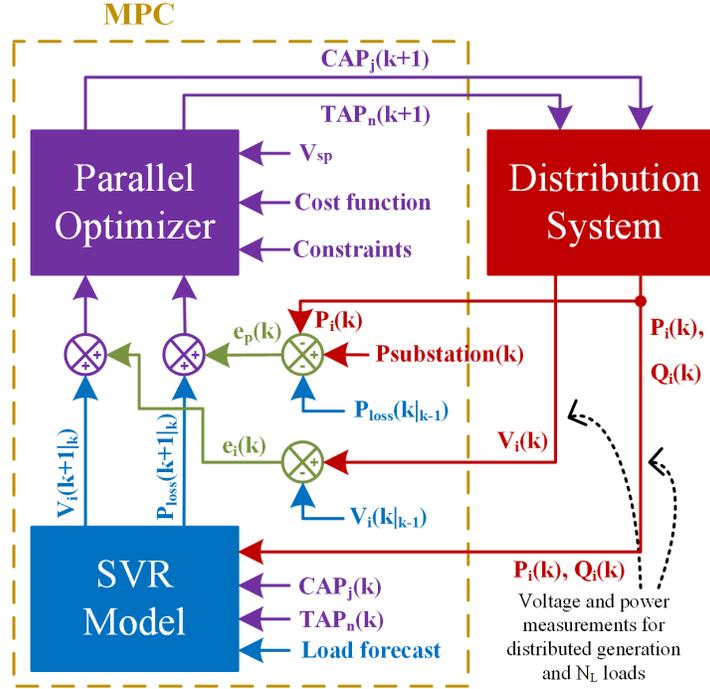


Figure 3.2: Block diagram of MPC for the VVO of a distribution system. MPC generates control decisions for control step $(k+1)$ based on measurements at control step (k)

builder unit is a nonlinear regression model that can predict the overall power loss and voltage amplitudes of load buses of the distribution grid. This regression model will then be utilized by MPC to predict the behavior of the system. MPC is a closed-loop controller that runs at periodic intervals T_s (15 minutes in this study) and generates its control decisions by solving a constrained optimization objective function, aiming to reduce power loss and to keep voltage profiles of load buses close to the setpoint value V_{sp} . These decision controls are then sent to capacitor banks and OLTCs.

Fig. 3.2 shows the operational procedure of the proposed scheme in more detail. As can be seen, MPC uses AMI power and voltage measurements at the control step k (the minute $t = kT_s$ of the simulation) to generate control decisions for capacitors $CAP_j(k+1)$ and tap changers $TAP_n(k+1)$ for the next control step $k+1$ (the minute $(k+1)T_s$). Consumed power measurements and the operational status of capacitors and OLTCs at the control step k are fed to the SVR model to predict voltages $V_i(k+1|k)$ and power loss $P_{loss}(k+1|k)$

of the system for the next control step. A parallel PSO is employed to solve the optimization problem required for generating control decisions. This high-speed optimizer ensures that the entire decision-making process is done in real-time.

Error feedback is an important feature of MPC which allows for compensating prediction errors of the model. Fig. 3.2 shows how voltage error feedback e_i and power loss feedback e_p are calculated and added to the output of the SVR model in order to enhance the prediction. This improves the quality of the generated control decisions significantly. Also, if short-term customer load forecasts are available, MPC can predict the behavior of the system over a longer horizon, therefore it can make more optimized control decisions. This, along with other operational aspects of the proposed scheme, is discussed in more depth in the next section.

3.3 The Framework of SVR-based MPC

This section provides details of SVR-based MPC. The first part of the section describes how the SVR builder unit trains a regression model using AMI measurements. The second part of the section depicts MPC formulation and implementation. Finally, the third and fourth part of this section discuss why a GPU-based PSO is used for handling the required optimization and how it is implemented on GPU.

3.3.1 Building An SVR Model of the Distribution Grid

As mentioned in section 2.1, capacitor banks and OLTCs are installed to improve the performance of the distribution system, whereas protective devices such as overcurrent relays and circuit breakers ensure the safe operation of the system. The objective of the VVO study is to optimize the operation of the system under normal conditions. The assumption of the normal operational condition implies that protective apparatus does not need to be involved in this type of study. Therefore, the VVO study can be performed by studying the dynamics between capacitors, OLTCs, loads, and DERs.

The SVR builder unit is responsible for building an SVR model of the system. This unit uses active and reactive load consumptions measured by AMI, measured power of the main feeder, kVAR injections of capacitor banks, tap positions of OLTCs, and active and reactive DER generation (in case DERs are installed) as its input training data (vector x in (2.12)). The target training data (vector y in (2.12)) are total power loss calculated using (3.3) and the measured voltage amplitudes of phases with installed load. The SVR builder unit employs SVM^{light} algorithm [18] to train an SVR model. The outputs of the training are $\alpha_i - \alpha_i^*$ and x_i for $i = 1, \dots, l$ and b (as in (2.13)) for power loss and the voltage of each load bus.

To train an SVR model, data representing the behavior of the system is required. This means capacitor banks and OLTCs must be switched randomly during the training stage. Random selection of reactive injections of capacitors and tap positions of OLTCs, along with ever changing customer load and DER generation profiles, ensures good generalization of the model. This means the SVR model can adequately capture the underlying relationship between the input and target training sets. Of course, all operational constraints of capacitor banks and OLTCs must be satisfied when these random changes are happening. This guarantees the safe operation of the distribution system during the training period.

It is assumed that measurements across the distribution system are performed in 15-minute intervals. Therefore, AMI can send 96 measurement samples per day to the SVR builder unit. The number of samples (or the number of days) required for the training process must be chosen carefully to avoid underfitting or overfitting of the regression model. Section. 3.4 provides analysis about how the selection of the size of the training set can affect the accuracy of the regression model.

The training process happens once at the beginning of the control procedure and the SVR model is then sent to MPC. This is an offline training, which means the algorithm waits for all required measurement samples to be collected, and then trains the SVR model. Therefore, MPC cannot control the system during the training stage. As mentioned earlier, varying input lev-

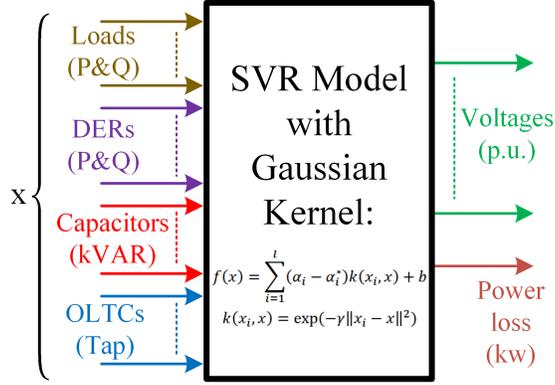


Figure 3.3: Inputs and outputs of the trained SVR model

els during the training stage ensure the generalization of the model. So, the trained model is able to accurately predict the output as long as the inputs vary in the same range as the training data. However, if a topology change such as adding a new DER or switch reconfiguration occurs, the model needs to be retrained. In this case the SVR builder unit can be recalled to generate a new regression model. The retraining procedure would be the same as the initial training of the SVR model. Fig. 3.3 shows the inputs and outputs of the SVR model. It is worth mentioning that the SVR model only predicts the voltages of load buses and omits the intermediate buses with no load or smart meter installed. This is not a problem because controlling the voltages of intermediate buses is not the objective of the VVO. However, if controlling the voltages of those buses is desired, SVR is able to estimate those voltages too as proven in [34].

3.3.2 Model Predictive Control

MPC is a finite-horizon optimal control scheme that runs at periodic intervals T_s (15 minutes in this research), aiming to optimize the process behavior and keep the output variables close to their reference setpoints over the prediction horizon T_p steps ($T_p * T_s$ minutes). The prediction is done using a model of the system, and control decisions for the control horizon T_c steps ($T_c * T_s$ minutes) where $T_c \leq T_p$ are generated by solving a constrained objective function.

Only the first control sequence (generated for the current control step) is sent to the system while the rest are discarded, and the entire control process is repeated in the next control interval[30]. MPC is very versatile in selection of the prediction model, optimization objectives, operational constraints, and the optimization algorithm. Such flexibility has resulted in introducing several variants of MPC. In [37]-[38], for example, MPC approaches based on SVR models have been studied. A generic objective function that MPC employs for making control decisions is as follows:

$$\begin{aligned} \min J(\Delta(u)) = & w_1 \sum_{j=1}^{T_p} |y_p(k+j|_k) + \beta e(k) \\ & - y_{sp}(k+j)| + w_2 \sum_{j=1}^{T_c} |\Delta u(k+j-1)| \end{aligned} \quad (3.4)$$

where k is the current interval, $y_p(k+j|_k)$ is the prediction model output for the minute $t = (k+j)T_s$, predicted at the current time $t = kT_s$ (k^{th} control step), and $\Delta u(k+j-1)$ is the decision control made at the current control step, to be applied at the minute $(k+j-1)T_s$. The prediction error $e(k)$ is equal to $y(k) - y_p(k|_{k-1})$ where y is the measured output. $(w_1, w_2) \geq 0$ and $0 \leq \beta \leq 1$ are constants. The flexibility of MPC allows us to tailor an objective function suitable for the VVO problem. The aim of such objective function would be to minimize voltage deviations of load buses from a setpoint voltage as well as minimize power loss in the system. The VVO objective function is described below:

$$\begin{aligned} \min VVO(\Delta Cap_i, \Delta Tap_i) = & w_1^i \sum_{j=1}^{T_p} \sum_{i=1}^{N_L} |V_i(k+j|_k) \\ & + \beta_1 e_i(k) - V_{sp}(k+j)| + w_2 \sum_{j=1}^{T_p} (P_{loss}(k+j|_k) \\ & + \beta_2 e_p(k)) + w_3^i \sum_{j=1}^{T_c} \sum_{i=1}^{N_{cap}} |\Delta Cap_i(k+j-1)| \\ & + w_4^i \sum_{j=1}^{T_c} \sum_{i=1}^{N_{tap}} |\Delta Tap_i(k+j-1)| \end{aligned} \quad (3.5)$$

in which $\Delta Cap_i(k+j-1)$ is the change in the injected power of the i^{th} capacitor bank to be applied at $t = (k+j-1)T_s$, and $\Delta Tap_i(k+j-1)$ is the tap ratio change of the i^{th} OLTC for $t = (k+j-1)T_s$. Both of these control variables are integers. This means they represent step changes in the values of KVAR of capacitor banks and secondary tap ratios of OLTCs, respectively, not the actual KVAR or secondary ratio. N_L is the total number of loads, N_{cap} the total number of capacitor banks, and N_{tap} is the total number of OLTCs in the grid. $V_i(k+j|k)$ and $P_{loss}(k+j|k)$ are the voltage of the i^{th} bus (in p.u.) and the total power loss of the system respectively, both predicted by the SVR model for $t = (k+j-1)T_s$.

w_1^i, w_2, w_3^i and w_4^i are non-negative coefficients, and V_{sp} (p.u.) represents the setpoint voltage for all buses.

The VVO objective function would not be complete without operational constraints. The operational constraints considered in this study are as follows:

$$V_{min} \leq V_i(k+j|k) \leq V_{max} \quad (3.6)$$

$$|\Delta Cap_i| \leq Cap_{max} \quad (3.7)$$

$$|\Delta Tap_i| \leq Tap_{max} \quad (3.8)$$

$$0 \leq CAP_i + Cap_i^{step} \Delta Cap_i \leq CAP_i^{max} \quad (3.9)$$

$$0 \leq TAP_i + \Delta Tap_i \leq TAP_i^{max} \quad (3.10)$$

$$\sum_{j=1}^{T_{cap}} (|\Delta Cap_i(k-j+1)| > 0) \leq 1 \quad (3.11)$$

$$\sum_{j=1}^{T_{tap}} (|\Delta Tap_i(k-j+1)| > 0) \leq 1 \quad (3.12)$$

constraint (3.6) makes sure voltage amplitudes remain within an acceptable range while (3.7)-(3.8) limit the changes in reactive injection of capacitors and tap ratios. Constraint (3.9) controls the total kVAR of capacitors, in which CAP_i^{max} is the total installed kVAR of the i^{th} capacitor bank and CAP_i is its online kVAR at the previous control step. Cap_i^{step} denotes kVAR rating of each capacitor in the i^{th} capacitor bank (it is assumed all capacitors of a capacitor bank have the same rating). Similarly, (3.10) limits the tap ratio of the i^{th} OLTC where TAP_i^{max} and 0 correspond to the highest and lowest secondary

ratios, respectively. Constraints (3.11)-(3.12) can be used to reduce physical wear and tear of devices. They limit the times a device can be switched, to once per T_{cap} control steps for capacitors and once per T_{tap} for OLTCs. $|\Delta Cap_i| > 0$ and $|\Delta Tap_i| > 0$ in the mentioned equations are actually binary variables that become true when Cap_i or Tap_i changes, and stay false when there is no change.

The optimization algorithm that solves (3.5) provides ΔCap_i and ΔTap_i for the SVR model (the optimization algorithm will be discussed later in this section). The rest of the inputs of the SVR model are measurements from AMI. Assuming that changes in load profiles of the system and generation profiles of DERs (if there are any) are not significant between two successive control intervals, one can use measurements at the minute $t = kT_s$ as the inputs of the SVR model to predict the output for the next control interval at $t = (k+1)T_s$. But this assumption does not hold if the prediction horizon is longer than one control step. Extending the prediction horizon requires short-term load forecast over the prediction horizon so that the SVR model can predict the output for $t = (k+j)T_s$ based on the forecasted load for $t = (k+j-1)T_s$. Short-term load forecasting is out of the scope of this paper and it is assumed that such forecasts are available if MPC needs them. A couple of studies for short-term load forecasting can be found in [39]-[40].

Choosing a long prediction horizon is not mandatory for good performance of MPC and the controller can actually do quite well with $T_p = 1$ step (15 minutes). However, extended prediction and control horizons enable MPC to enforce constraints like (3.11)-(3.12) more effectively. Such constraints set a limit on the number of times control decisions can be made, and by predicting the response of the system over a longer horizon, MPC can find the best moments to activate control variables.

Voltage and power loss prediction errors are calculated as $e_i(k) = V_i(k) - V_i(k|_{k-1})$ for $i = 1, \dots, N_L$ and $e_p(k) = P_{\text{loss}}(k) - P_{\text{loss}}(k|_{k-1})$, respectively. β_1 and β_2 ($0 \leq \beta_1, \beta_2 \leq 1$) are coefficients that control how much error feedback is added to voltages and power loss predictions, respectively. For example, $\beta_2 = 0$ means no compensation while $\beta_2 = 1$ means full error compensation

for power loss. Incorporating error feedbacks (error feedback and error compensation are used interchangeably in this paper) into the control procedure is a powerful feature of MPC as it allows for the compensation of model inaccuracy. Several simulations in Section 3.4 demonstrate how this closed-loop feature can improve the performance of MPC.

3.3.3 Need for A Fast Optimizer

V_i and P_{loss} in the VVO objective function (3.5) are the outputs of the nonlinear SVR model (2.12)-(2.13). This means that (3.5) is nonlinear as well. A suitable optimization algorithm for our MPC implementation would be an algorithm able to solve the nonlinear objective function (3.5), capable of handling constraints (3.6)-(3.12), and most importantly, able to provide a solution in a reasonable time, so that MPC can be implemented in a real-time manner.

A parallel GPU-based PSO is employed in this study in order to speed up the optimization process and make sure our proposed MPC can keep pace with real-time applications. Parallel PSO has shown great performance in [29] for solving the VVO problem. Utilizing a GPU-based PSO has several advantages. One is that PSO does not require numerical differentiation while dealing with the SVR model, and the model is treated as a black box. The second advantage is that a parallel PSO increases the scalability of the control scheme. Thus, it makes the proposed algorithm capable of controlling large-scale distribution systems. This is important in real-time applications of the control algorithm as the controller is supposed to provide control decisions before the next control step arrives. Time delays associated with physical actuation of capacitors and OLTCs must be considered as well, and if security concerns are present, it is preferable that the VVO controller provides control decisions within one minute [28].

The third advantage of employing a GPU-based optimizer lies in the nature of the mathematical operations required for calculating the output of the SVR model. Calculating (2.12)-(2.13) includes dot product, matrix-vector multiplication and reduction operations, all well-suited for GPU implementation. This adds another level of parallelism to a GPU-based optimizer, which ensures the

proposed MPC is suitable for real-time applications.

3.3.4 GPU-based Optimizer

PSO is a metaheuristic optimization algorithm that works based on share of knowledge between particles in a swarm. It was first introduced in 2005 by Kennedy and Eberhart [41] and later updated to its standard form in [42]. The PSO used in this study is a parallel version of the PSO proposed in [43], which is less prone to getting stuck in local optima, due to adding mutation to the position update procedure of particles. This PSO algorithm can be summarized in the following steps:

1. Initialize the position vector $X_i = [x_{i1}, \dots, x_{id}, \dots, x_{iD}]$ and velocity vector $V_i = [v_{i1}, \dots, v_{id}, \dots, v_{iD}]$ for $i = 1, \dots, n$, where $d = 1, \dots, D$ is the dimension of the optimization problem.
2. Evaluate fitness for each particle. Update $Gbest$ which is the best position ever encountered by the whole swarm. Update $Pbest_i$, the best position experienced by the i^{th} particle.
3. Update V_i and X_i in the following way for each particle:

$$v_{id} = \chi v_{id} + c_1 \times r_1 \times (Pbest_{id} - x_{id}) + c_2 \times r_2 \times (Gbest_d - x_{id}) \quad (3.13)$$

$$x_{id} = x_{id} + v_{id} \quad (3.14)$$

$$x_{id} = x_{id} \times (1 + g(0, \sigma)) \quad (3.15)$$

where c_1 and c_2 are accelerating coefficients, χ is the inertia weight, and r_1 and r_2 are two random numbers between 0 and 1. $g(0, \sigma)$ generates random numbers with Gaussian distribution with the variance σ .

4. Repeat steps 2 and 3 until the maximum number of iterations or another termination criterion is reached.

The objective function (3.5) is the fitness function that is evaluated for each particle at each iteration of PSO. For the sake of simplicity assume $T_p = T_c = 1$.

Therefore, $V_i(k+1|k)$ and $P_{loss}(k+1|k)$ will be the outputs of the SVR model that need to be calculated. We can re-write (2.12) in a matrix form as follows:

$$G = \exp(-\gamma \| X - SV^T \|^2) \quad (3.16)$$

in the above equation $X = [x_1, \dots, x_i, \dots, x_l]$ is the input to the SVR model where the first $D = N_{cap} + N_{tap}$ dimensions of X are variables that are optimized by PSO. In other words, they are the kVAR of capacitors and taps of OLTCs that are decided by PSO during its iterations. Dimensions representing tap positions are integer. Dimensions representing kVAR injections of capacitor banks are integer as well, however they change in Cap_{base} steps, in which Cap_{base} is the rating of each capacitor unit in a capacitor bank. For example, if $Cap_{base} = 50kVAR$, x_i can have values like 50 kVAR, 150 kVAR, *etc.* In order to make X integer, the output of (3.15) is rounded so that kVAR and tap values are forced to be integer. The rest of the $2 \times N_L$ dimensions of X are active and reactive load measurements provided by AMI. Therefore, the total dimension of X is $l = N_{cap} + N_{tap} + 2 \times N_L$. $SV_{v \times l}$ is a matrix comprised of v row-wise support vectors ($v \leq l$). Equation (3.16) can be expanded as follows:

$$G_{1 \times v} = \exp(-\gamma (\underbrace{\langle X_{1 \times l}, X_{1 \times l} \rangle}_{\text{first term}} - 2 \underbrace{X_{1 \times l} \times SV_{l \times v}^T}_{\text{second term}} + \underbrace{\langle SV_{l \times v}^T, SV_{l \times v}^T \rangle}_{\text{third term}})) \quad (3.17)$$

in which dot product is done column-wise for SV in the third term. Using (3.17), (2.13) can be expressed as:

$$f(X) = \langle \alpha_{1 \times v} - \alpha_{1 \times v}^*, G_{1 \times v} \rangle + b \quad (3.18)$$

in the above equation, $f(x)$ is the predicted output which could be power loss or the voltage amplitude of each load bus (each of these outputs has its own set of support vectors). In order to implement PSO on GPU, (3.17)-(3.18) have to be parallelized, besides the iterative procedure of updating particles.

Fig. 3.4 shows the basic architecture of a CUDA-enabled GPU. CUDA [44] is a programming language that allow general-purpose programming on

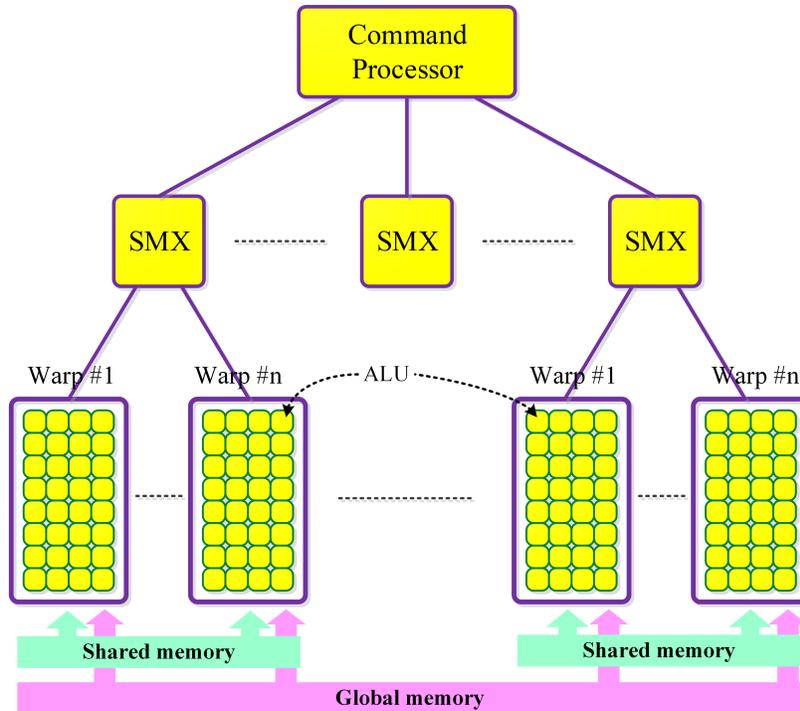


Figure 3.4: The architecture of a CUDA-enabled GPU

NVIDIA GPUs. A function that runs on a GPU is called a kernel and it is comprised of several threads. A command processor dispatches the threads of a kernel between Streaming Multiprocessors (SMXs). An SMX is a powerful processor that schedules and runs kernels. SMXs are autonomous and each can run a stream of instructions different than the others. Each SMX controls a few groups of Arithmetic-Logic Units (ALUs). ALUs are simple (less powerful and less flexible) processing units that can perform arithmetic and logical operations on data. Each group of ALUs is called a warp and contains 32 ALU units. A warp is designed based on Single-Instruction Multiple-Data (SIMD) architecture. It means ALUs of a warp can only execute a single instruction in parallel on a given clock cycle. If a piece of code is branched (there is a condition which is true for some threads in a warp, while it is false for the rest of them), the execution is done sequentially. The threads running within ALUs of a warp are always synchronized. The threads of a kernel that fit in a single SMX can be synchronized, unless warps of that SMX execute different instructions. If a kernel contains more than 1024 threads, or it is specified for

the compiler to run the kernel on more than one SMX, the compiler breaks the kernel into several blocks and the command processor assigns each block to an SMX to execute. The threads of a kernel that runs on multiple SMXs are not synchronized.

A GPU has an onboard global memory for storing data. This memory is accessible to all threads of a kernel. Data can be transferred between global memory and the RAM of a PC via a PCI port. The global memory is large enough to store big chunks of data. however, it is a dynamic memory and that makes it slow access. There is an alternative on-chip memory for each SMX called shared memory. This memory has small capacity (2×48 kB) and can be only accessed by threads running on that specific SMX. Shared memory is very fast and fetching data from it takes just a few cycles. An efficient parallel algorithm developed for a GPU should allow for breaking a big set of data into many small chunks, each can fit in the shared memory of an SMX. Otherwise, the computing capability of the GPU cannot be efficiently utilized. Besides memory considerations, one must develop parallel algorithms for a GPU, considering the SIMD architecture of it and data synchronization limitations.

The third term of (3.17) only involves support vectors. Hence, it can be calculated once the SVR model is trained and stored on the global memory of GPU. Equation (3.18) and the first two terms of (3.17) have to be calculated for each particle at each iteration of PSO. Equation (3.18) and the first term of (3.17) are dot products. Fig. 3.5 show a simple example of the GPU implementation of dot product of two vectors with a length of 16. The figure shows how dot product is done in parallel with a kernel of four threads that run on a single SMX. A block of only four threads is chosen for illustrative purposes. For real GPU implementation, the size of the kernel should not be less than 32, otherwise the rest of the threads run idle. It can be observed from the figure that the dot product is basically a type of reduction. Reduction is a primitive parallel operation in which a summary is obtained from a set of data, for example the minimum or the sum of a vector.

Each thread in Fig. 3.5 is represented by a specific color. During the first

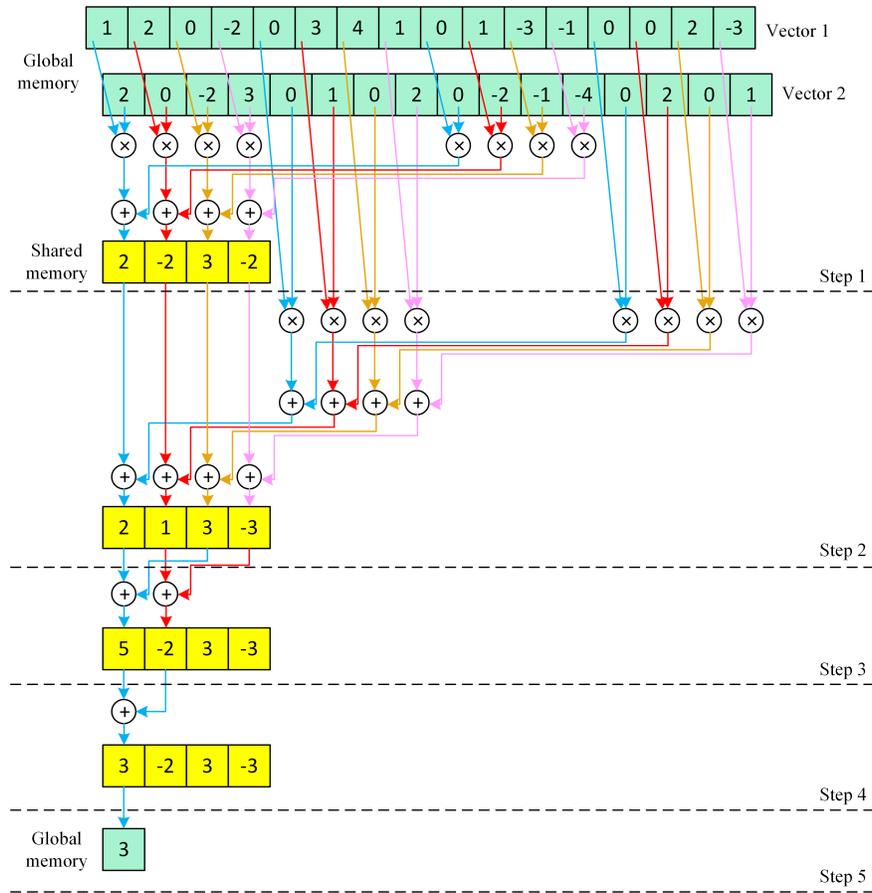


Figure 3.5: An illustrative example of parallel implementation of dot product on a GPU

and second steps, each thread fetches data from the global memory, performs two multiplications, adds them together, and adds the result to the value already written on the shared memory. During the third step, only half of the threads (2 threads) are active and each of them performs a sum operation on the data written on shared memory. At the fourth step, one thread performs the last sum operation and stores the final result on the first location of the shared memory. Finally, in the last step, the result of dot product is copied to the global memory.

The second term of (3.17) is a vector-matrix multiplication. This operation basically involves v times of dot product between X and each column of SV^T . Fig. 3.6 describes this procedure in which the vector-matrix multiplication kernel is split into v blocks. Each block performs dot product (the same as

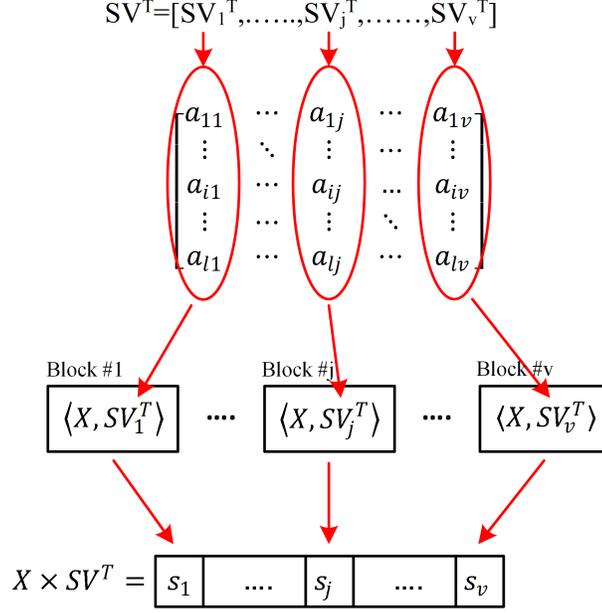


Figure 3.6: Parallel vector-matrix multiplication on GPU

Fig. 3.5 between X and one column of SV^T , and stores the result into the specified location on the global memory. Since a dot product between X and a given column of SV^T is not dependent to the results of dot product of other columns, all v blocks in Fig. 3.6 can be executed concurrently.

Fig. 3.7 show the optimization procedure of PSO implemented on a GPU. A GPU can run up to 32 streams concurrently. This means calculations for up to 32 particles can be performed in parallel. Therefore, the GPU-based PSO takes advantages of two levels of parallelism; one level is that the update procedure of up to 32 particles can be performed concurrently. The second level is that the operations required for the update procedure of particles (dot product, vector-matrix multiplication, *etc.*) are parallel themselves. Calculations of each term of (3.5) involves a sum operation that can be done using a reduction operation the same as the dot product in 3.5. Also, finding $Gbest$ requires finding the minimum of $Pbest$ of all particles. This is a reduction operation as well where the obtained summary is the minimum value of the $Pbest$ vector.

The constraints (3.6)-(3.12) are mainly handled by rejecting infeasible solutions. PSO starts the optimization procedure with finding at least one feasible

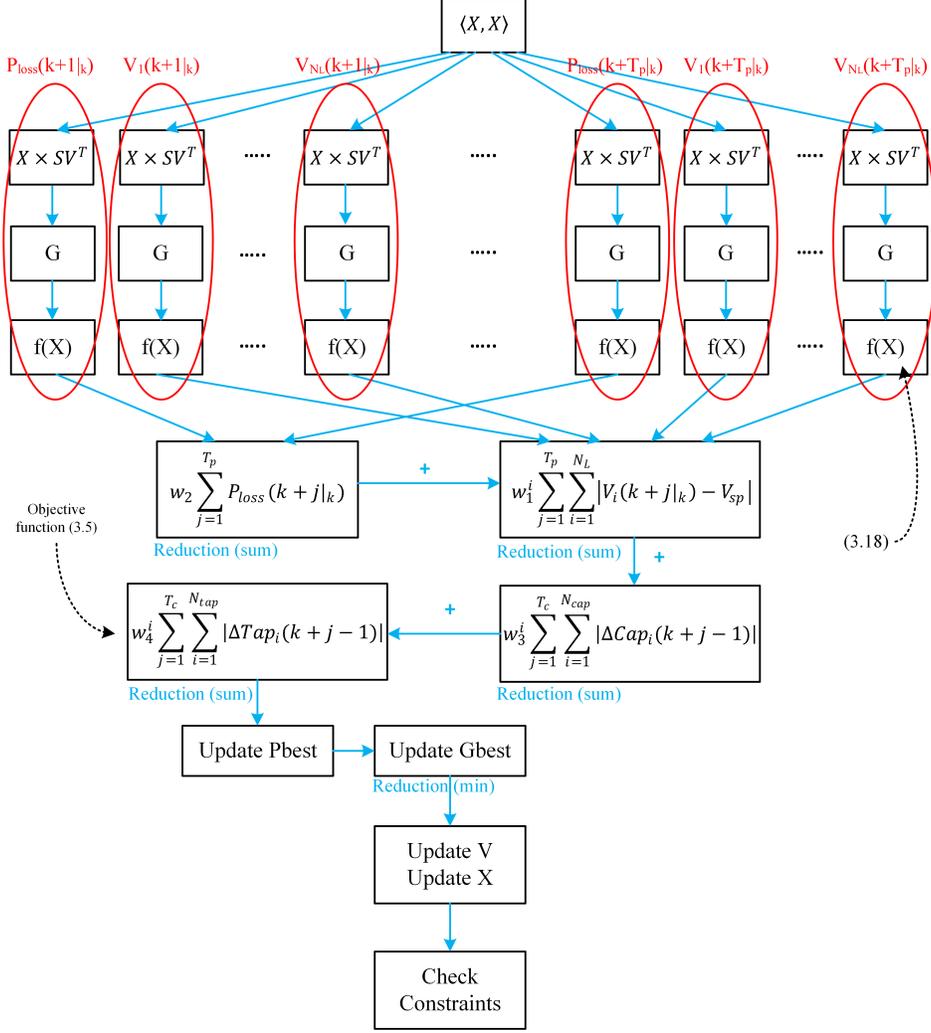


Figure 3.7: The update procedure of one particle at each iteration, implemented on a GPU

solution that satisfies all constraints. For the rest of the optimization procedure, only solutions that satisfy all constraints are kept, while the rest are discarded. In rare cases that there is a conflict between voltage constraint (3.6) and the rest of the constraints, finding a feasible solution is impossible. This means that voltage requirements for some nodes might not be met. To resolve such conflicts, penalty values are added to the objective function (3.5) for the voltage constraints of those nodes that cannot be fully satisfied. This makes finding a feasible solution possible, while it ensures those voltage constraints are satisfied as much as possible. It is worth mentioning that we did not face a conflict between constraints in our simulations, so the rejection of

the infeasible solution was always enough to handle the constraints.

3.4 Simulation Results

The effectiveness of the proposed SVR-based MPC is evaluated by conducting several simulation studies. The test system chosen for this study is IEEE123 bus test feeder [45]-[46] as shown in Fig. 3.8, in which NC stands for Normally-Closed switches while NO denotes Normally-Open switches. The secondary tap ratios of four OLTCs in the system can vary between 0.9 and 1.1. There are four capacitor banks in the system. The one in bus 83 is a 3-phase bank with a total capacity of 900 kVAR while the other three capacitor banks are single-phase, each has 200 kVAR rating. The rating of each capacitor unit in these banks is considered 50 kVAR. The information about seven DERs of the system is presented in Table 3.1. These DERs are only turned on in the simulations that study the effects of DERs on performance of the proposed control method.

The power flow simulation is performed using OpenDSS connected to Visual Studio via COM interface. The control algorithm is developed using C++/CUDA programming languages where the computationally heavy parts of the code (as described in Section 3.3.4) run on GPU (CUDA) and the rest of the algorithm is executed by CPU (C++). SVM^{light} C library [47] is used for training the SVR model. The system used has an Intel CoreTM i7 3.5 GHz processor with 16 GB memory. The GPU used is GeForce GTX 1080 with 2560 CUDA cores and 8 GB memory. The first part of this section describes how customer load profiles are modelled and simulated. The second part of this section represents simulation results for the training stage of the SVR model, while the third part of the section provides the simulation results of the implementation of the proposed MPC approach using the trained SVR model. The last part of this section compares the computation time of the proposed GPU-based MPC to its sequential counterpart.

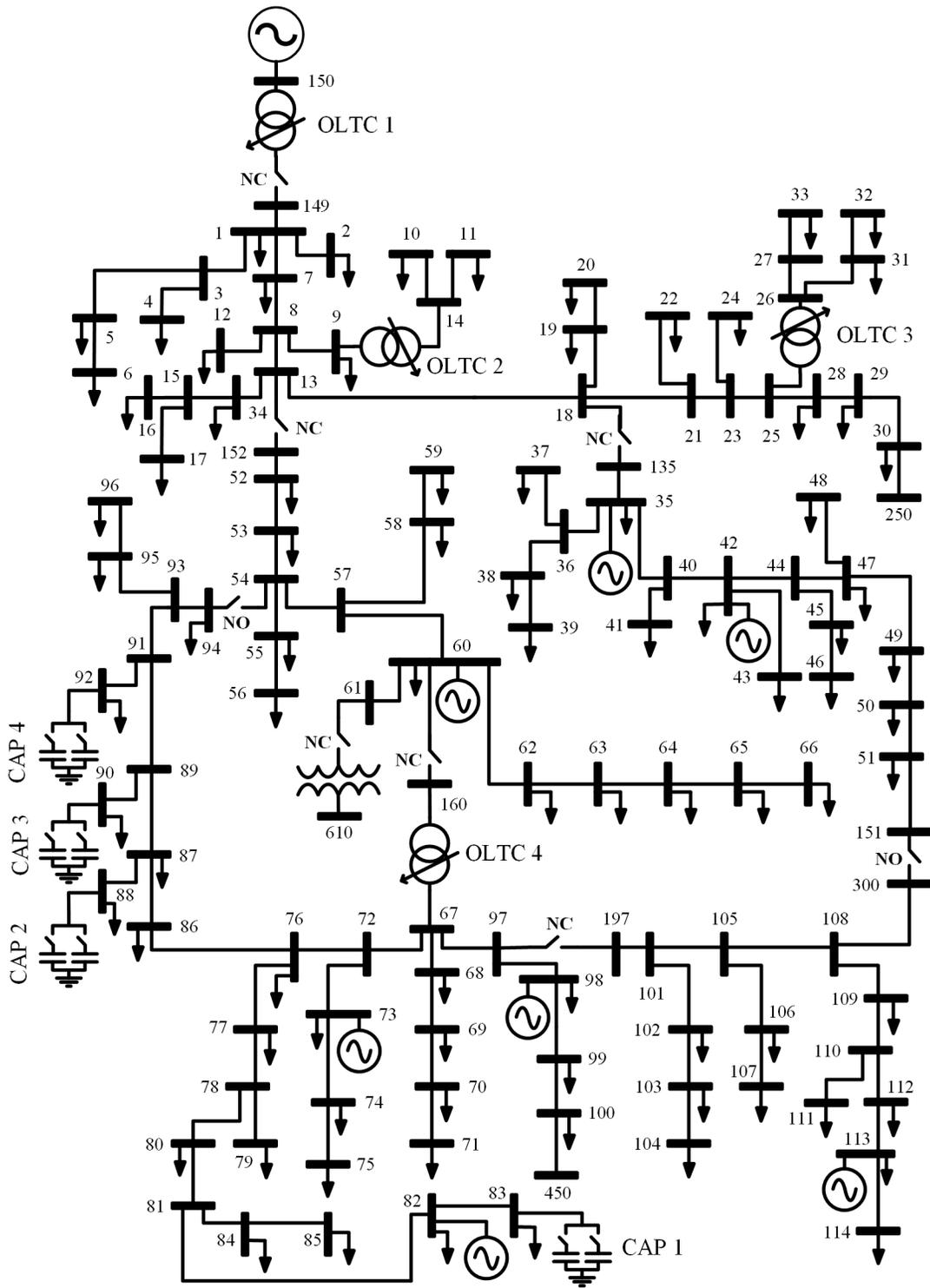


Figure 3.8: IEEE123 distribution test feeder

Phases:	Rating: (kVA)	Buses:
3-Phase	410	35, 42, 60, 98
1-Phase	125	73, 82
1-Phase	80	113

Table 3.1: The ratings and locations of the installed DERs

3.4.1 Load Profile Simulation

household load consumption changes drastically during a day. Power consumption is low at midnight and early morning, while it experiences two peaks before noon and in the evening. To simulate the time-varying load profile, the method proposed in [48] is used. [48] takes advantage of Monte Carlo simulation to generate realistic load profiles. Sequential Monte Carlo simulation methods are a set of simulation-based methods that provide a convenient approach for computing the posterior distribution of events [49]. Sequential Monte Carlo can be used to simulate the ON/OFF time of different household appliances with different probability distributions. We implemented a simplified version of [48] in which multi-stage appliances, the initial current of motors, and electric vehicles are not considered. The power consumption of home appliances is obtained from [50] and their probability distribution is adopted from [51].

IEEE123 is highly unbalanced as most of its loads are single-phase. It is assumed in this study that 70% of loads are residential with varying daily profiles, while the rest have constant load profiles. Each residential load bus in IEEE123 is modelled with a few household loads, where the number of households connected to a bus is decided based on the base kw of that bus. Fig. 3.9 shows 24-hour Monte Carlo simulation results for 6 household loads connected to a bus, and the total active load of that bus in p.u. The total active residential load profile for IEEE123 system for a 5-day simulation period with a resolution of 15 minutes is presented in Fig. 3.10.

3.4.2 Training the SVR Model

During the training stage, the SVR builder unit collects operational data of the system via AMI for a few days. Capacitor banks and OLTCs are supposed

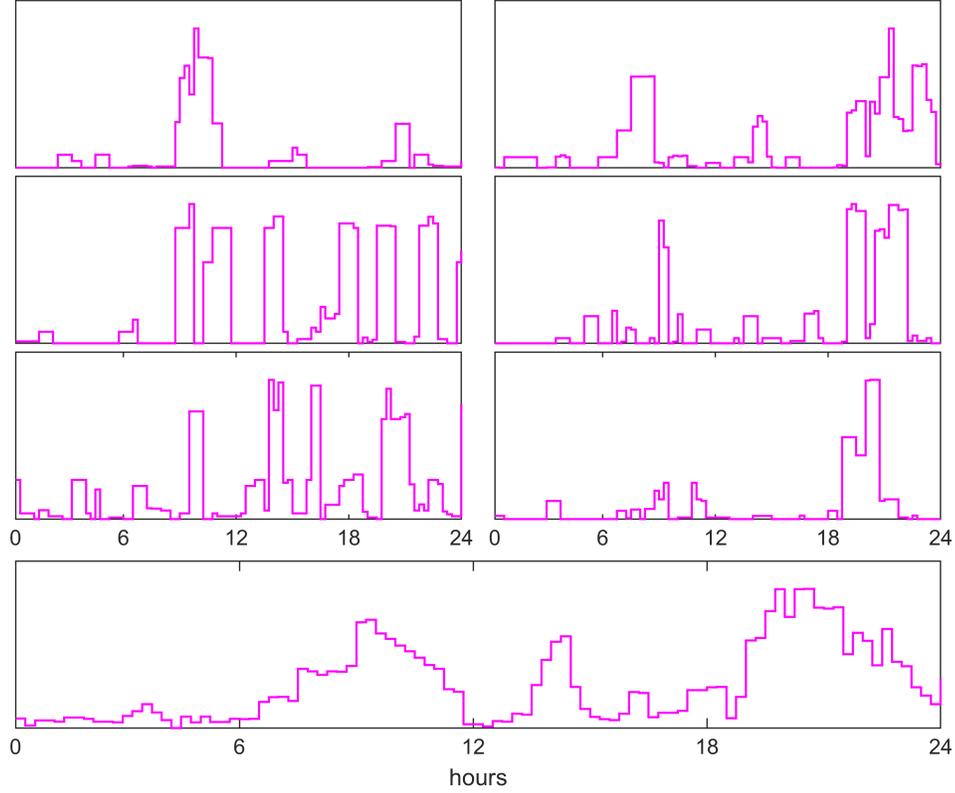


Figure 3.9: Daily active load profile of 6 residential loads and the total active load of the bus they are connected to

to be switched randomly during this period and residential load are simulated using Monte Carlo method. In order to evaluate the effect of the size of training data on the quality of the model, the SVR builder unit trained five models with different size of training data ranging from one day (96 samples) to five days (480 samples) using Gaussian kernel. The five generated models were evaluated by feeding a 24-hour load profile (different than the one used for training) to them and measuring the following indices:

$$E_{loss} = \sum_{j=1}^{N_S} \frac{|loss^m(j) - loss^p(j)|}{loss^m(j) \times N_S} \times 100 \quad (3.19)$$

$$E_{volt} = \sum_{j=1}^{N_S} \sum_{i=1}^{N_L} \frac{|V_i^m(j) - V_i^p(j)|}{N_L \times N_S} \times 100 \quad (3.20)$$

where E_{loss} is the percentage of average error in loss prediction during the simulation time and E_{volt} represents the average error of voltage prediction

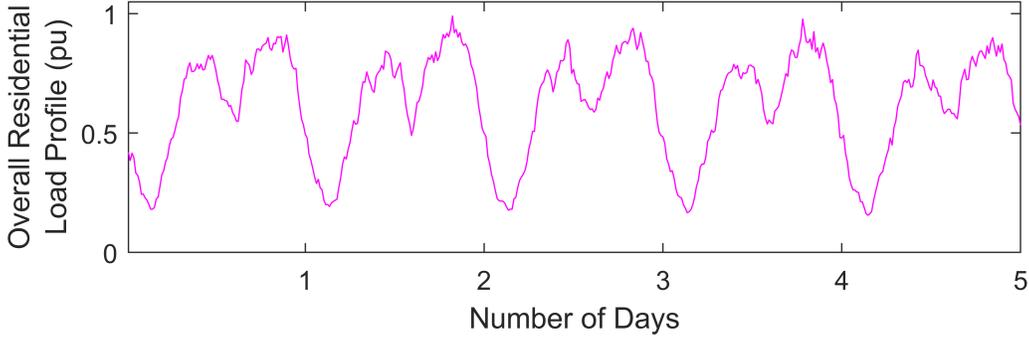


Figure 3.10: 5-day simulation of active residential load profile of IEEE123 test system

Models:	1Day	2Days	3Days	4Days	5Days
$E_{loss} : (\%)$	12.00	9.32	8.30	7.94	6.55
$E_{volt} : (\%)$	1.91	1.83	1.83	1.79	1.78

Table 3.2: Loss and Voltage Prediction Errors for 5 SVR Models

per bus in percentage. N_S is the number of simulation steps (96 steps during 24 hours), N_L is the number of load buses, $loss^m$ and $loss^p$ are measured and predicted losses, respectively. V_i^p denotes the predicted voltage of the i^{th} bus in p.u. while V_i^m is the measured voltage of that bus.

Table 3.2 shows loss and voltage prediction errors for the five trained models in which 1Day means the model trained with measurement data of one day, and so on. It can be observed that in both loss and voltage predictions the accuracy of models has direct relation with the size of training data. This is more prominent in loss prediction where E_{loss} for 1Day is almost twice the prediction error of 5Days. SVR models show higher accuracy when it comes to voltage prediction so that E_{volt} for all models is below 2%. Though 5Days model is the most accurate, $E_{volt} = 1.91\%$ for 1Day, which is trained using only 96 samples, is still impressive.

As the simulation results that are later provided in this section show, the error feedback feature of MPC can effectively compensate for the prediction errors of 5Days model. Therefore, we did not increase the size of training data to achieve a more accurate model. For higher accuracy, one can gather more

AMI measurements to train a more accurate SVR model. For example, using measurement data of 10 days (960 samples) results in an SVR model with $E_{volt} = 1.76\%$ and $E_{loss} = 4.46\%$. We also trained SVR models using polynomial kernel to evaluate the performance of that kernel. The results especially in voltage prediction were worse. For instance, E_{volt} for 5Days model using polynomial kernel was 1.95%, which was even worse than the performance of Gaussian kernel in training 1Day model. So, for the rest of the simulations only models based on Gaussian kernel were used.

3.4.3 MPC Control

In this section the effectiveness of the SVR-based MPC is validated under several operational scenarios, by comparing its performance to a model-based MPC. The following control approaches are considered:

- **No Cont.:** the system is not controlled
- **Cont. 1:** MPC with 5Days SVR model and parallel PSO as the optimizer with error compensation
- **Cont. 2:** same as Cont. 1, but without error feedback
- **Cont. 3:** MPC with 2Days SVR model and parallel PSO as the optimizer with error feedback
- **Cont. 4:** same as Cont. 3, but without error feedback
- **Cont. 5:** circuit-based MPC with sequential PSO as the optimizer with error compensation
- **Cont. 6:** same as Cont. 5, but without error feedback

No Cont. shows the behavior of the distribution system when there is no control strategy in place. Cont. 1 to Cont. 4 are four variants of the SVR-based MPC. The reason to study MPC with and without error feedback is to measure to what extent the error feedback can compensate the prediction errors of regression models or prediction errors caused by configuration changes

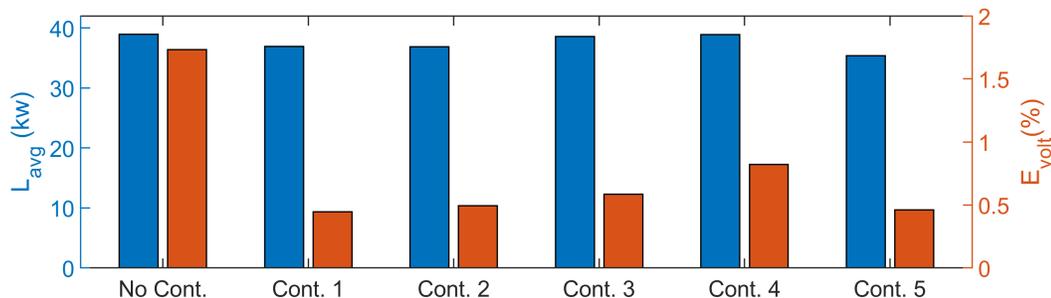


Figure 3.11: Performance of controllers under scenario 1 where $w_1 = 5000$, $w_2 = 1$

in the system. Cont. 1 is the main SVR-based MPC as it employs the accurate 5Days model and takes advantage of error feedback as well. Cont. 5 and Cont. 6 are MPCs that use circuit-based simulations (performed by OpenDSS) as their internal models. Cont. 5 is closed-loop whereas Cont. 6 does not receive error feedback.

If the circuit model is accurate, there will be no prediction error. Hence, there is no need for error feedback, and Cont. 5 and Cont. 6 can provide the optimal solution to the VVO problem (provided that their optimizers can converge to the global optimum). Therefore, their solutions can be used as a reference to evaluate the performance of SVR-based MPCs. In other words, the closer the solutions of Cont. 1 to Cont.4 to that of Cont. 5 (or Cont. 6), the better the SVR-based MPCs are performing. In case there are inaccuracies in the internal circuit and the SVR model, Cont. 5 delivers the best possible solution a closed-loop control method could offer. So again, it can be considered as the best possible reference to validate the effectiveness of an SVR-based MPC.

Several operational scenarios are considered to evaluate the proposed method. In all cases V_{sp} is considered 1 *p.u.* and the acceptable voltage tolerance is set to $\pm 0.05 p.u.$ E_{volt} in (3.20) is modified to evaluate average voltage deviation from the setpoint by replacing V_i^p with V_{sp} . The other considered criterion for evaluating the control methods is the average daily loss $L_{avg} = \sum_{j=1}^{N_S} loss(j)/N_S$. **Scenario 1:** IEEE123 system is tested under normal conditions for 24 hours.

For all MPCs, $T_c = T_P = 1$ step (15 minutes) and $\beta_1 = \beta_2 = 1$. There is no limit on the number of times capacitor banks and OLTCs can be switched, or how much they can change. This means in (3.5), w_3 and w_4 for all capacitors and tap changers are set to zero. w_1 for all load buses and w_2 are set to 5000 and 1, respectively. This puts more emphasis on minimizing voltage deviations than power loss as power utilities have obligations to keep voltage profiles within specified limits. The circuit model employed by model-based MPCs is accurate, so both Cont. 5 and Cont. 6 will generate the same results. Fig. 3.11 shows E_{volt} and L_{avg} for No Cont. and Cont. 1 to Cont. 5. SVR-based MPCs show great performance, with results close to the output of Cont. 5. The only exception is Cont. 4, with $L_{avg} = 38.89kw$ almost as high as No Cont. and the highest E_{volt} among controllers. Model inaccuracy, lack of error compensation and $w_1 \gg w_2$, all contribute to such poor performance. Cont. 1 delivers the best performance among SVR-based methods with the closest results to Cont. 5. This is not surprising as it takes advantage of the accurate 5Days model, along with error compensation. Tap positions of OLTCs and kVAR injections of capacitor banks for Cont. 1 are shown in Fig. 3.12 and Fig. 3.13, respectively.

Equation (3.5) is a multi-objective optimization problem and the solution to that problem depends on the selected values for w_1 and w_2 . A better insight to the performance of the proposed methods can be obtained when only one of the objectives of the VVO problem is considered. Fig. 3.14 provides simulation results for the case that only the minimization of power loss ($w_1 = 0$, $w_2 = 1$) is considered. Considering L_{avg} , it can be concluded that all MPCs can adequately minimize power loss. Also, by taking a look at the power loss of Cont. 1 during 24 hours, it is obvious that the performance of Cont. 1 is almost as good as Cont. 5. Both Cont. 1 and Cont. 5 can effectively lower down power loss compared to No Cont.

Fig. 3.15 represents simulation results for voltage deviation minimization ($w_1 = 1$ and $w_2 = 0$). Similar to power loss minimization, the SVR-based MPCs show satisfactory results, while Cont. 1 generates the closest results to Cont. 5. Also, comparing E_{volt} for Cont. 3 to that of Cont. 4 demonstrates

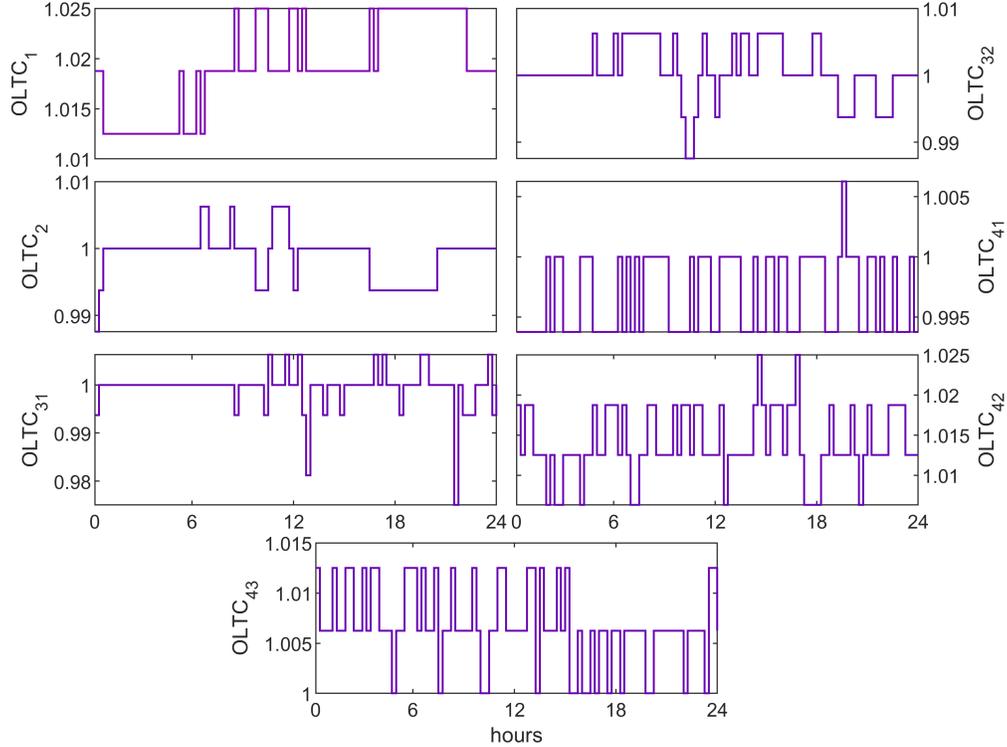


Figure 3.12: Tap positions of OLTCs for Cont. 1 under scenario 1

how effectively the prediction error feedback can improve the performance of data-driven MPCs. For instance, when only the minimization of voltage deviation is considered, $E_{volt} = 0.4579\%$ for Cont. 3. This means 44% improvement of performance compared to $E_{volt} = 0.8175\%$ for Cont. 4.

Scenario 2: Same as Scenario 1, but 1% error with Gaussian distribution is added to load and voltage readings of smart meters to model the inaccuracy of the meters. This scenario can be considered as sensitivity study of the proposed method, as it examines the robustness of the SVR-based MPC in the presence of noise. Fig. 3.16 shows simulation results for this scenario. The results are almost the same as scenario 1, although the measurement errors cause slight decline in performance for all controllers. For example, in case of $w_1 = 0$ and $w_2 = 1$, L_{avg} for controllers goes up by 1.36% on average. In fact, measurement errors affect the SVR-based MPC more than its model-based counterpart. This is the result of the presence of such errors in both training and operational stages of SVR-based MPCs, whereas measurement errors only affect Cont. 5

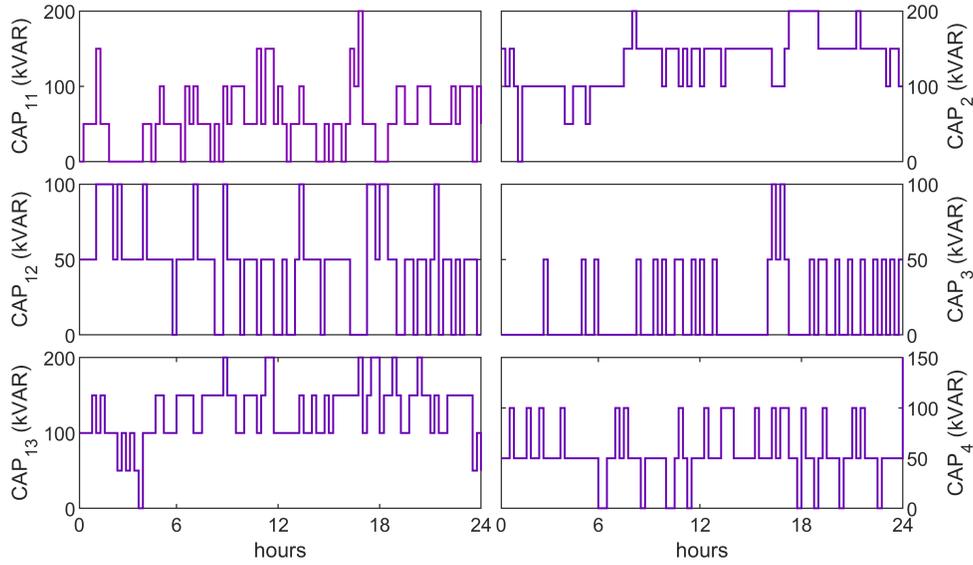


Figure 3.13: KVAR injections of capacitors for Cont. 1 under scenario 1

Controllers:	No Cont.	Cont. 1	Cont. 2	Cont. 5
$L_{\text{avg}} : (kw)$	39.82	36.69	36.88	35.22
$E_{\text{volt}} : (\%)$	1.733	0.492	0.531	0.485

Table 3.3: Simulation results of scenario 3

during its operational stage. Though, despite effects of measurement errors, Cont. 1 to Cont. 4 can still deliver satisfactory performance.

Scenario 3: Similar to scenario 1 ($w_1 = 5000, w_2 = 1$), but the values of capacitor banks and taps of OLTCs can change up to once per hour. This enforces wear and tear constraints the same as 3.11-3.12 to the VVO objective function. The simulation results for this scenario is presented in Table 3.3. Tap ratios of OLTCs and injections of capacitors are shown in Fig. 3.17 and Fig. 3.18, respectively. It can be observed that no OLTC or capacitor is switched in a period shorter than one hour (4 control steps). It proves that the optimizer is successfully enforcing constraints 3.11-3.12.

Scenario 4: Similar to scenario 3, but both prediction and control horizons of all MPCs are extended to 1 hour ($T_p = 4$ and $T_c = 4$). It is also assumed that the short-term load forecast for the one hour ahead is provided for MPCs. Extended prediction and control horizons allow MPCs (both SVR-based and

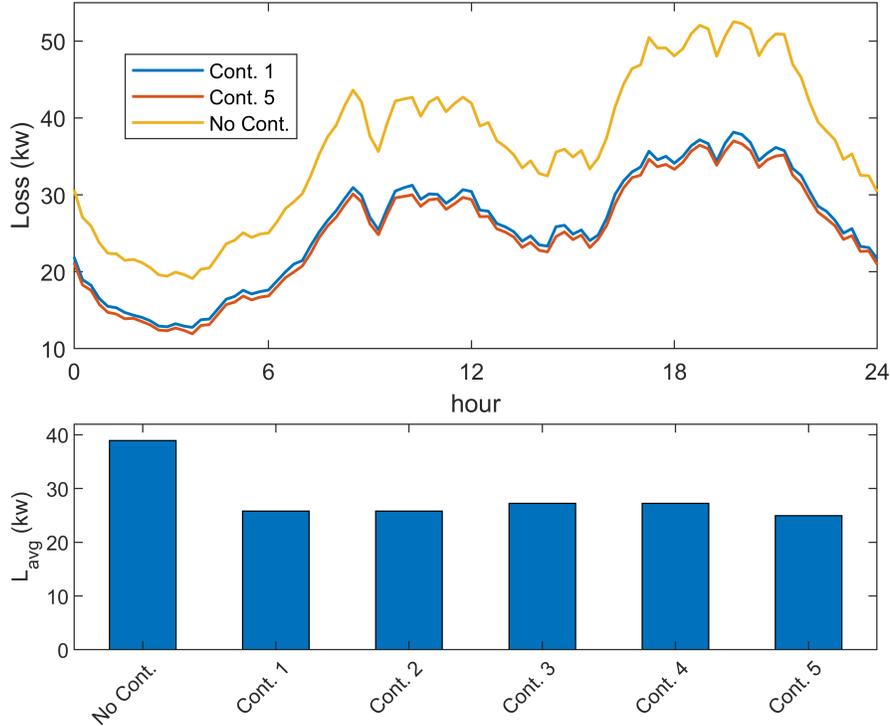


Figure 3.14: Performance of controllers under scenario one where $w_1 = 0$ and $w_2 = 1$ (minimizing power loss)

model-based) to better manage the switching constraints. Such improvement in performance can be observed in Fig. 3.19 where the simulation results of scenarios 3 and 4 are compared. Though only a slight improvement in loss reduction can be seen in Fig. 3.19, the decrease of E_{volt} for all controllers is significant. Therefore, working with an extended prediction horizon is recommended in case short-term load forecasts are available.

Scenario 5: In this scenario, all DERs shown in Fig. 3.8 are plugged in. 5Days SVR model is retrained to include DERs, and the circuit model of Cont. 5 considers those DERs as well. $w_1 = 7000$, $w_2 = 1$, $T_c = 1$, and $T_p = 1$. Table 3.4 represents the simulation results for this scenario. The results show that the power injection of DERs has reduced power loss and voltage deviations in the system, compared to scenario 1. Cont. 1 has predicted the effects of DERs successfully, and using error compensation, it has been able to reduce L_{avg} and E_{volt} to values comparable to Cont. 5.

Scenario 6: Similar to scenario 5, however the SVR and circuit models are

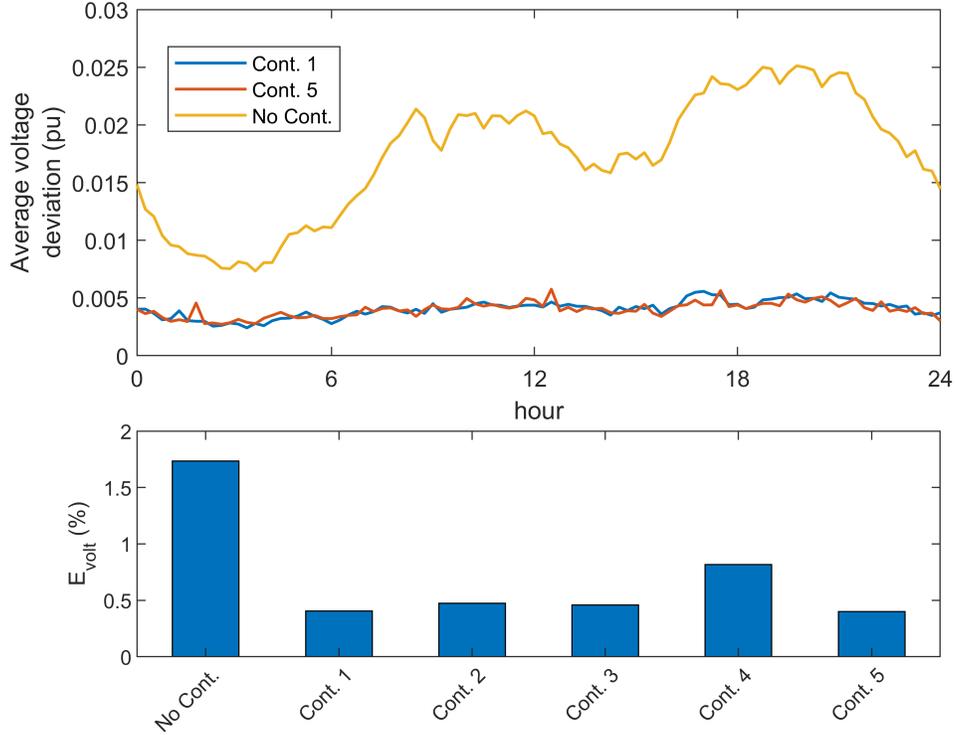


Figure 3.15: Performance of controllers under scenario one where $w_1 = 0$ and $w_2 = 1$ (minimizing voltage deviation)

Controllers:	No Cont.	Cont. 1	Cont. 2	Cont. 5
$L_{\text{avg}} : (kw)$	10.152	7.971	7.996	7.334
$E_{\text{volt}} : (\%)$	0.6703	0.308	0.364	0.268

Table 3.4: Simulation results of scenario 5: control under the presence of DERs

not updated, so they do not include DERs. The reason to study this case is to figure out how a wrong regression model will affect the performance of the proposed method, and to what extent the error compensation can mitigate adverse effects of a wrong model. Fig. 3.20 provides a comparison between scenario 6 and scenario 5. It is obvious by looking at Fig. 3.20 that employing a wrong SVR model for Cont. 2 and a wrong circuit model for Cont. 6 while no error feedback is available, has dramatically exacerbated the performance of those two controllers. This performance deterioration is more evident in minimizing voltage deviation where E_{volt} for both controllers jumps to more than 2%, far worse than $E_{\text{volt}} = 0.67\%$ for the system under no control. On the other hand, Cont. 1 and Cont. 5 that take advantage of error compensation

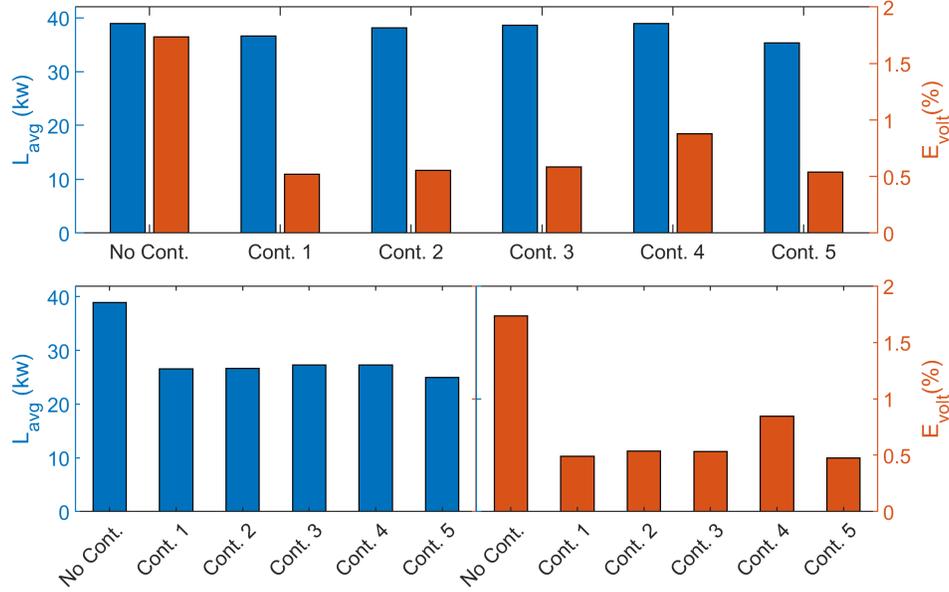


Figure 3.16: Simulation results for scenario 2. (Top): $w_1 = 5000, w_2 = 1$, (Bottom-Left): $w_1 = 0, w_2 = 1$, (Bottom-Right): $w_1 = 1, w_2 = 0$.

have been able to maintain their performance close the levels under scenario 5. The error feedback helps both MPCs to effectively compensate prediction errors even when an entirely wrong model is employed.

A closer insight on how the error feedback can improve the performance of the controllers can be obtained by observing the voltage profiles of the system for Cont. 1 and Cont. 2 under scenario 6. As can be seen in Fig. 3.21, voltage profiles of most of the load buses are above $1p.u.$ for Cont. 2. This is due to the fact that the SVR model of Cont. 2 does not include the active and reactive power injections of DERs. Therefore, Cont. 2 brings more capacitors online and increases secondary taps in order to level up voltage profiles. Cont. 1 does not do any better at the first control interval. This is because there is no error feedback at the beginning of the simulation. However, at the second control step when the prediction error is fed to the controller, the performance of Cont. 1 improves noticeably and the controller is able to maintain its good performance until the end of the simulation. This proves how useful the closed-loop feature of MPC is in controlling distribution systems. Distribution grids are subject to many topology changes such as adding DERs, switch reconfiguration, *etc.*, and the closed-loop feature can, to some extent,

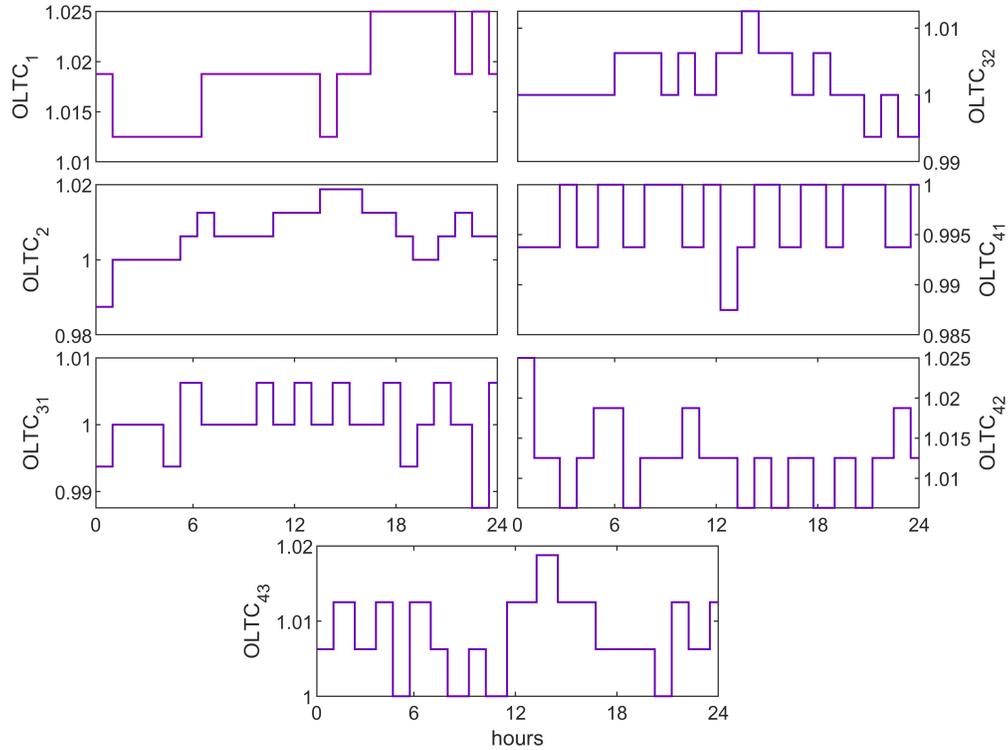


Figure 3.17: Tap positions of OLTCs for Cont. 1 under scenario 3 (constrained switching)

compensate for prediction errors in case the controller is not aware of the occurrence of such changes.

Scenario 7: In this scenario $T_p = T_c = 1$ step, $w_1 = 5000$ for all nodes, and $w_2 = w_3 = w_4 = 0$. An SVR model for IEEE123 test system is trained while DERs are turned off. However, in the beginning of the simulation, the test system is reconfigured so that the switch between nodes 18 and 135 in Fig. 3.8 is opened, whereas the switch between nodes 151 and 300 is closed. As a result, the trained SVR model is no longer valid. Fig. 3.22 compares the performance of Cont. 1 and Cont. 2 with an SVR model trained before reconfiguration (wrong SVR model) to the performance of Cont. 1 with a correct model trained after reconfiguration. The figure also provides simulation results for Cont. 5 and Cont. 6 with circuit models before (wrong circuit model) and after (correct circuit model) reconfiguration. E_{volt} for Cont. 2 and Cont. 6 that are using wrong models is substantially higher than E_{volt} for Cont. 1 and Cont. 5 with correct models. This comes as no surprise as

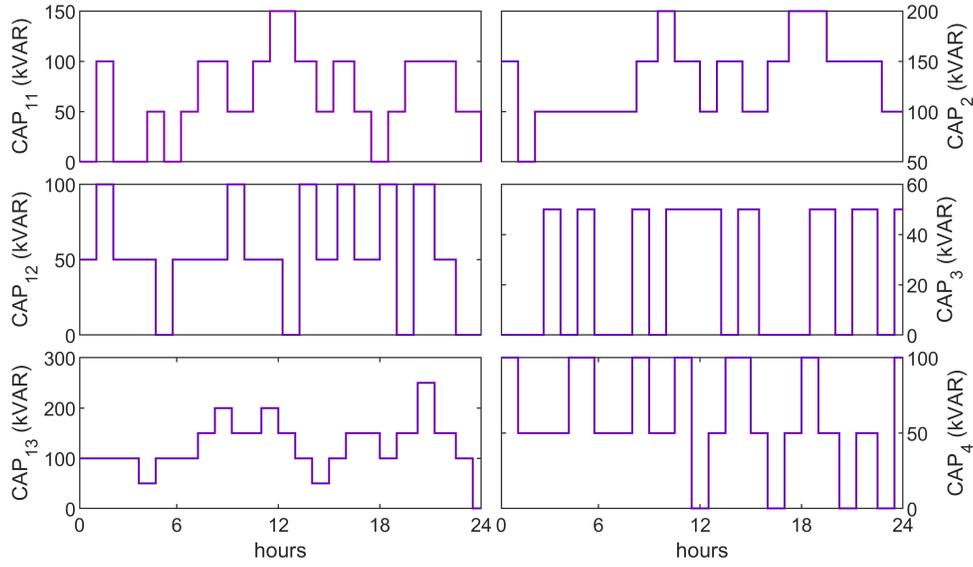


Figure 3.18: KVAR injections of capacitors for Cont. 1 under scenario 3 (constrained switching)

their models (SVR or circuit) are wrong and no error feedback is available. Feedback compensation, however, helps Cont. 1 and Cont. 5 with wrong models to maintain satisfactory performance. It can be observed in Fig. 3.22 that E_{volt} for Cont. 1 and Cont. 5 with wrong models is considerably lower than that of Cont. 2 and Cont. 6. The advantage of error compensation studied in scenario 6 and scenario 7 proves that a closed-loop control approach like our proposed method can outperform the open-loop algorithm in [33] in the presence of prediction errors caused by changes in the distribution system.

Scenario 8: Similar to scenario 1, but all switches shown in Fig. 3.8 are closed and the SVR model is retrained. This turns IEEE123 to a meshed system that many VVO methods such as the MPC methods in [31]-[32] or the data-driven algorithm in [33] that are designed for radial systems cannot control. Fig. 3.23 shows the power loss and average voltage deviation per bus for Cont. 1 and Cont. 5 under this scenario. The results show that while Cont. 1 delivers fair performance in minimizing power loss compared to Cont. 5, its ability to reduce voltage deviation is as good as its model-based counterpart.

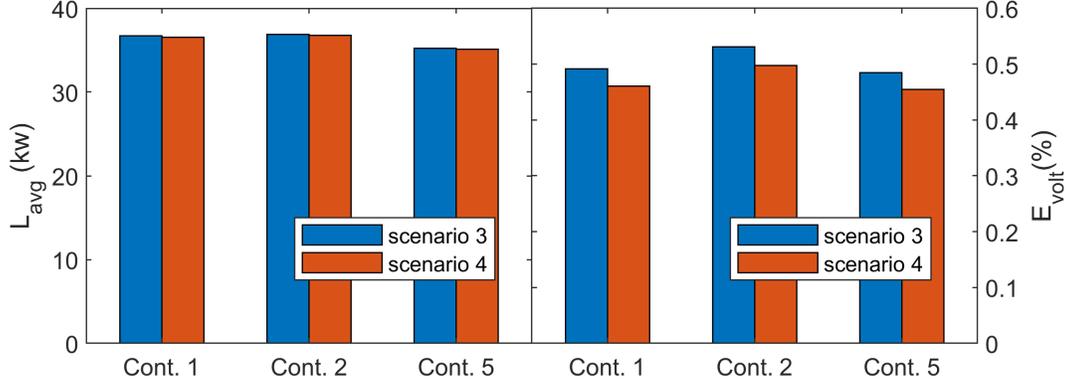


Figure 3.19: Comparing the results of scenario 3 and scenario 4

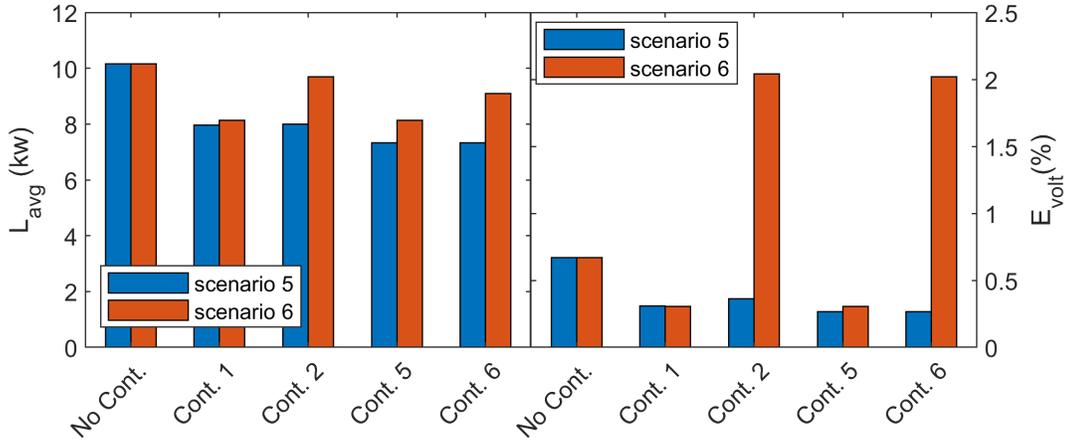


Figure 3.20: Comparing the results of scenario 5 (correct models) and scenario 6 (wrong models)

3.4.4 GPU-vs-CPU Speedup

In order to compare the performance of the GPU-based MPC to the sequential MPC, these two methods were executed under scenario 1 for different number of particles of PSO. The termination criterion for PSO was set to reaching 50 iterations. The average time each method spends on generating control decisions is presented in Table 3.5. The results are averaged over 12 runs for each method/Num. of particle. It is obvious that the GPU-based MPC

Num. of particle:	10	25	50	100	250	500
GPU comp. time (s):	0.446	2.045	4.067	8.195	20.608	40.993
CPU comp. time (s):	23.303	58.833	116.466	233.224	583.180	1190.330

Table 3.5: Computation time of GPU-based MPC and sequential MPC

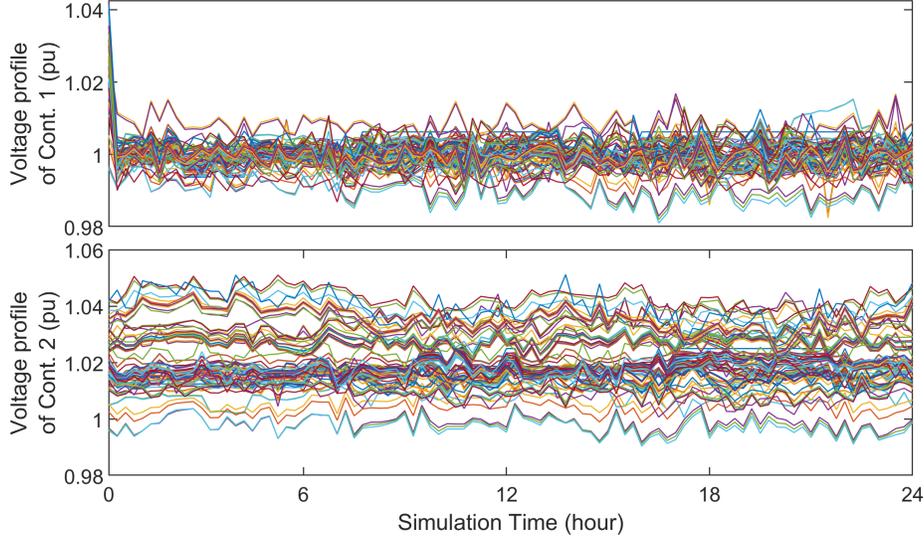


Figure 3.21: Voltage profiles of load buses of IEEE123 controlled by Cont. 1 and Cont. 2 under scenario 6

is substantially faster than its sequential counterpart. For any number of particles up to 500, the parallel MPC can generate decision controls in seconds, while the sequential MPC requires over 19 minutes (longer than the 15-minute control step) to generate a solution using 500 particles. Fig. 3.24 show the GPU-vs-CPU speedup for these two methods. While the speedup is over 52 times for 10 particles, it reaches a steady rate about 29 times for the number of particles 25 and over.

Fig. 3.25 shows the runtime chart of the vector-matrix multiplication kernel (3.17) on GPU 32 streams are pre-defined for CUDA compiler. It can be observed that the concurrency of streams is achieved, so that up to 20 kernels are being executed at each given time. The reason not all 32 streams are running concurrently is that GTX 1080 that is executing the code has 20 SMXs. Hence, it can launch and execute up to 20 streams at once ¹.

¹Technically, an SMX can launch and execute up to two kernels concurrently. However, the SMX can only do so if there are enough resources including shared memory and registers (registers are used for saving constant variables, variables defined inside of the kernel, and intermediated calculation results) available for each thread of the kernel.

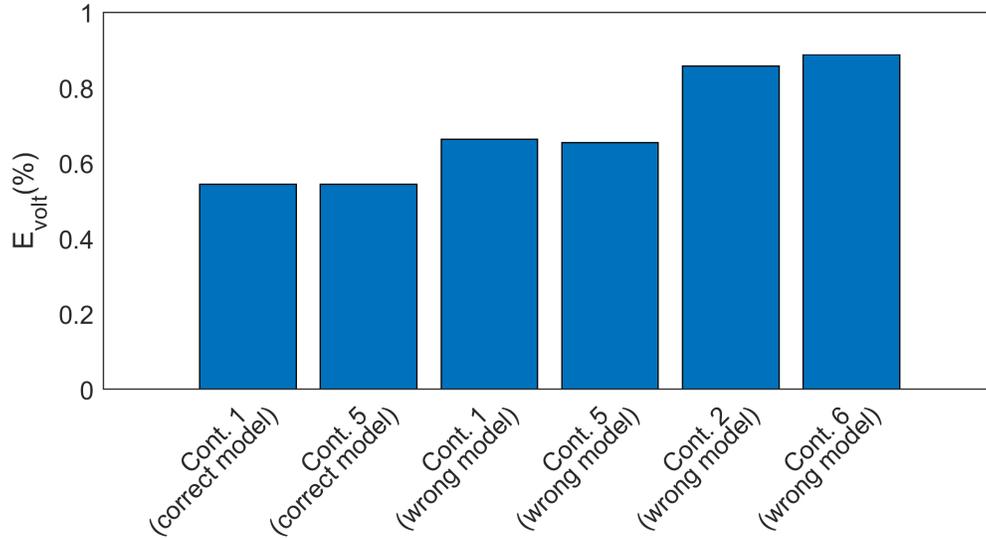


Figure 3.22: E_{volt} for control approaches under scenario 7

3.5 Summary

A new data-driven VVO scheme has been presented. The method uses AMI measurements to train an SVR model for the distribution grid. The trained model is employed in a closed-loop predictive controller to perform Volt-Var Optimization. The performance of the proposed controller was assessed by applying it to IEEE123 bus test system. Simulation results showed that the proposed approach is able to deliver close to optimal results, even in the presence of DERs, noise, and changes in the configuration of the system. The following observations are revealed from simulation results:

- Support Vector Regression can adequately model distribution systems even with a limited number of training samples. It also does not require capabilities such as phasor or high-resolution measurements that the existing smart meter and AMI technologies are not equipped with.
- The closed-loop nature of MPC helps the proposed algorithm to satisfactorily control the system even if topology changes occur.
- The controller can perform well under the presence of measurement noise.
- The GPU implementation of the algorithm substantially reduces the

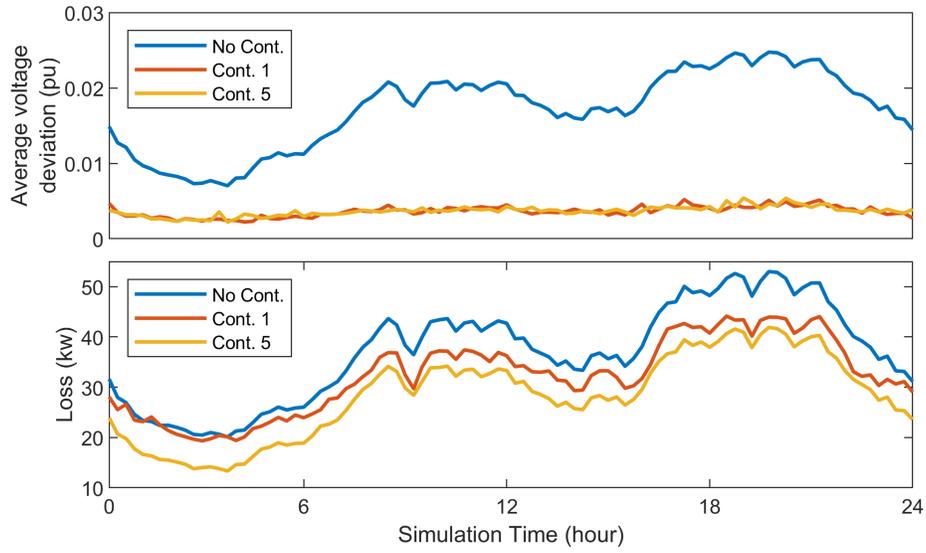


Figure 3.23: Power loss and average voltage deviation of Cont. 1 and Cont. 5 under scenario 8

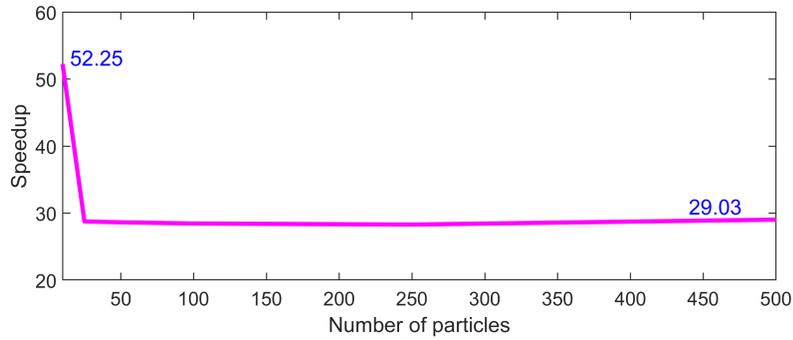


Figure 3.24: The speedup achieved by parallel MPC for different number of particles

computation time and make the algorithm suitable for real-time applications.

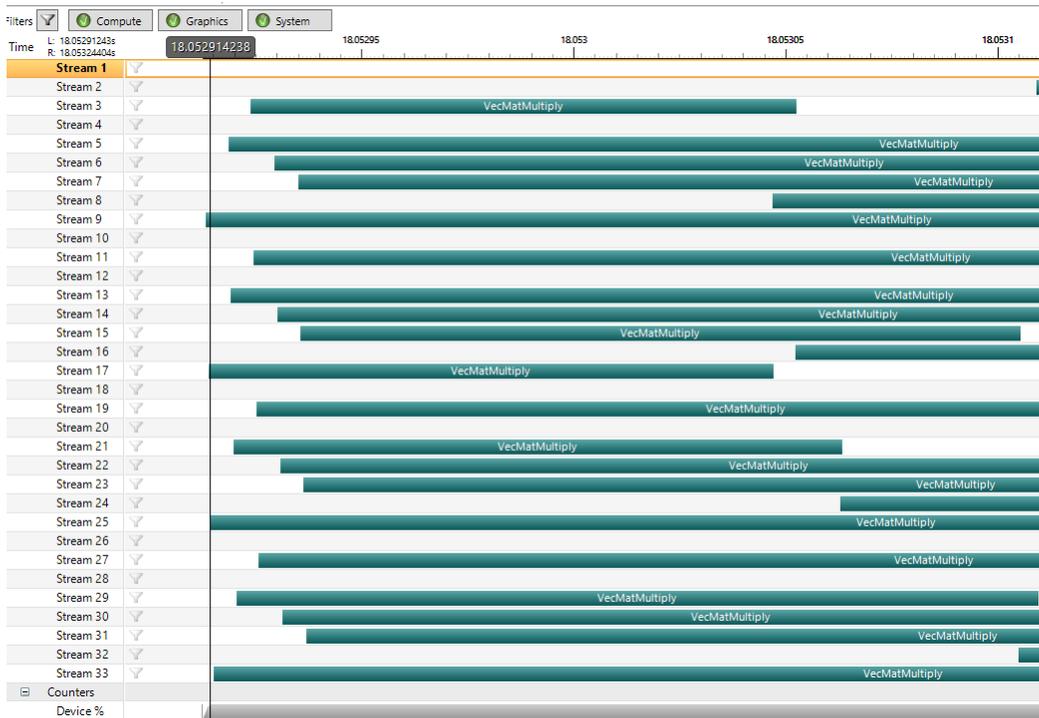


Figure 3.25: Concurrent launching of several vector-matrix multiplication kernels on GPU

Chapter 4

Optimal Placement of Smart Meters for Locating Faults

In this chapter, a novel method is proposed for the optimal placement of smart meters for the purpose of locating faults. The optimal placement is treated as an optimization problem. The distribution grid is divided into segments. Each segment represents the smallest piece of a grid where faults can occur, and locating faults means identification of the segment with a fault.

During the optimization procedure, segments which cannot be correctly identified as locations of faults are merged. They are called zones. The goal is to find such a placement of smart meters so that the number of zones is maximized. In the best case scenario the number of zones is equal to the number of segments, while in the worst case scenario the whole grid is a single zone.

At each iteration of the optimization procedure a set of locations for smart meters is chosen and a classification model is trained to identify segments with faults. A merging routine is applied to merge segments with misclassified faults into larger zones. This process continues until there are no more segments/zones with misclassified faults.

The optimization procedure is performed in two stages. At the first stage, a random search is performed with the Naive Bayesian classifier used for classification. The objective of this stage is to find a good initial solution for the second stage. During the second stage, a Simulated Annealing based search process is utilized to find optimal locations for smart meters. The SVM classi-

fication is used here. A single SVM model is developed for locating faults for each type of fault. Also a fault-type classifier is constructed using the SVM classification. The results of this optimization procedure are the followings: 1) the best placement for smart meters; 2) a zoning map (partitioning of a grid ensuring a full fault location ability); 3) classification models for locating faults; and 4) a model for detecting types of fault.

Simulation results of applying the proposed method on IEEE34 test system and its extended version shows the excellent performance of the proposed scheme in placing meters and locating faults.

4.1 Literature Review

Upon occurrence of a fault in a distribution grid, the protection devices act to isolate the fault from the rest of the system. If a fault is temporary, a recloser tries to clear it out. If it is permanent, other protection devices such as sectionalizers and fuses trip in order to confine that fault. In case sectionalizers and fuses do not act, or if the fault happens on the main feeder on their upstream, the overcurrent relay will sense the fault and send a trip command to the circuit breaker in the substation to de-energize the feeder until the fault is cleared out.

The traditional way of finding the outage area in distribution systems, which is still practiced in many utilities, is to map the outage area based on customer phone calls. Historical outage data and weather data can help to narrow down the faulted area as well. Crews are then sent to the specified area to locate the fault and clear it. Tripped protection devices are reset, and the system is re-energized after the fault is cleared out.

This is in contrast to transmission systems in which well-defined and robust methods for locating faults are available. Transmission systems are equipped with several types of protective relaying including overcurrent, frequency, distance and differential, as well as advanced measuring devices such as PMUs. This allows for developing precise fault locating methods. For example, the distance relay can precisely locate a fault by dividing the voltage phasor by

current phasor and comparing the result to the impedance of the line.

Developing an analytical fault locating method for distribution grids is generally more difficult than transmission systems. This is due to the fact that distribution systems are not well-monitored the way transmission systems are, they are highly branched with many lateral cables connected to the feeder, loads are unbalanced, and different grounding techniques are implemented. However, there are many analytical methods proposed to improve fault location estimation across distribution grids. These methods can be divided into two groups; methods that locate the outage area and approaches that find the location of fault.

A method based on historical data proposed in [52] to locate the outage area. The historical data used in that study includes historical weather conditions, cause of fault, situation of fault, and the faulty equipment. A data mining method called rough set theory is employed to derive patterns and tools for estimating the outage area. In [53] historical data including weather conditions, natural degradation of apparatus, traffic accidents and overload is used to locate the outage area and the faulty equipment. A probabilistic Bayesian network is implemented to predict the outage area Based on historical data.

Fault indicators are devices installed across transmission and distribution systems to locate the outage area. Fault indicators can have different characteristics such as overhead/underground, manual/automatic reset, visual/remote indication, *etc.* [54]. Installing fault indicators is not enough to locate fault area and an algorithm is required to analyze the data from fault indicators and locate the faulty segment. A heuristic method using fault indicators' data is proposed in [55] based on fault current detected by fault indicators and the observation that line sections between fault indicators can be considered as potential fault locations. Another method based on fault indicators is studied in [56] that uses depth-first search to divide and search the potential faulted area identified by fault indicators, until the faulty line segment is found.

Since it is not economical to install fault indicators at every corner of the distribution system, studies such as [57]-[58] have tried to find the opti-

mal placement for a limited number of fault indicators across the distribution system. Both mentioned studies have employed metaheuristic optimization algorithms for optimal placement. While [57] uses a variant of Genetic Algorithm called Immune Algorithm for optimization, the optimization algorithm used in [58] is PSO.

AMI data can also be used for locating outage area. Some smart meters are equipped with ‘last grasp’ capability. Last grasp is a signal sent by a smart meter to distribution management system following occurrence of a fault, right before that smart meter being de-energized. The aim is to let the power utility know that the customer has lost power. Last grasp signal or the fact that some smart meters are offline and no longer responding can be utilized to estimate the outage area. Locating the outage area using AMI data has been studied in [59]–[61].

The goal of fault location methods is to estimate the location of faults more precisely than outage area approaches. Impedance-based algorithms are a common approach for locating fault in distribution systems. These algorithms follow the same concept that a distance relay does to locate fault in transmission systems. They use voltage and current phasors of the feeder, measured during fault occurrence to estimate the impedance between the substation and the location of fault. Then, they compare the result to the circuit model of the grid to predict potential fault locations. The circuit analysis can be done based on symmetrical component analysis and Clarke transformation. The analysis, however, is more complicated than the fault study in transmission systems, as there are many distributed loads and several laterals across the distribution system. The equations have to be solved sequentially from the feeder towards each downstream load. The algorithms proposed in [62]–[64] take advantage of symmetrical component analysis to locate fault in distribution grids.

If a system is unbalanced, it is not possible to decouple it into positive, negative, and zero sequences. Thus, symmetrical component analysis cannot be applied to the circuit, and power flow equations under fault conditions have to be solved in phasor domain. Impedance-based methods proposed in [65],

[66] use phasor calculations to locate faults.

Impedance-based algorithms are simple to implement and only require measurements at the feeder. The major drawback of these methods is that they are unable to provide a unique location for a fault. A typical distribution system is comprised of several branches and laterals. There might be multiple locations in a distribution system that have similar calculated impedance seen from the feeder. Hence, these algorithms suggest a set of possible fault locations rather than a unique coordinate.

Voltage sag based methods are a group of fault locating techniques that work based on the observation that each fault causes voltage sags with different amplitudes across the system. These methods are measurement based, therefore they rely on devices that can record voltage sags during faults. Measuring devices are either smart meters that have voltage sag recording capability or digital fault recorders. Digital fault recorders are power quality devices that are installed in a few spots across transmission and distribution systems and graphically record voltage and current waveforms during fault conditions with a high sampling rate [67].

The concept of locating faults based on voltage sag measurements was first introduced in [68] and then implemented in [69]. To pinpoint a fault, these algorithms assume a fault at each node of the distribution system and run powerflow calculations similar to impedance-based methods to calculate voltage sag at each node. After assuming and calculating a fault for each node, they compare the calculated voltage sag patterns to the measured voltage sags from digital fault recorders or smart meters. The node whose voltage sag calculations is the closest match to the recorded voltage sag, is considered the closest node to the location of the fault.

The proposed method in fault [70] has improved the approach in [69] by adding another stage to the procedure that employs a decision tree to approximately locate the zone in which the fault happened. Hence, the fault locating approach is faster than [69]. The approach in [71] uses the outage area map besides voltage sag data from smart meters to further narrow down fault location. Voltage sag based algorithms use the same type of calculation as the

impedance-based methods do. Therefore, they can be considered as a category of impedance-based methods. However, they take advantage of measurements across the system, not just the main feeder. Thus, they are more accurate than impedance-based methods and can predict a unique coordinate as the location of fault.

Another group of fault location methods work based on machine learning. In these methods, a set of training data is used to train a machine learning model. The trained model will be able to classify fault types and pinpoint fault location. The output of the trained model could be the distance between the feeder and fault location, the impedance of fault, or the zone in which fault happens. In [72], the SVM is employed to classify and locate fault. It uses power flow simulations to generate a training set. Once the model is trained, smart meter voltage readings can be used to predict the type of fault and the zone in which fault occurs. Another approach based on the SVM is studied in [73] that can predict single line to ground faults in loop distribution systems. The mentioned method uses powerflow data to locate the zone of fault. Then it analyzes the fault transient waveforms using wavelet transform to exactly pinpoint fault location.

The method presented in [74] takes advantage of both the SVM and neural networks to locate fault. The SVM is used to classify fault types, whereas neural network predicts the impedance of fault. The method presented in [75] employs a combination of neural network, fuzzy logic and wavelet transform to locate fault in distribution grids. The output of the model is the distance between the feeder and the fault location. The same combination of learning methods has been employed in [76] considering current patterns as the input. The output of the trained model is the zone that contains the fault location. The advantage of machine learning based fault locating methods is that their online calculations can be done much faster than impedance or voltage sag based methods. However, the machine learning based algorithms that analyze voltage and current transients do not seem as practical as methods that use smart meter measurements. Capturing transient waveforms requires measurement devices with very high sampling rates, which might not be an economical

option for many distribution systems.

Many voltage sag and machine learning based approaches for locating fault rely on smart meters with voltage sag recording capabilities to operate. Installing such smart meters for all customers is not practical from an economical point of view. When the number of available smart meters is limited, it is important to where to install them to optimally record voltage sag patterns. This leads to a better approximation of fault location. In addition, in many fault locating algorithms such as [70], [72] and [76] the output of the method is a zone that contains the location of fault. These zones are pre-defined by the user and comprised of a few adjacent nodes. The number and size of these zones affect the overall accuracy of the algorithm. Hence, pre-defining these zones is not the best approach to gain the maximum fault locating accuracy.

The aim of this study is to address both aforementioned problems. The algorithm proposed in this study is able to find the best nodes for installing smart meters. Additionally, it can define zones across the distribution grid so that fault locating can be achieved with the maximum accuracy. To the best of our knowledge, such a study has not been conducted before. The proposed method divides the distribution system (buses and lines) into several small-size segments. It finds the optimal place for smart meters' installation via an optimization procedure. During the optimization procedure, for each combination of smart meters' locations, a model is trained using the SVM or NB, aiming to classify as many segments as possible correctly. If the machine learning model fails to precisely classify two segments, those segments are merged and form a new zone through a special merging procedure. Therefore, the higher the number of zones, the better the placement of smart meters is. So, the goal of the proposed algorithm is to place smart meters so that the machine learning model can precisely detect as many zones as possible.

4.2 Meter Placement Process: Overview

There are different types of fault in a distribution system. They can be categorized based on the number of phases affected and if those phases are short-

Single phase to ground (LG):	Phase A to ground (LGa) Phase B to ground (LGb) Phase C to ground (LGc)
Phase to phase (LL):	Phase A to phase B (LLab) Phase B to phase C (LLbc) Phase C to phase A (LLca)
Phase to phase to ground (LLG):	Phase A to phase B to ground (LLGab) Phase B to phase C to ground (LLGbc) Phase C to phase A to ground (LLGca)
Three phase (LLL)	
Three phase to ground (LLLG)	

Table 4.1: Types of fault in distribution systems

circuited to the ground. Table 4.1 lists all 11 possible types of fault that can occur in a grid. A single phase to ground (LG) fault is the most common, while a three phase to ground (LLLG) fault is the most dangerous one. A fault location algorithm must be able to detect not only the location but also the type of occurred fault. The goal of the proposed method is to find the best placement of smart meters so the measurements obtained from them allow for the estimation of fault locations with the maximum accuracy.

The first step of the proposed process is to divide the distribution system into several small segments. Fig. 4.1 shows how a piece of a distribution system which is a line between two buses can be divided into segments. A segment represents the smallest part of a grid that the proposed method can identify as the location of fault. The size of segments should be considered reasonably small to provide acceptable approximation of the location of fault. The maximum length of segments in this study is chosen to be 1000 ft (\approx 300 m). Eventually, the proposed method generates a zoning map for each type of fault, in which each zone is composed of segments – from one to several. During the execution of the method, segments associated with misclassified faults are merged into a single zone. However, if the proposed method can perfectly detect a segment as the location of a fault, then that segment becomes a zone. In other words, the proposed approach results in a zoning map built in such a way that faults can be located to the level of a single zone. A larger number of zones means better fault locating capability.

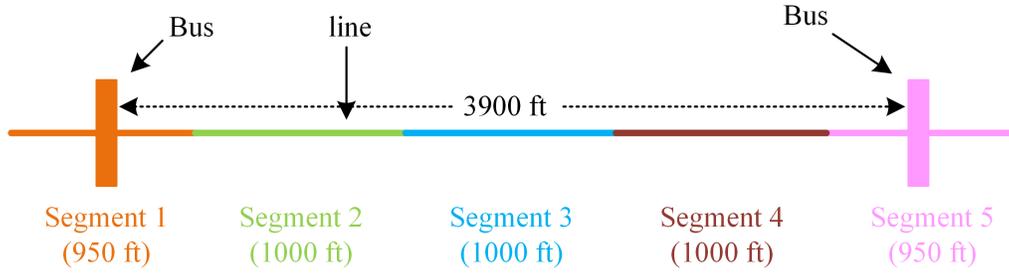


Figure 4.1: Dividing a piece of a distribution system into several segments not bigger than 1000 feet

The optimal placement of smart meters is obtained via solving an optimization task. At each iteration of the optimization process, the location of a pre-defined number of smart meters is updated, and a machine learning method is applied to construct classifiers able to identify which segment/zone is the location of fault.

Once the location of smart meters is updated, training datasets are constructed using voltage amplitude measurements (in p.u.) of the new locations of smart meters. A single data set is generated for each type of fault based on power flow study under fault conditions. Fault samples of a given type are simulated in random places across the grid including lines and buses. The voltage values measured by smart meters constitute the inputs to a classifier being built, while the output is the id of the segment where a fault occurs. A classification model for a given type of fault is trained using such a dataset.

Ideally, the trained classification model should be able to accurately classify all segments with no misclassifications. This means that the trained model can precisely recognize a fault location, i.e., identify a segment that contains the fault. However, due to the limited number of measurement points (smart meters), the accurate classification of segments is not possible. Therefore, after the model is trained, a confusion matrix is analyzed in order to find ‘misclassified segments’. If two or more segments are misclassified, these segments are merged together and build a new zone. The merging process of segments and constructing new zones continue, until there is no misclassification zones. The bigger the number of zones after the merging procedure, the more accurately

the model can predict the location of faults. Hence, the goal of the optimization procedure is to find such a location for smart meters which results in the maximum number of zones.

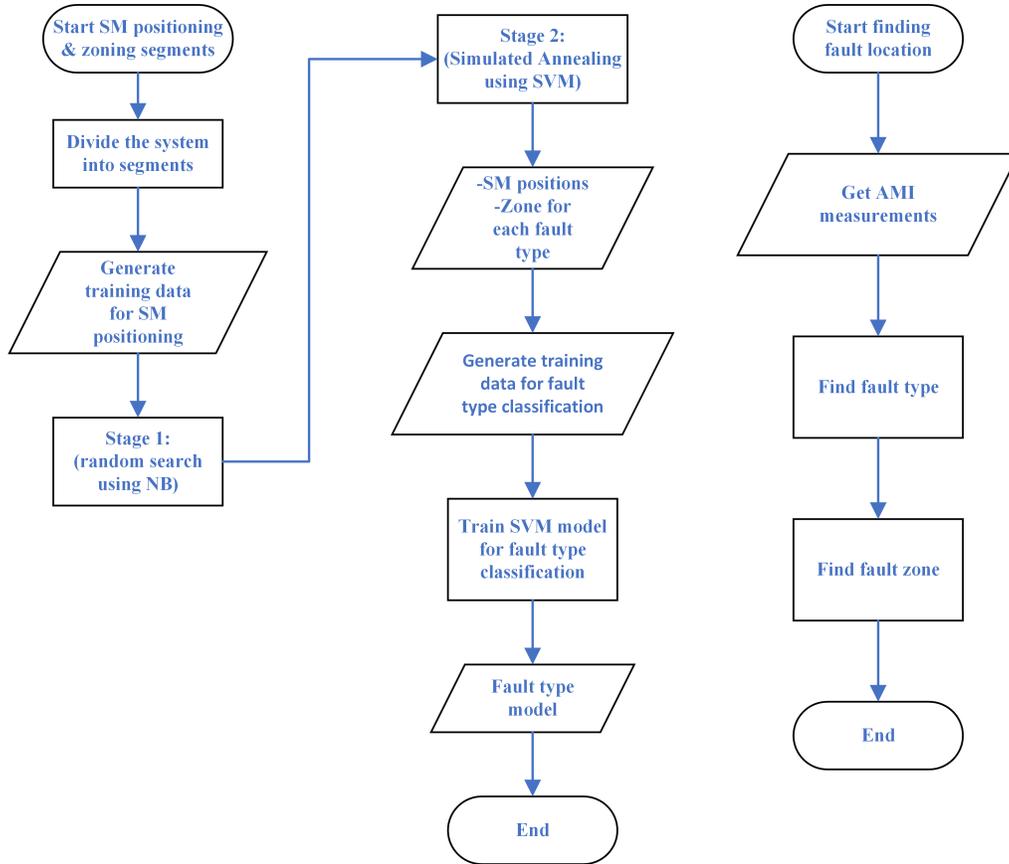


Figure 4.2: Flowchart for the optimal placement process and fault location detection process

Fig. 4.2 depicts the flowchart of the proposed process for the optimal placement of smart meters, as well as the flowchart for fault location detection after the optimal placement is obtained and the classifiers are trained. The process is performed in two stages:

- **1st stage** is a random search in which smart meters are placed randomly across a distribution grid, and an NB classification model is built and used for classifying segments; the NB classifier is chosen for the first stage because it is fast, however its performance is not as good as of the SVM; the output of the first stage is an initial smart meter positioning for the second optimization stage.

- **2nd stage** includes a Simulated Annealing based search to further optimize the location of smart meters, while the SVM is used for classification; the outputs of the second stage are the optimal locations for smart meters, zoning maps and fault location SVM classifiers; a separate SVM is generated for each type of fault.

Once the optimal placement of smart meters is achieved, a training set is built with voltage measurements at the obtained locations of smart meters. It is done using powerflow study under fault conditions. Each data point – measurements for a specific fault – is labeled with the type of simulated fault as depicted in Table 4.1. Then, an SVM-based classifier for detecting the type of fault is constructed.

Therefore, the final result of the proposed algorithm includes: optimal locations for smart meters, a zoning map and a fault-locating SVM classifier for each type of fault, and an SVM classifier for identifying the type of fault.

Fig. 4.2 also includes the flowchart for predicting the location of fault using the trained models and AMI measurements. Upon occurrence of a fault in a system, voltage readings of smart meters are fed to the SVM fault-type classifier. Once the type of fault is determined, the AMI measurements are fed into the SVM zone classifier for the previously detected type of fault. The zoning map for the detected fault type is then used to interpret the output of the SVM zoning classifier and pinpoint the location of fault.

4.3 Meter Placement Algorithm: Details

In this section details about the proposed algorithm are provided. The details include meter requirements, the procedure of merging segments, the two stages of optimal placement, the justification of selecting both the Naive Bayesian and Support Vector Machine as classifiers used during two optimization stages, and an illustrative example.

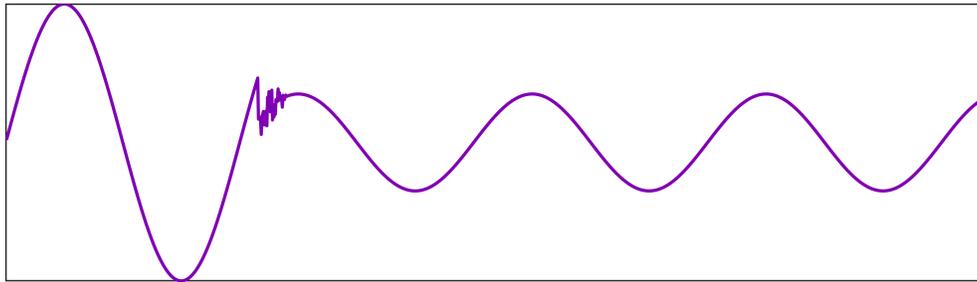


Figure 4.3: A typical voltage sag caused by a fault

4.3.1 Smart Meter Specification

The proposed method assumes that smart meters deployed in a distribution system satisfy the following requirements:

- they are able to record time-based voltage sags;
- they are equipped with a communication technology besides powerline communication;
- they have a backup power source.

A typical voltage sag caused by a fault is illustrated in Fig. 4.3. A voltage below 0.9 p.u. is considered a voltage sag. Voltage sags can happen as a result of temporary and permanent faults or when a large load like a motor is switched on. The smart meter should be able to record time-based voltage sags and send it through AMI. Several existing smart meters have this capability.

The smart meter must have the capability to transfer/send measurements without using powerline communication. It is required due to the fact that protective devices such as sectionalizers, reclosers and circuit breakers de-energize parts or all of the distribution feeder following the occurrence of a fault. As a result, there is no physical connection between de-energized parts of the distribution feeder and the substation. Therefore, powerline communication cannot be used. Other types of smart grid communication technologies are mentioned in Section 2.2.

Following a fault incident, protection devices de-energize all or parts of the downstream feeder. Hence, if a smart meter is not equipped with a backup power source, it will not be able to operate and communicate. This back up power is different and should last more than the residual power that some smart meters store to send the last grasp outage signal.

4.3.2 Process of Merging Segments

It is quite possible that some smart meters record the same voltage sag pattern from the faults happening at two distinguished points on the grid. As a result, the classifier is likely to misclassify the segments with those faults. At each iteration of the proposed optimization procedure, a classification model is trained and a confusion matrix is obtained for the model. The confusion matrix can be obtained using either k-fold cross-validation or a testing dataset which is not used for the training. This matrix reveals which segments are misclassified as locations of faults. A merging routine analyzes the confusion matrix and finds such segments or zones. The segments/zones that have misclassified faults are merged into a new zone.

The segment merging process should be done carefully: the new zones created by merging misclassified segments/zones are not allowed to contain disconnected parts. Hence, not all misclassified segments or zones can be simply merged. Fig. 4.4 shows two examples that illustrate what kind of merging is acceptable and what is not.

The merging routine iteratively checks the confusion matrix for misclassified segments and zones (some segments might be already merged to a new zone during previous iterations of the merging routine), merges them, and updates the confusion matrix. If it finds two misclassified segments or zones, it tries to find a path in a grid that connects those segments or zones. Then, it merges those two segments/zones together with all segments and zones along that path into a single zone, Fig. 4.4. This guaranties the newly created zone can never have disconnected parts.

It should be noted that a radial distribution grid can be modelled as a connected undirected acyclic graph with nodes representing segments/zones.

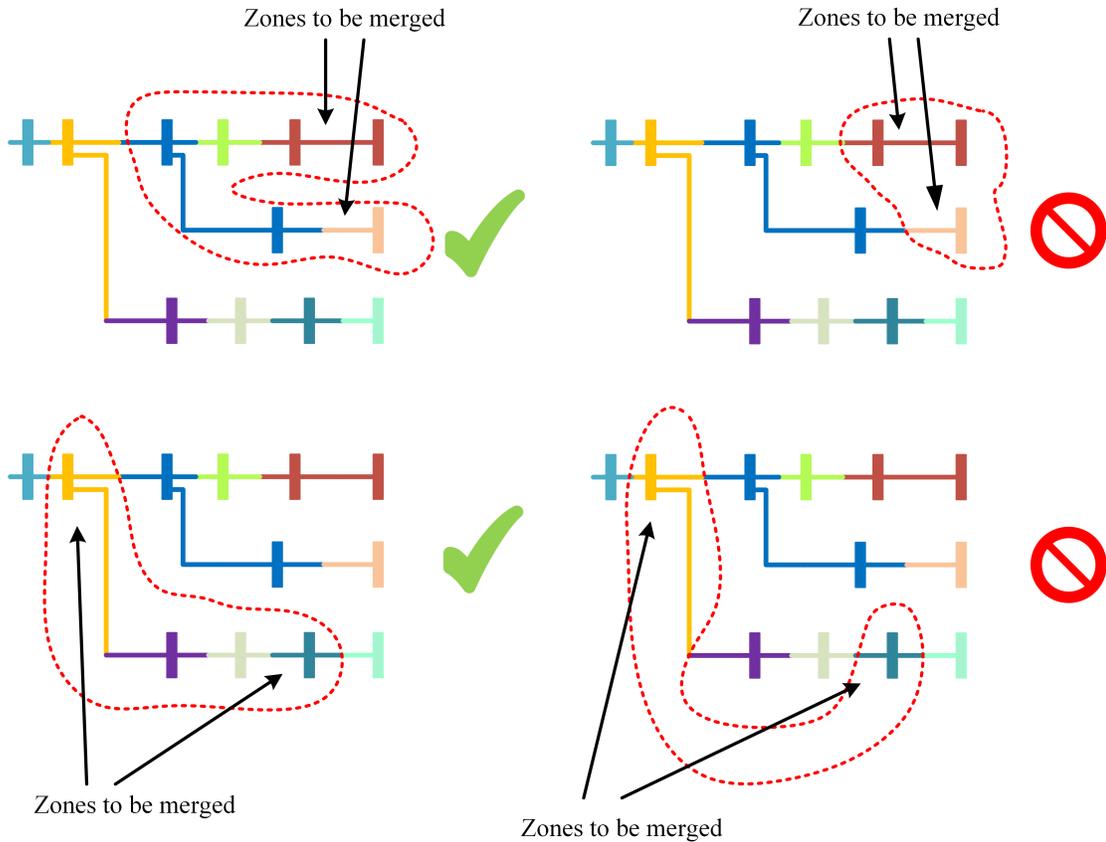


Figure 4.4: (Top): examples of acceptable and wrong ways of merging two zones that are located in two different branches of a grid.

(Bottom): Acceptable and wrong merging for two segments located on a single branch. Each color represents one zone

Such a graph is a tree (considering one-line diagram of the grid or a separate diagram for each phase). In that graph there is only one path between two given nodes. Thus, the merging routine of two zones in a radial system is equal to finding the only path between those zones and merging all zones along that path into a single zone. If the grid is looped or meshed, it can be modelled with an undirected cyclic graph. Hence, there could be more than one path between two to-be-merged zones. In this case, the merging routine finds the shortest path between two merged zones. The shortest path in this case can be defined as the path that contains the least number of segments across the path.

The merging routine iteratively merges misclassified zones and updates the

confusion matrix accordingly, until there is no more misclassifications. This means that the trained classifier is able to identify the fault zone with an absolute accuracy. The output of the merging process is a map comprised of some original segments and some newly created zones. The original segments that were not required to be merged, are treated as zones that contain only one segment.

4.3.3 Meter Placement Procedure: 1st Stage

As mentioned earlier, the optimal placement of smart meters is performed in two stages. The first stage is a random search in which smart meters are placed randomly on load buses across the system, while the NB classifier is used for classifying segments, i.e., identifying segments with a fault.

The output of the first stage is a reasonably good placement of meters that is used as an initial guess for the second optimization stage. Though the NB is not as accurate as the SVM, the fact that it does not require an iterative optimization procedure for training makes it a highly scalable and fast classifier when dealing with large datasets. Hence, it is a good candidate for probing the search space and providing a good initial solution for the next optimization stage. This speeds up the overall optimization procedure, especially in the case that neither Simulated Annealing nor the SVM that are used during the second optimization stage are fast algorithms. A comparison between performance and computational times of the NB and SVM is provided in the further section.

It is assumed that there is a limited number of smart meters available for placing and it does not change during the optimization procedure. At each iteration of the first stage, a random set of load buses is generated for the placement of smart meters. This set is generated based on the following rules:

- At the first iteration, a specific number of smart meters is assigned to each phase. The algorithm tries to equally dispatch smart meters between the three phases. If the number of available smart meters is not divisible by 3, the phases that supply a higher number of loads are as-

signed more smart meters. Unbalanced distribution systems are common. However, this does not mean there is a drastic difference between the number of loads that each phase supplies. Intuitively, installing a similar number of smart meters on each phase helps the algorithm to more effectively capture voltage sag patterns across a grid.

- At each following iteration, smart meters are randomly placed on load buses of the phase they are assigned to. It is not allowed to relocate them to other phases.
- A maximum of one smart meter can be installed at a given load bus. Though it is quite possible that multiple customers are supplied from a load tap, there is no benefit of having more than one smart meter installed on a given load tap, if the fault locating study is concerned. The number of smart meters is limited and the meters connected to a specific load tap would record the same voltage sag pattern, as they are connected in parallel.
- Each DER is equipped with a smart meter and that smart meter cannot be relocated.

At each iteration, an NB classifier is trained for the randomly placed meters. As mentioned in Table 4.1, there are 11 types of fault in a distribution system. Therefore, 11 classifiers, one for each type of fault, have to be trained. There is a separate training dataset for each type of fault. Each dataset contains voltage measurements of all load buses in the system under one out of 11 types of fault condition. However, the classifier only uses voltage measurements of those smart meter locations that are randomly selected at each iteration of the optimization procedure. An NB classifier is trained for each type of fault, and the merging routine is applied to the confusion matrix. That leads to 11 zoning maps at the end of each iteration.

To evaluate the ‘goodness’ of smart meter locations a fitness function is defined. Considering the fact that a higher number of zones means a more

accurate classification of fault locations, a fitness function is defined as follows:

$$F(L_{SM}) = \frac{1}{3}NZ(LG) + \frac{1}{6}(NZ(LL) + NZ(LLG)) + \frac{1}{6}(NZ(LLL) + NZ(LLLG)) \quad (4.1)$$

in which F is the fitness function, L_{SM} is a vector containing the locations of smart meters, and $NZ()$ stands for the number of zones for each type of fault. The coefficients in (4.1) are chosen to put the same emphasis on single, two, and three phase faults. One could modify those coefficients in favour of a specific type of fault. For example, LG faults are the most common faults in distribution systems, so a larger coefficient can be assigned to $NZ(LG)$.

The objective of the first stage is to provide a good initial solution for the second optimization stage. One could argue that it is not critical at this stage to evaluate a fitness function considering all 11 types of fault, and evaluating the fitness function with a fewer number of fault types would suffice. For example, a fitness function could be defined taking into account only single phase to ground and three phase faults:

$$F(L_{SM}) = \frac{1}{2}NZ(LG) + \frac{1}{2}NZ(LLL) \quad (4.2)$$

This argument holds, as the goal of the first stage is only to find an initial solution, not to develop a fault prediction model or a zone map for each type of fault.

Finally, at the end of the first stage, the set of locations that results in the largest fitness value is passed to the second optimization stage as the initial solution. The following is the pseudocode of the first stage:

```

for  $i = 1 : iter_1$ 
  -generate a random location vector for smart meters  $L_{SM}$ 
   $F_i(L_{SM}) = 0, F_{max} = 0$ 
  for  $j = 1 : num_{fault\_type}$ 
    -train an NB classifier for fault_type  $j$ 
    -merge segments and calculate  $NZ(fault\_type_j)$ 
     $F_i(L_{SM}) = F_i(L_{SM}) + coef_j \times NZ(fault\_type_j)$ 

```

end

if $F_i(L_{SM}) > F_{max}$ *then* $F_{max} = F_i(L_{SM})$, $L_{SM}^{best} = L_{SM}$

end

4.3.4 Meter Placement Procedure: 2nd Stage

The purpose of the second stage is to find the optimal placement of smart meters via ‘tuning’ the best random placement obtained at the first stage of the process. It is done using Simulating Annealing as a tool to explore the search space for possible placements of smart meters, and the SVM, as a more sophisticated classifier when compared to the NB, for identifying/classifying segments/zones with faults.

Simulated Annealing [77] performs the optimization of meters’ placement. Details about this optimization algorithm is provided in Appendix A. Simulated Annealing locally probes the search space for the best solution. At each iteration, a set of locations is selected for smart meter placement and the fitness function (4.1) is evaluated. The number of smart meters allocated to each phase remains the same as the first stage and smart meters are locked to phases they were initially assigned to at the beginning of the first stage. However, the placement of meters is not random anymore. Random selection of locations is replaced by generating a new placement as a neighboring solution for the current placement.

In order to generate a neighboring solution, the smart meter vector placement L_{SM} is modified. The modification of the vector is done in a way that one smart meter ‘changes’ its location. A smart meter (one member of L_{SM}) is randomly selected, and all immediate adjacent load buses for this smart meter (located on the same phase) are identified. One of these adjacent load buses is randomly selected, and the smart meter is moved to that randomly selected bus. Let us illustrate the process. Fig. 4.5 shows neighbor load buses for three smart meters, each assigned to one phase. The smart meter which is installed on phase C in that figure can only be moved to a neighbor location on the same phase. It is despite the fact that the smart meter is measuring a branch of a Δ -connected load, so it is technically connected to both phases A

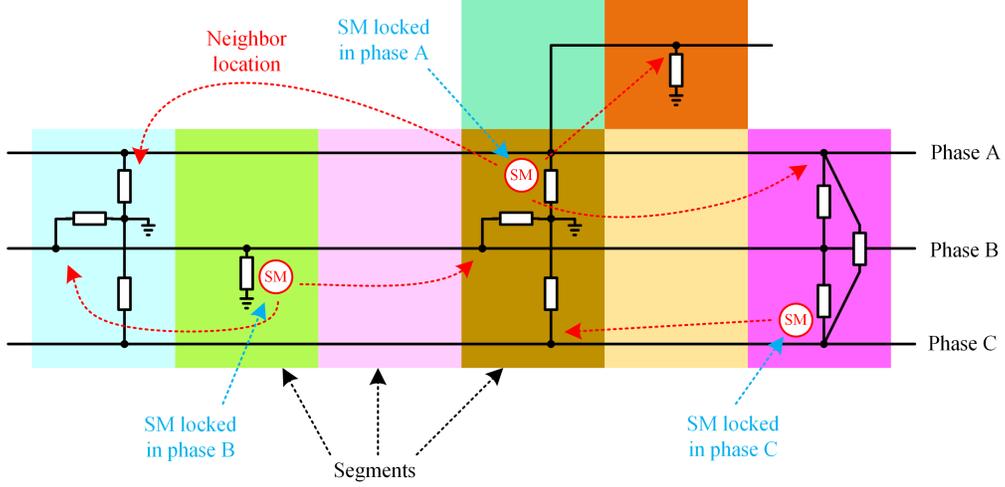


Figure 4.5: Neighboring solutions for three smart meters, each assigned to one phase

and B.

The SVM performs classification during the second stage. The fitness function (4.1) includes all 11 types of fault. The outputs of this stage are: the optimal locations for smart meter placement, a zoning map for each type of fault, as well as 11 SVM classifiers, each for one type of fault. The pseudocode for this stage is following:

```

 $F_c = F_{first\_stage}, T = T_{init}, L_{SM}^c = L_{SM}^{first\_stage}$ 
for  $i = 1 : iter_2$ 
  -reduce temperature:  $T = c \times T, 0 < c < 1$ 
  -generate a neighboring solution  $L_{SM}^{new}$  for current solution  $L_{SM}^c$ 
  - $F_{new}(L_{SM}^{new}) = 0$ 
  for  $j = 1 : num_{fault\_type}$ 
    -train an SVM classifier for fault_type  $j$ 
    -merge segments and calculate  $NZ(fault\_type_j)$ 
     $F_{new}(L_{SM}^{new}) = F_{new}(L_{SM}^{new}) + coef_j \times NZ(fault\_type_j)$ 
  end
  if  $F_{new} \geq F_c$  then  $F_c = F_{new}, L_{SM}^c = L_{SM}^{new}$ 
  elseif  $rand() \geq \exp(\frac{F_c - F_{new}}{kT})$  then:  $L_{SM}^c = L_{SM}^{new}, F_c = F_{new}$ 
end

```

Each iteration of the optimization process involves training a classifier and invoking the process of merging segments. This means that using analytical optimization algorithms is not possible. So, a metaheuristic algorithm has to be employed. Simulated Annealing has been selected because it works well as a local search algorithm. In this case, the process of identifying neighbor solutions for the current solution can be easily defined. This is not the case for population-based methods like PSO. For PSO, development of a method that controls movement of particles towards each other in a distribution system with several branches seems complicated. Otherwise, any optimization algorithm with a well-defined method for exploring nodes of a distribution grid can be employed instead of Simulated Annealing.

It is worth mentioning that the proposed approach is not limited to the use of the NB and SVM classifiers. Any fast classifier with an acceptable accuracy can be used for the first stage of the optimization, while any high-accuracy classifier with an acceptable computational time can perform the classification task during the second stage.

Once the optimization procedure is finished, an SVM classifier is trained for detecting a single type of fault. The training dataset includes voltage measurements for optimally placed smart meters, while labels represent types of fault.

4.3.5 Fault Locating Procedure

Detecting fault locations based on the proposed method is straightforward. Assuming that the smart meters are placed at locations suggested by the proposed algorithm, the process of locating a fault can be performed according to the following three steps:

1. collecting voltage sag measurements from smart meters; picking the minimum (smallest amplitude) of voltage from each meter recorded at the time of fault occurrence; if a smart meter has not recorded a time-based voltage sag (because voltage for that meter has not dropped below 0.9 p.u.), the minimum recorded voltage of that smart meter is taken; the

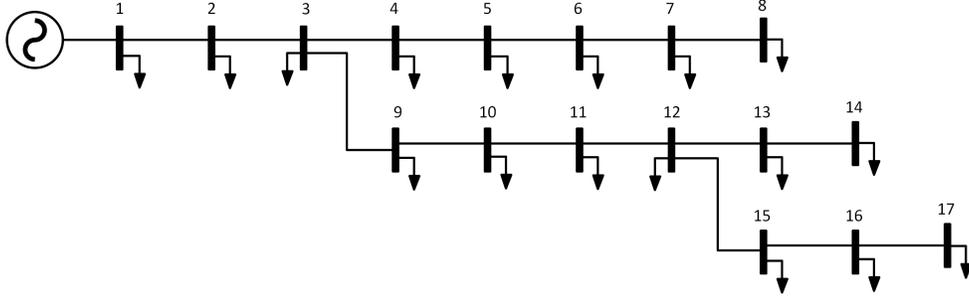


Figure 4.6: The 17-bus single-phase model used for illustrating the optimal placement procedure

minimum recorded voltage is likely caused by the fault;

2. inputing voltage measurements to the SVM type classifier which identifies the fault type;
3. inputing voltage measurements to the SVM fault location classifier as well as using the zone map for the detected fault type to identify/find the zone where the fault occurred.

4.3.6 An Illustrative Example

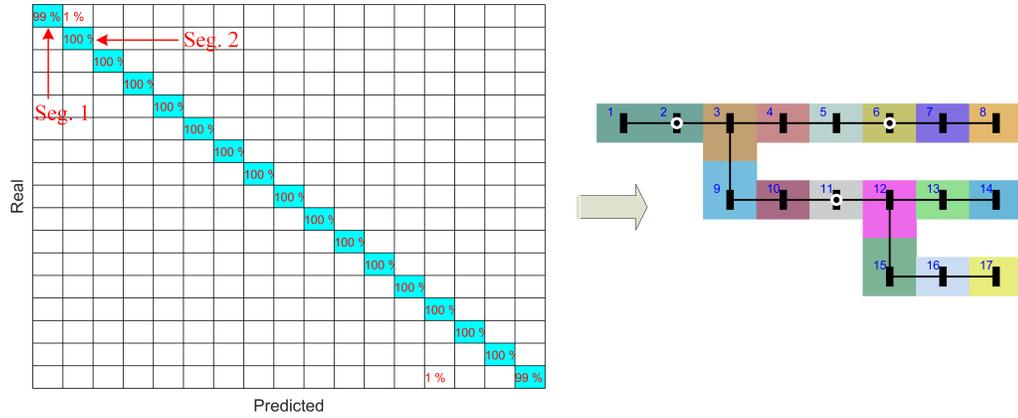
Before we present the simulation results of applying the proposed method to IEEE34 test system (Section 4.4), we show how the two stages of optimal placement and merging routine work when they are applied to a small 17-bus single-phase system, Fig.4.6. The system has 17 buses and each bus is supplying a load. It is divided into 17 segments, where each segment encloses a bus. It is assumed that faults can only happen on the buses, not on the lines. So, lines are not considered as parts of segments. The number of iterations is set to 3 for the first stage, and 20 for the second. Three smart meters are available for placement.

- First iteration of the first stage: Fig. 4.7 shows the merging procedure for the first iteration. Buses 2, 6, and 11 are randomly chosen for placement of smart meters, and an NB classifier is trained. The confusion matrix is calculated and the merging routine is invoked. By observing the confusion matrix at the first iteration of the merging routine, it is obvious

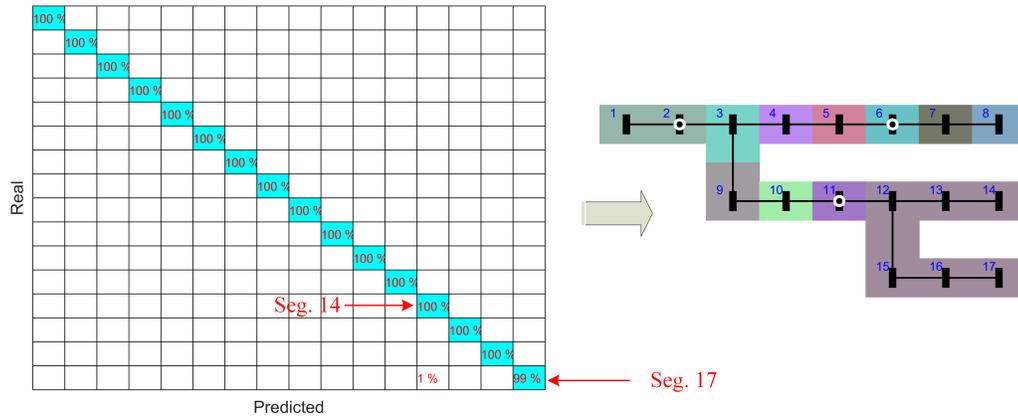
that in some cases segment 2 is misclassified as segment 1. Therefore, the merging routine merges segments 1 and 2 into a new zone. The merging result is shown in Fig. 4.7, where each color represents one zone and white circles represent the location of smart meters. The confusion matrix is updated after merging those two segments. The updated matrix shows that segment 17 is misclassified as segment 14. So, those two segments have to be merged. However, those two segments are not adjacent. Therefore, the merging routine has to find a path between them. This results in merging six segments 14, 13, 12, 15, 16, and 17 into a single zone. The confusion matrix is updated and as it is shown in Fig. 4.7, there is no misclassifications. So, the merging routine terminates, resulting in 11 zones.

- Second iteration of the first stage: Figures 4.8 and 4.9 show the merging procedure at the second iteration. The randomly chosen locations for smart meters are buses 2, 10, and 15. At the first iteration of the merging routine, segments 1 and 2 are merged into one zone. Let us refer this zone as zone 100. At the second iteration, the confusion matrix shows that some samples of the newly created zone 100 are misclassified as segment 7. A path between zone 100 and segment 7 is found, and all the zones (each segment is a zone as well) along the path are merged into zone 101. The updated zone map is shown in Fig. 4.8. At the third iteration, zone 101 and segment 8 are merged. This creates a new zone called zone 102. The updated confusion matrix for the fourth iteration implies that segment 10 is misclassified as zone 102. A path between the zone 102 and the segment 10 includes the segment 9 as well. Therefore, all these three zones are merged to a single zone, shown in Fig. 4.9. The updated confusion matrix at the fifth iteration reveals no misclassification. So, the merging routine terminates with 8 zones.
- Third iteration of the first stage: Buses 5, 8, and 14 are randomly chosen, Fig. 4.10 shows the final zone map for this selection. As we can see, this locations of smart meters results in only 5 zones. After finishing three

First iteration of merging routine:



Second iteration of merging routine:



Third iteration of merging routine:

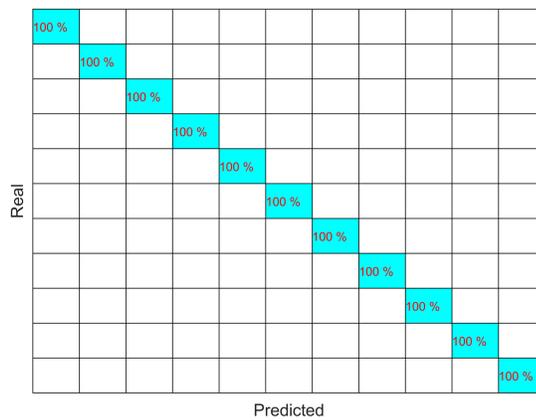


Figure 4.7: The merging routine procedure at the first iteration of the first stage

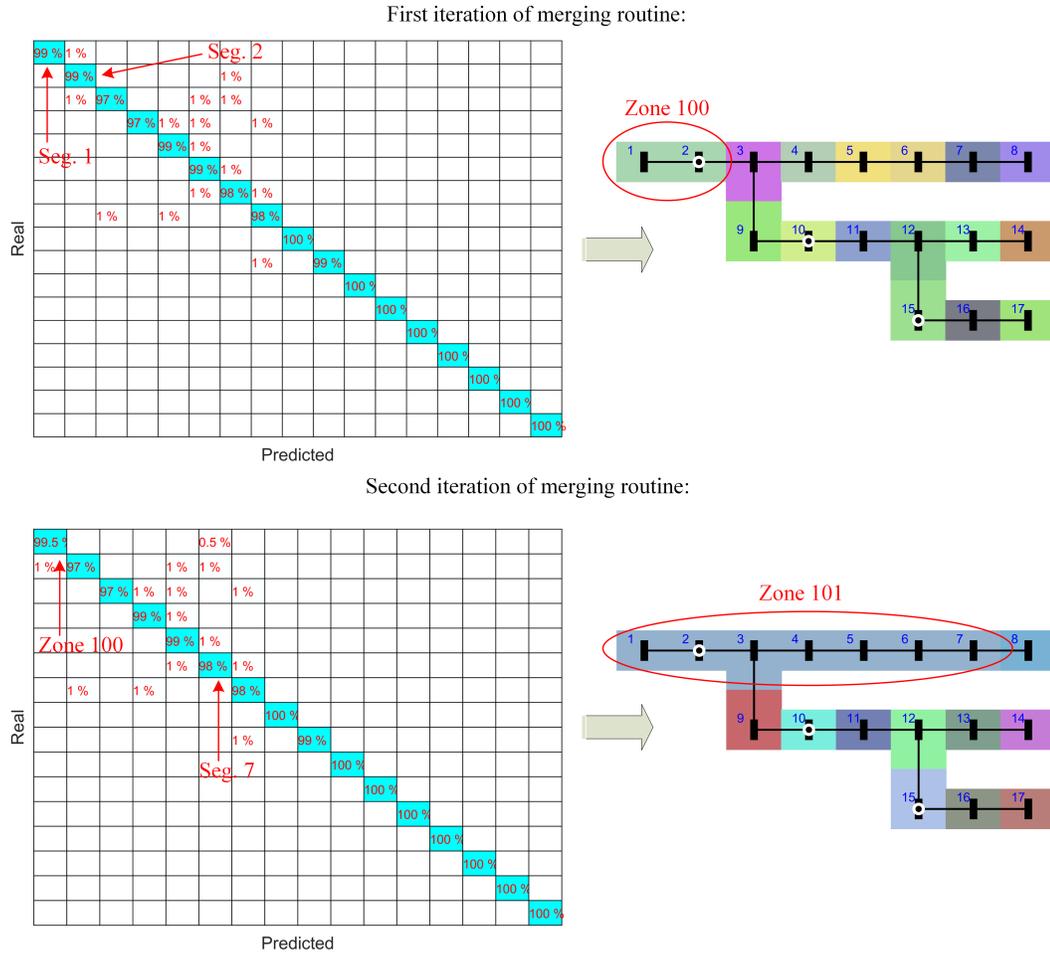
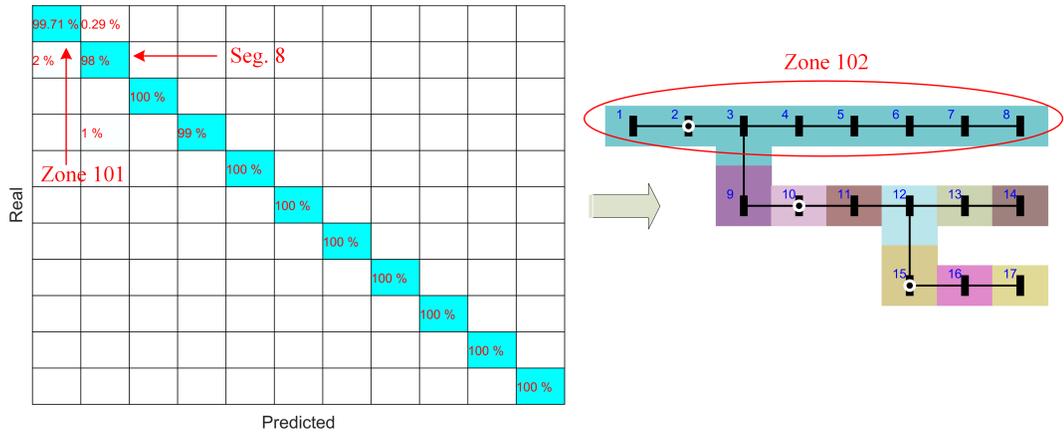


Figure 4.8: The first two iterations of the merging routine at the second iteration of the first stage

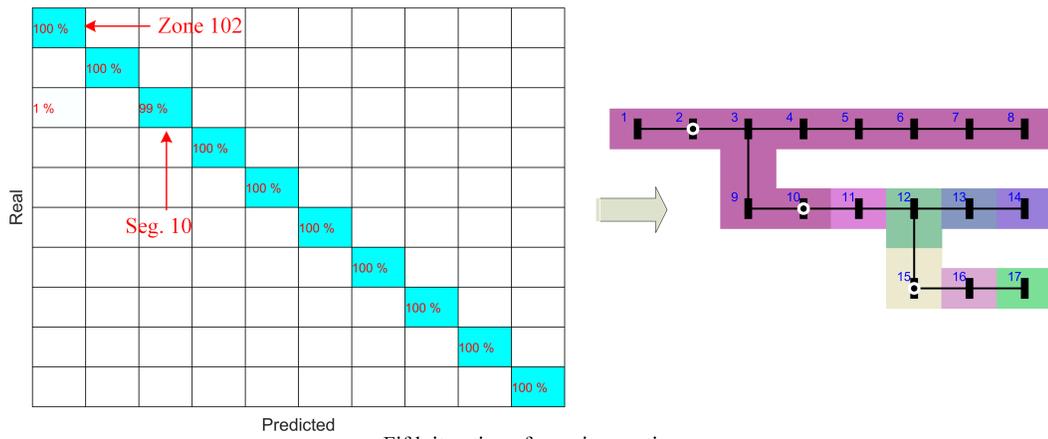
iterations, it is obvious that the set of buses 2, 6, and 11 has the best performance with 11 zones. Hence, this set is sent to the next stage as the initial solution.

- The second optimization stage: Simulated Annealing runs for 20 iterations and the SVM is used for classification. Fig. 4.11 represents the final results of this stage, which are a zoning map and the optimal locations for smart meter placement. The optimal locations are buses 1, 5, and 12, which result in 15 zones.

Third iteration of merging routine:



Forth iteration of merging routine:



Fifth iteration of merging routine:

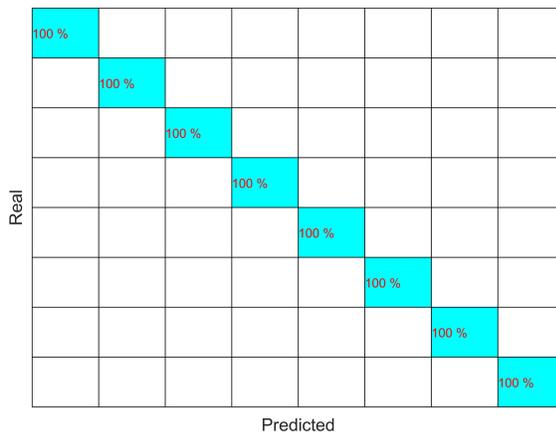


Figure 4.9: Iterations 3 to 5 of the merging routine at the second iteration of the first stage

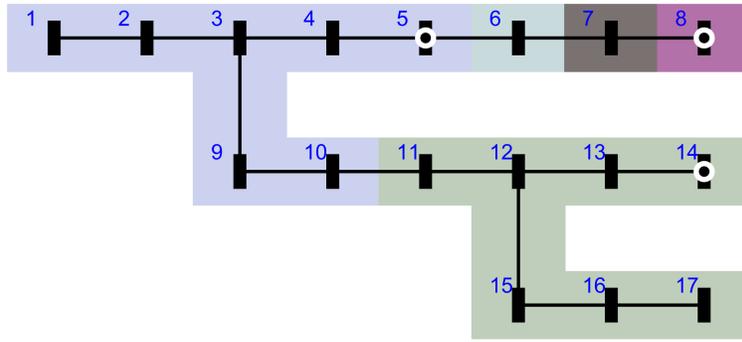


Figure 4.10: The zone map for the third iteration of the first stage

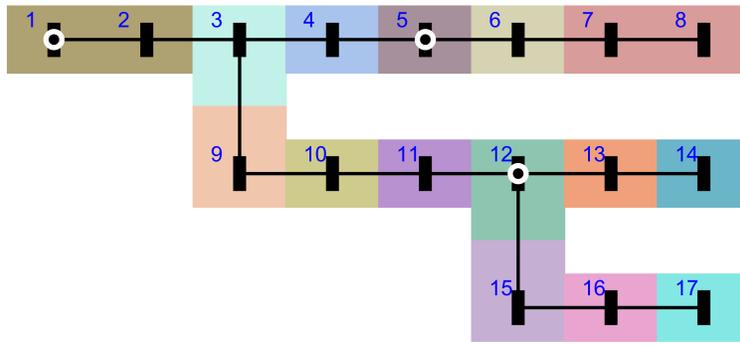


Figure 4.11: The zone map and optimal location of smart meters at the end of the optimization procedure

4.4 Simulation Results

The first part of this section compares the performance of two classifiers NB and SVM in terms of accuracy and training time. The second part shows the simulation results of applying the proposed optimal placement algorithm to IEEE34 test bus system, while the third part includes the simulation results when the proposed approach is executed on an extended version of IEEE34 system.

4.4.1 Comparison between the NB and SVM

Fig. 4.12 shows one-line diagram of IEEE34 system [45]-[46]. A 50 kVA single-phase DER is added to the phase B of bus 838. The powerflow study for generating training sets is performed using OpenDSS, while the implemen-

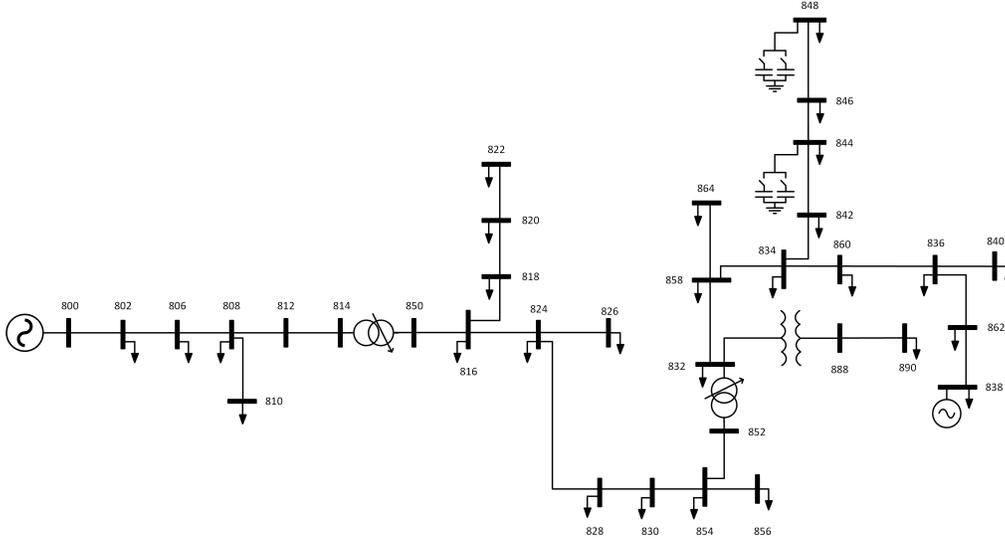


Figure 4.12: IEEE34 test bus system

tation of the proposed algorithm is done in MATLAB. Machine learning and statistics toolboxes of MATLAB are used for training NB and SVM classifiers. The system that runs the code has an Intel CoreTM i7 3.5 GHz processor with 4 hyper-threaded cores and 32 GB memory.

Table 4.2 shows the comparison results between the NB and SVM classifiers based on 9 randomly placed smart meters. The training dataset contains 4600 samples of LG fault. The machine learning toolbox of MATLAB performs hyperparameter optimization for both classifiers by optimizing parameters of the SVM and picking the best multi-class method (one-versus-one or one-versus-all), as well as finding the right sample distribution for the NB.

The training time and the number of zones for the SVM with linear, polynomial, and Gaussian kernels, along with the results for the NB are provided in that table. The results are averaged over 5 runs of each method. It can be observed that the SVM with Gaussian kernel outperforms other methods in terms of accuracy with an average number of zones of 27.0 out of total 34. It is assumed that each segment contains one bus and faults only happen on buses. So, the lines are not considered. Surprisingly, the performance of the NB is better than the SVM with linear and polynomial kernels.

In terms of the training time, the NB is the fastest with only 38.02 s

Models:	NB	SVM (linear)	SVM (polynomial)	SVM (Gaussian)
Num. of zones:	11.0	9.6	9.5	27.0
Sim. time: (s)	38.02	1096.88	2583.14	200.06

Table 4.2: Comparison between the NB and SVM classifiers in terms of accuracy and training time

average time, followed by the SVM with Gaussian kernel with 200.06 *s*, both much faster than the SVM with linear or polynomial kernels. It is worth mentioning that the training was done on 4 hyper-threaded cores what means that the sequential run will increase the training time. For example, the time for the sequential run of the NB and SVM with Gaussian kernel is 111.20 *s* and 1093.35 *s*, respectively.

Based on the simulation results presented in Table 4.2, one can conclude that the NB is faster than other methods, so it is the right choice for randomly exploring the search space during the first stage of the optimization. In addition, the SVM with Gaussian kernel is the most accurate classifier. Therefore, it is suitable for the second optimization stage which requires high accuracy.

The simulation results, Table 4.2, prove that the SVM with Gaussian kernel has superior accuracy compared to the NB. So, the SVM is likely to result in a greater number of zones compared to the NB for a given set of meter locations. This raises a question whether considering the maximum number of zones as the fitness function for the first stage of optimization is the right choice, or there are other better options. In order to investigate the best fitness function for the first stage, a study is performed with three different fitness functions:

1. minimization of misclassification ratio;
2. maximization of the number of zones with 100% accuracy;
3. maximization of the number of zones with 98% accuracy.

To evaluate the first fitness function, misclassification ratio (the percentage of misclassified samples over the total samples in the confusion matrix) is minimized, while the merging procedure is omitted. This can be considered as the fitness function for a typical feature selection method. The second fitness func-

Fitness:	Min. ratio	Max. Num. of zones (100% accuracy)	Max. Num. of zones (98% accuracy)
Num. of zones:	22.75	27.75	25.5

Table 4.3: Comparison between different fitness functions for the first optimization stage

tion is the function explained and used in this study so far, similar to the one used in the illustrative example. This fitness function requires the confusion matrix to be absolutely accurate after the merging procedure is done. The third fitness function is similar to the second function, yet it does not enforce the absolute accuracy during the merging procedure. Instead, the requirement for classification accuracy for each zone is set up to 98%. For example, if up to 2% of samples of zone A are misclassified as zone B, the merging routine will not merge those two zones.

The study is done by running the first optimization stage using each of the mentioned fitness functions for 15 iterations and running the second stage with the SVM for a single iteration. The training set is similar to the set used in the comparison study between the SVM and NB on IEEE34 system, while 9 randomly placed meters are used. Table 4.3 represents the simulation results of this study. Results are averaged over 4 runs of each method. Simulation results show that the maximum number of zones with 100% accuracy is indeed the best fitness function for the first optimization stage. Simulation results also suggest that a typical feature selection method that employs a fitness function such as the minimization of misclassification ratio is not suitable for the first optimization stage.

4.4.2 Simulation Results For IEEE34

This section presents simulation results when the proposed optimal placement method is applied to IEEE34 test system. It is assumed that the voltages of all three phases of the main bus (bus 800) are always measured and the DER added to bus 838 is equipped with a smart meter. Six smart meters are available for placement. The following fitness function is considered for both

Type:	LG	LLab	LLbc	LLca	LLGab	LLGbc	LLGca	LLL	LLLG
NZ:	30	24	24	22	18	22	21	25	23

Table 4.4: Number of zones for each type of fault after optimal meter placement

optimization stages:

$$\begin{aligned}
F(L_{SM}) = & \frac{1}{3}NZ(LG) + \frac{1}{18}(NZ(LLab) + NZ(LLbc) \\
& + NZ(LLca) + NZ(LLGab) + NZ(LLGbc) + NZ(LLGca)) \\
& + \frac{1}{6}(NZ(LLL) + NZ(LLLG))
\end{aligned} \tag{4.3}$$

All training samples for LGa, LGb, and LGc are combined into one training set LG. This means that at the end of the optimization procedure there will be 9 zone maps, one for LG faults and one for each other type of fault. The training set for LG includes 3860 samples, while training sets for other types of fault contain 1120 samples, each. The first optimization stage is done with 25 iterations, while the Simulated Annealing in the second iteration runs for 40 iterations. The confusion matrix is calculated based on 10-fold cross-validation. It is assumed that faults occur only on buses not lines. Based on this assumption, IEEE34 system is divided into 34 segments for LG faults and 26 segments for other types of fault. The reason for the difference in the number of segments is that some buses are single-phase, hence double-phase and three-phase faults cannot be simulated on them.

Figures 4.13-4.15 show zoning maps for 9 types of fault and optimal locations for 6 smart meters. The number of zones for each type of fault is presented in Table 4.4. One can observe that the number of zones for all types of fault are high. This means that the proposed algorithm has placed the smart meters very well. This is despite the fact that there was only 6 meters available.

The fitness function (4.3) ensures that the average of the number of zones for 9 types of fault is maximized. This means that the algorithm is not meant to maximize the number of zones for every single type of fault. The reason is obvious: the set of meter locations that results in the maximum number of

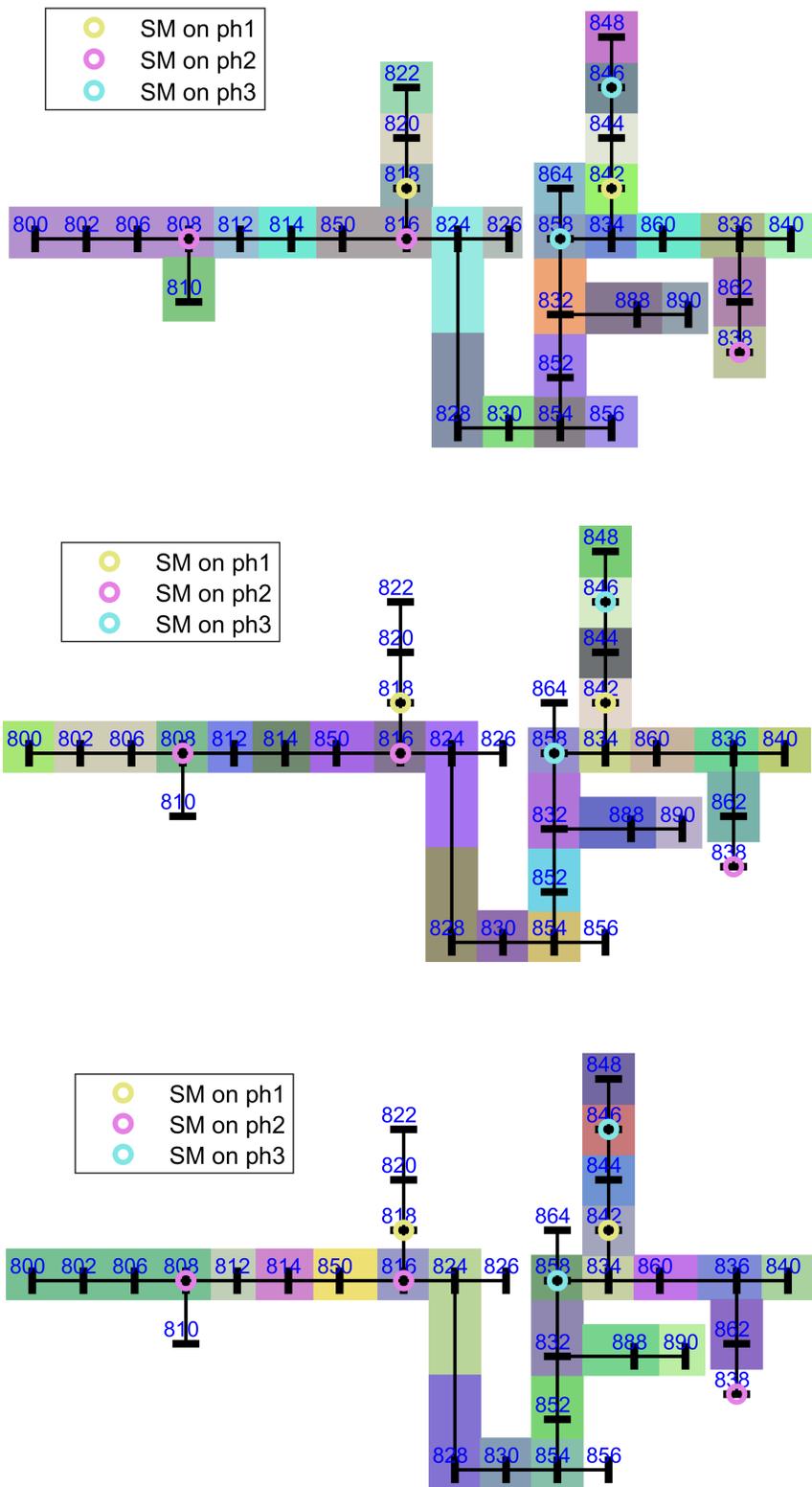


Figure 4.13: (Top): zone map for LG faults, (Middle): zone map for LLL faults, (Bottom): zone map for LLLG faults.

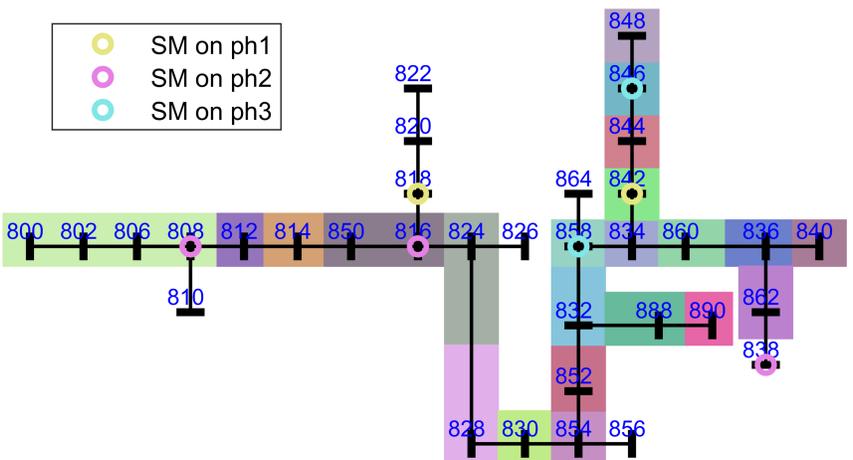
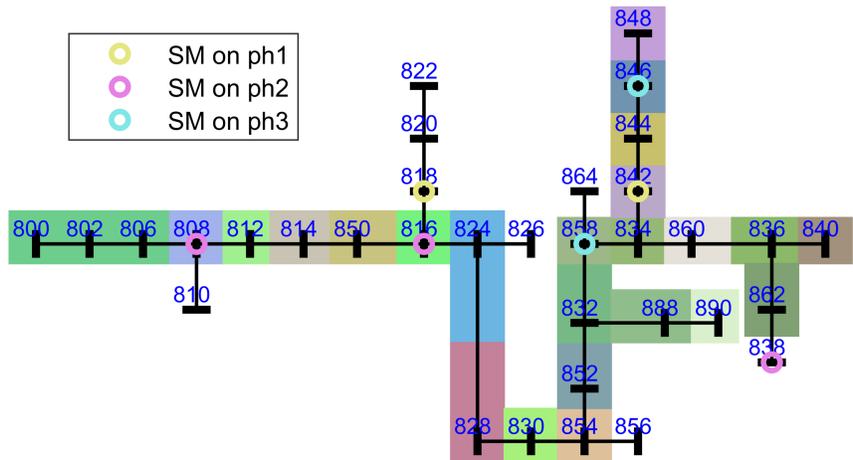
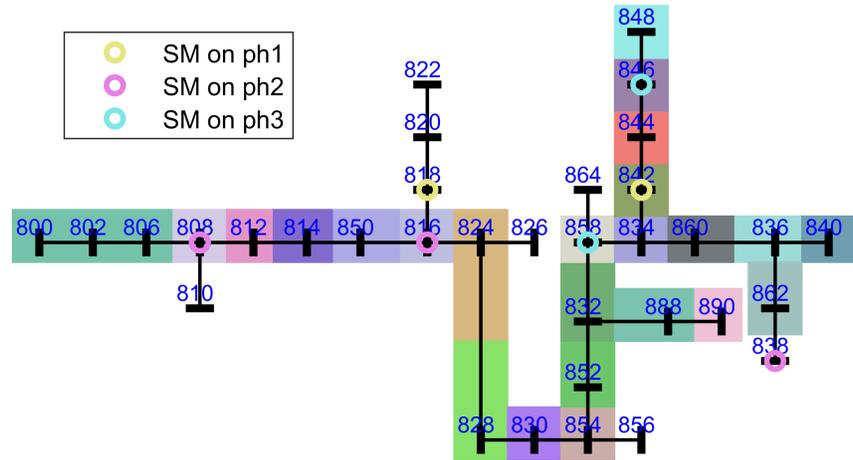


Figure 4.14: (Top): zone map for LLab faults, (Middle): zone map for LLbc faults, (Bottom): zone map for LLca faults.

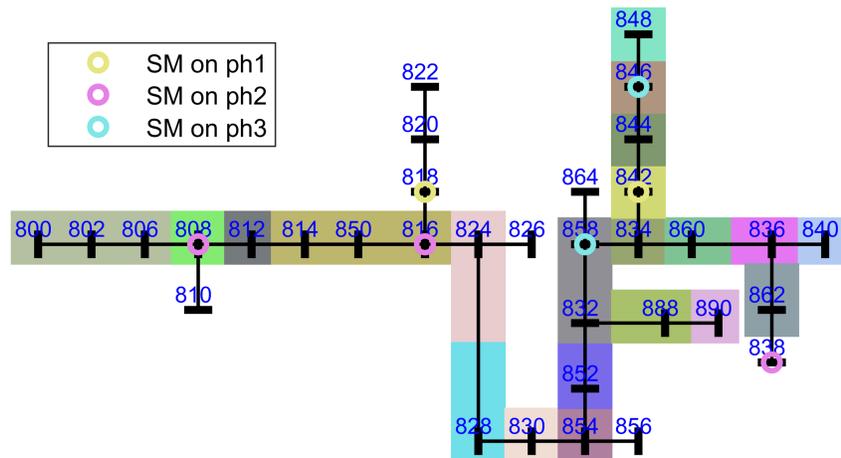
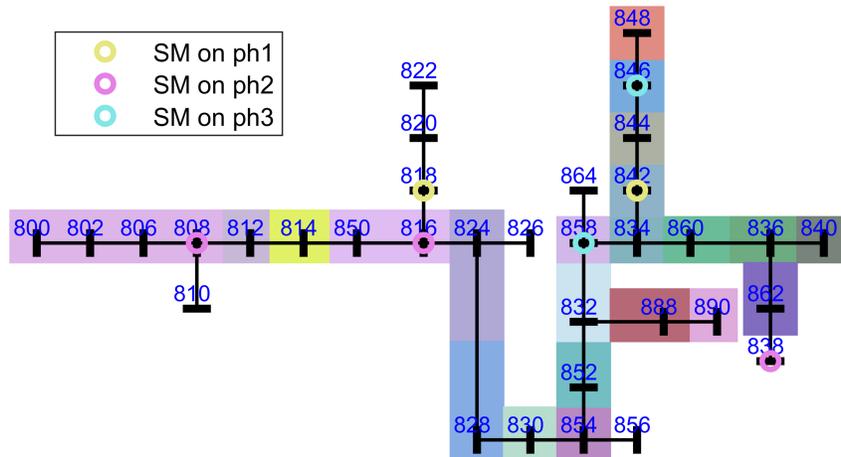
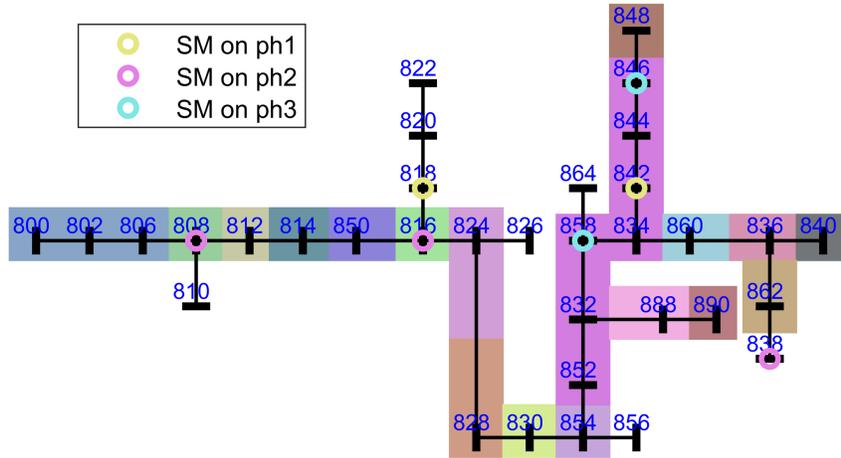


Figure 4.15: (Top): zone map for LLGab faults, (Middle): zone map for LLGbc faults, (Bottom): zone map for LLGca faults.

Type:	LG	LLab	LLbc	LLca	LLGab	LLGbc	LLGca	LLL	LLLG
NZ: (Overall)	30	24	24	22	18	22	21	25	23
NZ: (Each)	30	25	24	23	22	24	23	25	25

Table 4.5: Comparison between the number of zones if the fitness function (4.3) is replaced by individual $NZ()$ for each fault type

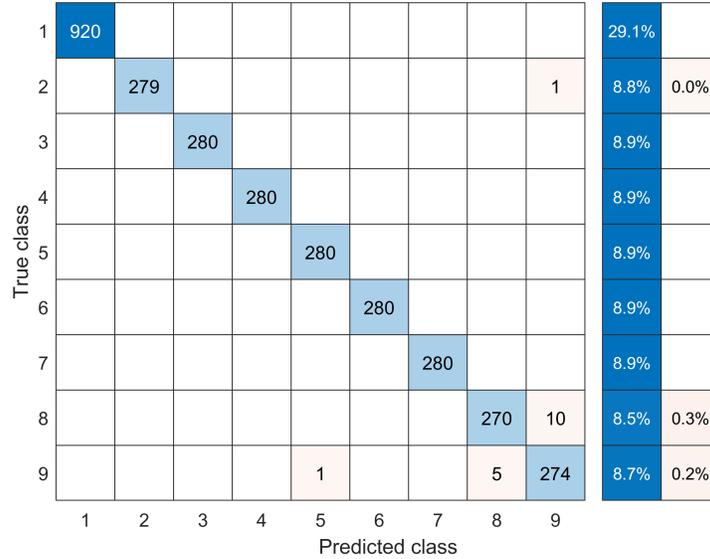


Figure 4.16: Confusion matrix for the SVM classifier of types of fault in IEEE34 system (classes are numbered in order as in (4.3))

zones for one type of fault is not necessarily the same as the set of locations that maximizes the number of zones for another type of fault. Table 4.5 shows what the number of zones for each type of fault would be if the fitness function (4.3) had only focused on individual $NZ()$ for each type of fault. This requires 9 separate runs of the optimization algorithm, each time considering the number of zones only for one type of fault.

After the meters are placed, an SVM classifier has to be trained to detect the type of faults. To do so, a training set with 3160 samples labelled for 9 types of fault in (4.3) is generated and used for the training purposes. Fig. 4.16 shows the confusion matrix for the trained classifier. As can be seen, it can effectively detect the types of fault, although it slightly misclassifies LLL and LLLG faults as each other.

Fault type:	LG	LL/LLG	LLL/LLLG
Accuracy: (%)	100	99.79	99.82

Table 4.6: The accuracy of the proposed method in locating single, double, and three-phase faults

To evaluate the accuracy of the trained SVM classifier in locating faults, a testing set containing 10% of the training set has been created. The test dataset includes voltage measurements for locations indicated by the proposed optimal placement algorithm. The following table shows the accuracy of trained SVM classifier in locating the zones of faults. The results prove the high accuracy of SVM classifiers and zoning maps in the task of locating faults.

4.4.3 Simulation Results For Extended IEEE34

Simulation results of applying the proposed method to an extended version of IEEE34 test system is presented in this section. Details about this test system can be found in Appendix B. For the rest of this study, this extended version of IEEE34 is referred as extended IEEE34 test system. A typical distribution system is comprised of spot and distributed loads. As their name suggests, distributed loads are spread across laterals in the system. Modelling distributed loads one by one is cumbersome, as it requires defining a bus for each load and breaking the line that supplies distributed loads into several pieces. Instead of modelling individual distributed loads, most powerflow models divide the total amount of distributed loads across a line into two equal portions, and assign those two portions to the buses on the two sides of that line. This method of load modelling, which is also used in the original IEEE34 test system, makes circuit modelling easy. However, it is by no means an accurate circuit model when the placement of smart meters is concerned. Extended IEEE34 test system spreads distributed loads across lines. Therefore, it is a more realistic circuit model for the meter placement study compared to the original IEEE34 system.

There is a single-phase 50 kVA DER at the phase B of bus 838. The system

is divided into 351 segments where the maximum length of each segment is 1000 feet. This is a total number of segments, although the number of segments for each type of fault is less than 351. There are 9240 and 8400 training samples for LGa and LGb, respectively. The number of training samples for other types of fault is 7110. The following fitness function is considered for the first stage:

$$\begin{aligned}
F(L_{SM}) = & \frac{1}{9}(NZ(LGa) + NZ(LGb) + NZ(LGc)) \\
& + \frac{1}{9}(NZ(LLGab) + NZ(LLGbc) + NZ(LLGca)) \\
& + \frac{1}{3}NZ(LLLG)
\end{aligned} \tag{4.4}$$

For the second stage, the following fitness is used, which considers all 11 types of fault:

$$\begin{aligned}
F(L_{SM}) = & \frac{1}{9}(NZ(LGa) + NZ(LGb) + NZ(LGc)) \\
& + \frac{1}{18}(NZ(LLab) + NZ(LLbc) + NZ(LLca)) \\
& + \frac{1}{18}(NZ(LLGab) + NZ(LLGbc) + NZ(LLGca)) \\
& + \frac{1}{6}(NZ(LLLG) + NZ(LLL))
\end{aligned} \tag{4.5}$$

The voltages of all three phases of the bus 800 are measured. Also the DER is equipped with a meter. Besides the smart meter of the DER, there are 15 smart meters available. Faults can happen anywhere on the system whether it is a bus or a line. The number of iteration for the two stages of the optimization is 10 and 15, respectively. Table 4.7 shows the number of zones for each type of fault after the optimization procedure is done. The average number of zones based on (4.5) is 201. This means that the proposed optimal placement method has successfully placed smart meters in extended IEEE34 test system, considering the fact that there was only 15 meters available for placement. It is worth mentioning that achieving the maximum accuracy (which means the number of zones is equal to the number of segments) might never be possible for many distribution grids. This is due to the fact that smart meters can be only installed at load buses, while there could be no tapping

Fault type:	Number of Zones	Number of Segments
LGa	252	308
LGb	206	280
LGc	205	237
LLab	183	237
LLbc	206	237
LLca	220	237
LLGab	204	237
LLGbc	205	237
LLGca	195	237
LLL	190	237
LLLG	170	237

Table 4.7: Number of zones for each type of fault

point in some portions of a grid for installing smart meters. This reduces the ability of measurement based methods in locating faults.

Fig. 4.17 represents the zoning map for LGa, while Fig. 4.18 shows the zoning map for LLL. Zoning maps for other types of fault are not shown. Finally, Fig. 4.19 presents the confusion matrix of the SVM classifier trained for identifying types of fault. A training set of 13605 samples has been used for training the SVM.

4.5 Summary

In this chapter a novel method for the optimal placement of smart meters was proposed. The goal of the optimal placement was to find the best locations for installing a limited number of meters, so that faults can be located with the maximum accuracy. In addition to finding the best meter placement, the proposed algorithm divides a distribution system to the maximum number of possible zones, which guarantee the maximum fault-locating accuracy. The simulation results of applying the proposed method to both test systems IEEE34 and extended IEEE34 prove the excellent performance of the proposed approach in optimal placement of smart meters and locating faults. The following observations can be made about the proposed method.

- The goal of the proposed algorithm is to find the best locations for

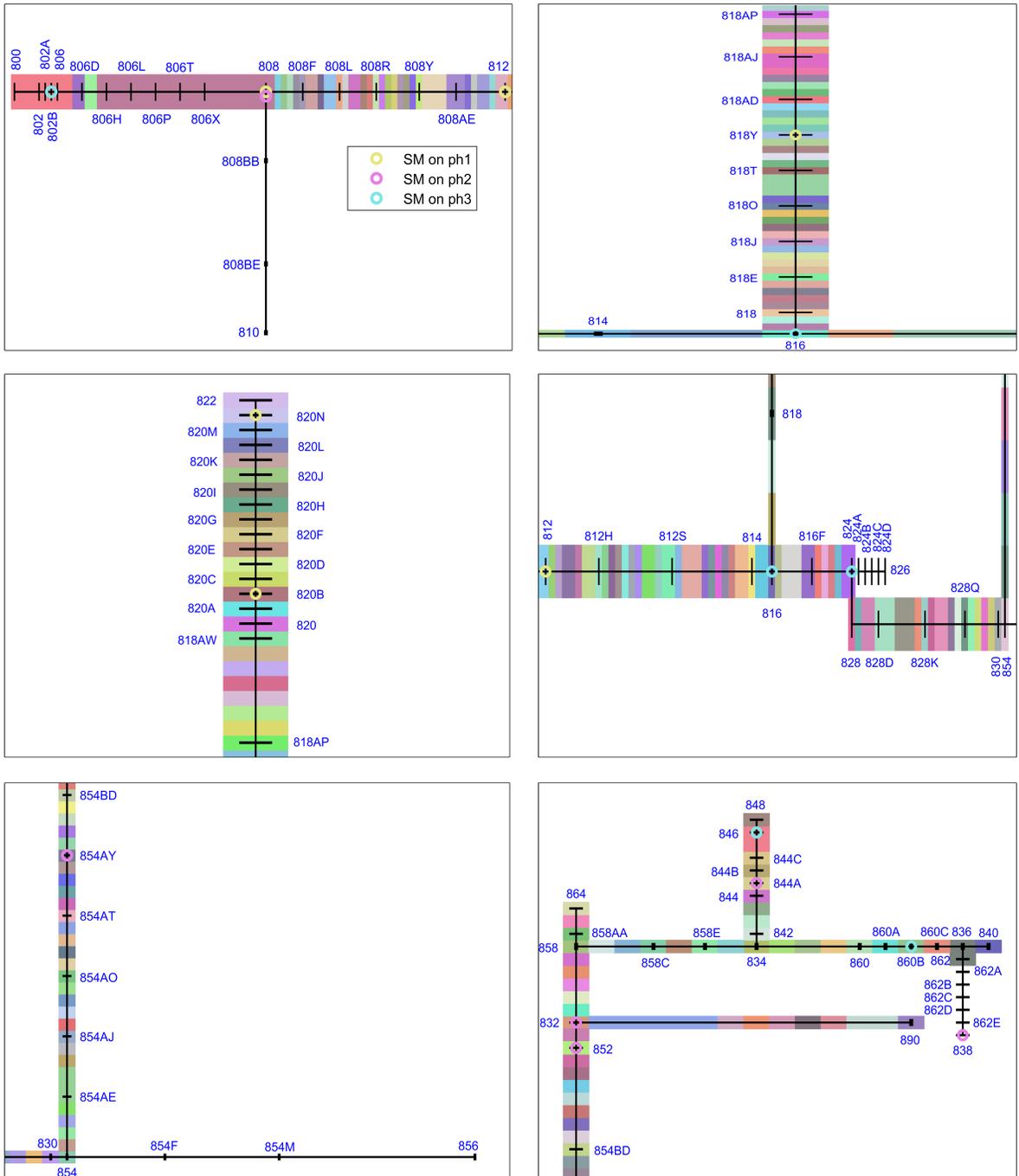


Figure 4.17: Zoning map for LGa

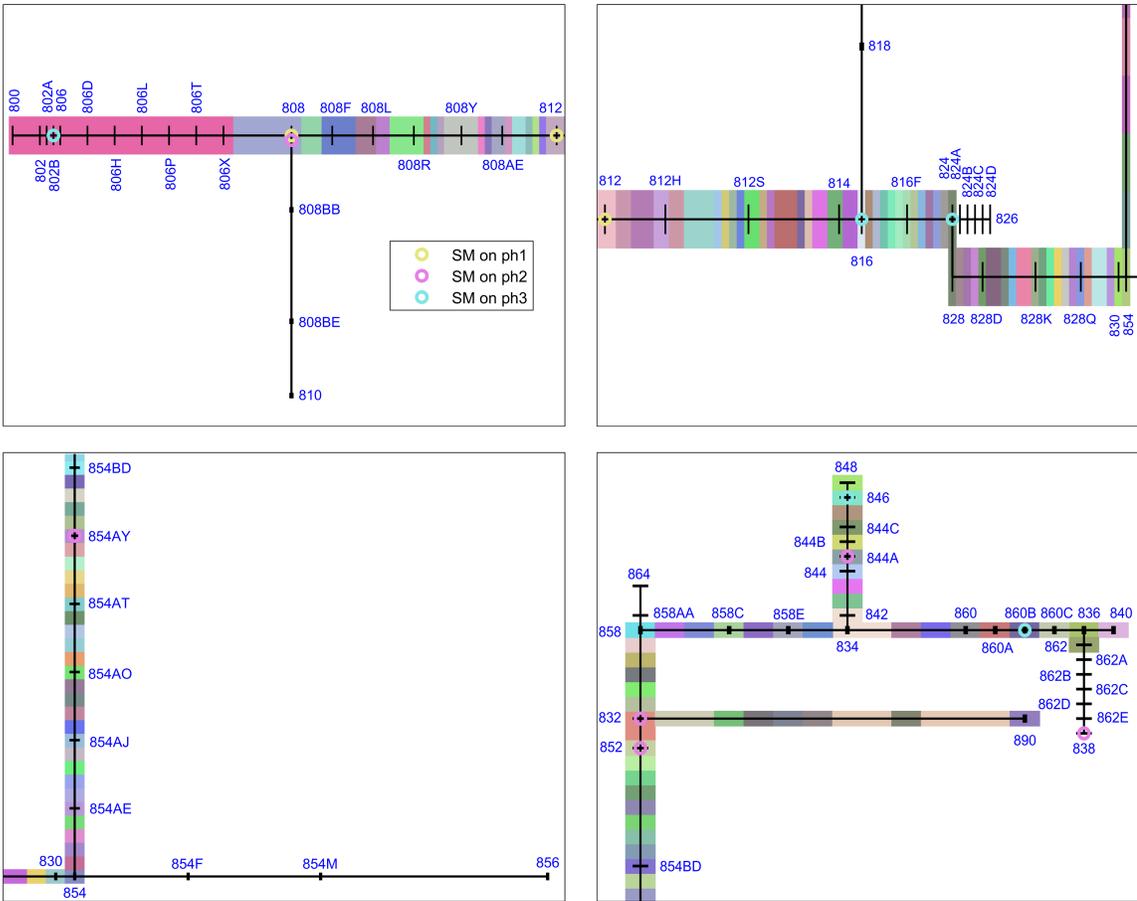


Figure 4.18: Zoning map for LLL

the smart meters. Since each possible meter location is a feature (or a predictor) of the training set, the overall optimal placement problem can be considered as a feature selection problem. However, it is not a regular feature selection; As our simulation results suggest, because of the presence of the merging process, implementing a regular feature selection method based on the minimization of misclassification ratio or other similar criteria is not adequate for the optimal selection of smart meters.

- The procedure of merging segments can be considered as a clustering problem. However, it is not just a simple clustering; Clusters (zones) must only contain segments that are physically connected, so that there is a path between two given segments in a zone.

- Due to the nature of the optimal placement problem that involves training classifiers, hyperparameter optimization and merging segments, analytical optimization methods cannot be used. Among many metaheuristic optimization methods, Simulated Annealing seems to be a suitable choice because it relies on local search and the process of generating new solutions in the neighborhood of a specific solution in a distribution system is relatively easy.

Chapter 5

Conclusion And Future Work

The smart grid is a concept aimed to address numerous inadequacies of conventional distribution systems. It is an effort to upgrade the distribution grid in order to meet modern energy needs, integrate DERs, reduce greenhouse gas emissions, improve power quality indices, and enhance protection and security strategies. To do this, the smart grid takes advantage of advanced measurement and communication technologies to improve the observability of the grid, sophisticated control and optimization approaches to enhance power quality and provide necessities for the integration of DERs, and smart protection and restoration mechanisms.

Advanced Metering Infrastructure, as a part of the smart infrastructure, provides crucial operational data about the grid. In this thesis, we show the capabilities and benefits of applying machine learning methods to process measurement data from AMI. We developed data-driven approaches that aim at enhancing the performance of management and protection systems of smart grids.

5.1 Volt-Var Optimization

In the first part of the thesis, we introduced a new data-driven Volt-Var Optimization scheme. It was assumed that all customers across the grid were equipped with AMI-enabled smart meters. Therefore, there was full observability of the grid. The data collected from smart meters was used to develop a model of the distribution system. In particular, a nonlinear model of the

system was developed using the SVR, based on AMI measurements. The VVO scheme proposed in the thesis used that regression model as a part of Model Predictive Control to optimally control the distribution grid and enforce VVO objectives. Unlike model-based approaches, our procedure does not rely on circuit-based simulations.

In the thesis, we study the effect of the size of the training dataset on the accuracy of the constructed regression model of the grid. The capabilities of the SVR-based MPC were proven via several case studies conducted on IEEE123 test feeder. It has been shown that the closed-loop nature of the predictive control is quite effective in compensating model prediction errors and helps the SVR-based MPC to deliver close to optimal results. Further studies have also revealed the effectiveness of the proposed technique in optimizing the operation of the meshed grids as well as controlling systems containing DERs.

Based on the performed experiments and analysis of the obtained results, we can envision a number of topics that should be investigated and worked on in order to fully utilize the benefits of the proposed approach.

- MPC has a flexible structure. This allows for the expansion of its objective function to include additional operational or security constraints. Additional objectives like conservative voltage reduction can be readily added to the objective function of MPC.
- The effect of the active controllers of DERs was not studied in the thesis. The nonlinear SVR is able to map those controllers. Therefore, the proposed method can be expanded in future to include such controllers as well.
- In the current implementation of our proposed method, if there is a need for retraining the model, MPC needs to stop operation in order to collect a new set of training data. This could be a topic for further research to see if it is possible to collect training data while the controller is still in operation.

- This study assumed that all loads were equipped with smart meters. There are cases in which smart meters are only installed for a portion of loads. An interesting topic for future work could be designing a method, probably a hybrid circuit-machine learning approach, that can still enhance the Volt-Var Optimization, even though AMI is not fully implemented.

5.2 Smart Meter Placement

In the second part of this research, we proposed a novel method for the optimal placement of smart meters for the purpose of locating faults in a grid. The study considers a situation in which the number of available smart meters with voltage sag measurement capability is limited. Through an optimization procedure, the proposed algorithm locates the optimal places for installing smart meters, so that the accuracy of locating faults is maximized.

The optimization is performed in two stages: random search with an NB classifier; and Simulated Annealing based search with an SVM classifier. The output of the optimization procedure is the optimal location of smart meters, a zoning map and a classification model for each types of fault, as well as a classification model for detecting the type of faults.

Simulation results of applying the optimal meter placement method to IEEE34 and extended IEEE34 test systems have demonstrated the effectiveness of the proposed method in finding the optimal placement for meters and locating fault zones.

To the best of our knowledge, this study is the first of its kind. Hence, there is a number of topics that can be investigated to improve the performance of the proposed method. The followings are some suggestions for future work on this subject.

- Accuracy of classification models: an outage area information can be added to the training process to enhance the performance of classifiers. In its current form, the proposed optimal placement method only utilizes voltage sag measurements for training classification models. Upon

availability, outage area information can narrow down the faulty area. Therefore, adding this information as an input to classification models can potentially decrease a number of misclassifications.

- Fault locating methods: the proposed method utilizes classification models developed using measurement data during the optimization procedure to locate faults. Further studies should be done to investigate if the proposed optimal placement algorithm could be beneficial to and used in conjunction with other smart meter based fault locating methods.

References

- [1] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid — the new and improved power grid: A survey,” *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 944–980, 2012. 2
- [2] M. Sugiyama, *Statistical Reinforcement Learning: Modern Machine Learning Approaches*, ser. Chapman & Hall. CRC Press, 2015. 4
- [3] T. A. Short, *Electric Power Distribution Handbook*, 2nd ed. CRC Press, 2014. 11, 13, 15, 30
- [4] “IEEE guide for smart grid interoperability of energy technology and information technology operation with the electric power system (eps), end-use applications, and loads,” *IEEE Std 2030-2011*, pp. 1–126, 2011. 16
- [5] H. Farhangi, “The path of the smart grid,” *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010. 16
- [6] (2010). Nist framework and roadmap for smart grid interoperability standards, release 1.0, [Online]. Available: https://www.nist.gov/sites/default/files/documents/public_affairs/releasessmartgrid_interoperability_final.pdf. 16
- [7] (2018). Residential electrical metering: GE grid solutions, [Online]. Available: https://www.gegridsolutions.com/products/brochures/i210_family.pdf. 17
- [8] E. Kabalci and Y. Kabalci, Eds., *Smart Grids and Their Communication Systems*. Singapor: Springer, 2019. 18
- [9] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, “A survey on advanced metering infrastructure,” *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473–484, 2014. 18
- [10] “IEEE guide for the interoperability of energy storage systems integrated with the electric power infrastructure,” *IEEE Std 2030.2-2015*, pp. 1–138, 2015. 18
- [11] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT ’92, Pittsburgh, Pennsylvania, USA: ACM, 1992, pp. 144–152. 19

- [12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. 19
- [13] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995. 20, 21
- [14] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Muller, E. Sackinger, P. Simard, and V. Vapnik, “Comparison of classifier methods: A case study in handwritten digit recognition,” in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*, vol. 2, 1994, pp. 77–82. 21
- [15] K. S., P. L., and D. G., “Single-layer learning revisited: A stepwise procedure for building and training a neural network,” in *Neurocomputing*, S. F.F. and H. J., Eds., vol. 68, Berlin: Springer, 1990. 22
- [16] J. Platt, N. Cristianini, and J. Shawe-Taylor, “Large margin dag’s for multiclass classification,” in *Proc. Advances in Neural Information Processing Systems 12*, 1999. 22
- [17] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004. 23
- [18] T. Joachims, “Making large-scale SVM learning practical,” in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds., Cambridge, MA: MIT Press, 1999, ch. 11, pp. 169–184. 24, 37
- [19] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” Tech. Rep. MSR-TR-98-14, 1998, pp. 1–21. 24
- [20] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012. 25
- [21] M. Murty and V. Devi, *Pattern Recognition: An Algorithmic Approach*, ser. Undergraduate Topics in Computer Science. Springer London, 2011. 26
- [22] G. H. John and P. Langley, “Estimating continuous distributions in bayesian classifiers,” *CoRR*, vol. abs/1302.4964, 2013. 27
- [23] H. Zhang, “The optimality of naive bayes,” in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, May 17-19, 2004 (Miami Beach, Florida, USA), V. Barr and Z. Markov, Eds., AAAI Press, 2004. 28
- [24] M. E. Baran and Ming-Yung Hsu, “Volt/var control at distribution substations,” *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 312–318, 1999. 30, 31
- [25] H. Ahmadi, J. R. Martí, and H. W. Dommel, “A framework for volt-var optimization in distribution systems,” *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1473–1483, 2015. 31

- [26] R. R. Jha, A. Dubey, C. Liu, and K. P. Schneider, “Bi-level volt-var optimization to coordinate smart inverters with voltage control devices,” *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 1801–1813, 2019. 31
- [27] A. Padilha-Feltrin, D. A. Quijano Rodezno, and J. R. S. Mantovani, “Volt-var multiobjective optimization to peak-load relief and energy efficiency in distribution networks,” *IEEE Transactions on Power Delivery*, vol. 30, no. 2, pp. 618–626, 2015. 31
- [28] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage security assessment,” *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1232–1239, 2000. 31, 42
- [29] A. Anwar, A. N. Mahmood, J. Taheri, Z. Tari, and A. Y. Zomaya, “HPC-based intelligent volt/var control of unbalanced distribution smart grid in the presence of noise,” *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1446–1459, 2017. 31, 42
- [30] J. B. Rawlings, “Tutorial overview of model predictive control,” *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000. 31, 39
- [31] Z. Wang, J. Wang, B. Chen, M. M. Begovic, and Y. He, “MPC-based voltage/var optimization for distribution circuits with distributed generators and exponential load models,” *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2412–2420, 2014. 31, 64
- [32] R. Zafar, J. Ravishankar, J. E. Fletcher, and H. R. Pota, “Multi-timescale model predictive control of battery energy storage system using conic relaxation in smart distribution grids,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7152–7161, 2018. 31, 64
- [33] P. Bagheri and W. Xu, “Model-free volt-var control based on measurement data analytics,” *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1471–1482, 2019. 31, 32, 64
- [34] J. Yu, Y. Weng, and R. Rajagopal, “Robust mapping rule estimation for power flow analysis in distribution grids,” in *2017 North American Power Symposium (NAPS)*, 2017, pp. 1–6. 31, 32, 38
- [35] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, “Data-driven power flow linearization: A regression approach,” *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2569–2580, 2019. 32
- [36] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, “Application of big data and machine learning in smart grid, and associated security concerns: A review,” *IEEE Access*, vol. 7, pp. 13 960–13 988, 2019. 32

- [37] Y. Xu, “Generalized predictive control model based on support vector machines,” in *2012 8th International Conference on Natural Computation*, 2012, pp. 84–87. 39
- [38] Y. Shi, D. Sun, Q. Wang, S. Nian, and L. Xiang, “A nonlinear model predictive control based on least squares support vector machines narx model,” in *2007 International Conference on Machine Learning and Cybernetics*, vol. 2, 2007, pp. 721–725. 39
- [39] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Short-term residential load forecasting based on resident behaviour learning,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087–1088, 2018. 41
- [40] H. Jiang, Y. Zhang, E. Muljadi, J. J. Zhang, and D. W. Gao, “A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3341–3350, 2018. 41
- [41] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948. 43
- [42] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73. 43
- [43] N. Higashi and H. Iba, “Particle swarm optimization with gaussian mutation,” in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No.03EX706)*, 2003, pp. 72–79. 43
- [44] (2019). Cuda toolkit, [Online]. Available: <https://developer.nvidia.com/cuda-zone>. 44
- [45] W. H. Kersting, “Radial distribution test feeders,” *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975–985, 1991. 50, 95
- [46] (2019). PES test feeder, [Online]. Available: <http://sites.ieee.org/pes-testfeeders/resources/>. 50, 95
- [47] (2019). SVM^{light} source code, [Online]. Available: <http://svmlight.joachims.org/>. 50
- [48] R. Torquato, Q. Shi, W. Xu, and W. Freitas, “A monte carlo simulation platform for studying low voltage residential networks,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2766–2776, 2014. 52
- [49] A. Doucet, A. Smith, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. Information Science and Statistics. Springer New York, 2001. 52
- [50] (2018). Appliance usage chart, toronto hydro electric system, [Online]. Available: <https://www.torontohydro.com/sites/electricsystem/residential/yourbilloverview/Pages/ApplianceChart.aspx>. 52

- [51] V. Vito and A. Maagoe, “Ecodesign preparatory study on smart appliances (lot 33), meerp tasks 1-6,” European Commission, Tech. Rep., Jan. 2017. 52
- [52] J. Peng, C. F. Chien, and T. L. B. Tseng, “Rough set theory for data mining for fault diagnosis on distribution feeder,” *IEE Proceedings - Generation, Transmission and Distribution*, vol. 151, no. 6, pp. 689–697, 2004. 72
- [53] Chen-Fu Chien, Shi-Lin Chen, and Yih-Shin Lin, “Using bayesian network for fault location on distribution feeder,” *IEEE Transactions on Power Delivery*, vol. 17, no. 3, pp. 785–793, 2002. 72
- [54] F. J. Muench and G. A. Wright, “Fault indicators: Types, strengths applications,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 12, pp. 3688–3693, 1984. 72
- [55] J. Teng, W. Huang, and S. Luan, “Automatic and fast faulted line-section location method for distribution systems based on fault indicators,” *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1653–1662, 2014. 72
- [56] I. Dzafic, P. Mohapatra, and H. T. Neisius, “Composite fault location for distribution management systems,” in *2010 Conference Proceedings IPEC*, 2010, pp. 795–800. 72
- [57] C. Ho, T. Lee, and C. Lin, “Optimal placement of fault indicators using the immune algorithm,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 38–45, 2011. 72, 73
- [58] W. Li, W. Chen, C. Guo, Y. Jin, and X. Gong, “Optimal placement of fault indicators in distribution system using pso algorithm,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 375–380. 72, 73
- [59] K. Kuroda, R. Yokoyama, D. Kobayashi, and T. Ichimura, “An approach to outage location prediction utilizing smart metering data,” in *2014 8th Asia Modelling Symposium*, 2014, pp. 61–66. 73
- [60] S. Kitamura, T. Takano, Y. Izui, and N. Itaya, “Disconnection detection method for power distribution lines using smart meters,” in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2015, pp. 1–5. 73
- [61] A. Ye, X. Liu, G. Zheng, and Z. Yan, “Optimization method of advanced metering infrastructure power outage site diagnosis,” in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016, pp. 1029–1033. 73
- [62] X. Yang, M. Choi, S. Lee, C. Ten, and S. Lim, “Fault location for underground power cable using distributed parameter approach,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1809–1816, 2008. 73

- [63] S. Das, N. Karnik, and S. Santoso, "Distribution fault-locating algorithms using current only," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1144–1153, 2012. 73
- [64] R. Das, M. S. Sachdev, and T. S. Sidhu, "A fault locator for radial sub-transmission and distribution lines," in *2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134)*, vol. 1, 2000, 443–448 vol. 1. 73
- [65] E. C. Senger, G. Manassero, C. Goldemberg, and E. L. Pellini, "Automated fault location system for primary distribution networks," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1332–1340, 2005. 73
- [66] M. Choi, S. Lee, S. Lim, D. Lee, and X. Yang, "A direct three-phase circuit analysis-based fault location for line-to-line fault," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2541–2547, 2007. 73
- [67] A. A. M. Zin and S. P. A. Karim, "The utilization of digital fault recorders in protection system analysis on tenaga nasional berhad transmission system," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2040–2046, 2007. 74
- [68] Z. Galijasevic and A. Abur, "Fault location using voltage measurements," *IEEE Transactions on Power Delivery*, vol. 17, no. 2, pp. 441–445, 2002. 74
- [69] R. A. F. Pereira, L. G. W. da Silva, M. Kezunovic, and J. R. S. Mantovani, "Improved fault location on distribution feeders based on matching during-fault voltage sags," *IEEE Transactions on Power Delivery*, vol. 24, no. 2, pp. 852–862, 2009. 74
- [70] Y. Dong, C. Zheng, and M. Kezunovic, "Enhancing accuracy while reducing computation complexity for voltage-sag-based distribution fault location," *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 1202–1212, 2013. 74, 76
- [71] F. C. L. Trindade, W. Freitas, and J. C. M. Vieira, "Fault location in distribution systems based on smart feeder meters," *IEEE Transactions on Power Delivery*, vol. 29, no. 1, pp. 251–260, 2014. 74
- [72] R. Pérez and C. Vásquez, "Fault location in distribution systems with distributed generation using support vector machines and smart meters," in *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, 2016, pp. 1–6. 75, 76
- [73] "Fault location in loop distribution network using svm technology," *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 254–261, 2015. 75
- [74] D. Thukaram, H. P. Khincha, and H. P. Vijaynarasimha, "Artificial neural network and support vector machine approach for locating faults in radial distribution systems," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 710–721, 2005. 75

- [75] A. Rafinia and J. Moshtagh, "A new approach to fault location in three-phase underground distribution system using combination of wavelet analysis with ann and fls," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 261–274, 2014. 75
- [76] J. J. Mora, G. Carrillo, and L. Perez, "Fault location in power distribution systems using anfis nets and current patterns," in *2006 IEEE/PES Transmission Distribution Conference and Exposition: Latin America*, 2006, pp. 1–6. 75, 76
- [77] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. 87, 122

Appendix A

Simulated Annealing

Simulated Annealing [77] is a metaheuristic optimization algorithm inspired by the annealing process in metallurgy. It is an iterative algorithm that starts with an initial solution, while at each iteration of the algorithm a neighboring solution is generated and its fitness function is evaluated. If the fitness function is better (greater in maximization problems and smaller in minimization), the current solution is replaced by the neighboring solution. One way to generate a neighboring solution for a discrete optimization space is to randomly change the value of a randomly chosen dimension.

The algorithm might accept a neighboring solution even though its fitness function is worse than the current fitness function. This allows the algorithm to explore the search space for the global optima and avoid trapping in local optima. To do so, the algorithm starts with an initial temperature $T = T_{init}$ at the first iteration, while the temperature is gradually reduced at each iteration until it reaches the final temperature $T = T_{final}$ at the last iteration. This feature of the algorithm simulates the slow energy reduction in metals during the annealing process.

At each iteration of Simulated Annealing, the neighboring solution replaces the current solution if it offers a better fitness function. If not, a random uniform number between 0 and 1 is generated. If the random number is greater than $\exp(\frac{f_c - f_{new}}{kT})$ for maximization problems, the current solution is replaced by the neighboring solution, despite having a worse fitness value. The following pseudocode describes Simulated Annealing for maximization problems:

-start with an initial solution: $s_c \leftarrow s_{init}, f_c \leftarrow f_{init}, T \leftarrow T_{init}$
-for $i = 1 : num_iter$
 -reduce temperature: $T \leftarrow T \times c, 0 < c < 1$
 -generate a neighboring solution s_{new}
 -evaluate fitness f_{new} for the neighboring solution
 -if $f_{new} \geq f_c$ then: $s_c \leftarrow s_{new}, f_c \leftarrow f_{new}$
 -elseif $rand() \geq \exp(\frac{f_c - f_{new}}{kT})$ then: $s_c \leftarrow s_{new}, f_c \leftarrow f_{new}$
-end

Appendix B

Extended IEEE34 Test System

This appendix provides details about the extension of IEEE34 test system. The extended system is called Extended IEEE34 test system. Extended IEEE34 is more realistic as it distributes distributed loads across the line they are connected to. A few extra loads are also added to the system. As a result, there are 102 loads in the system. Fig. B.1 shows a realistic (lines are displaced proportional to their actual lengths) view of this system. A single-phase 50 kVA DER is added to phase B of bus 838. The spot loads are kept the same as the original IEEE34 system. However, distributed loads are distributed according to the following tables. Besides, a few distributed loads are added to the system.

Name:	Bus	Distance from bus 802 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
802_806_1	802.b	0	PQ	7.5	3.75	14.367	1
802_806_2	802A.b	1290	PQ	7.5	3.75	14.367	1
802_806_3	802B.b	2150	PQ	7.5	3.75	14.367	1
802_806_4	806.b	2580	PQ	7.5	3.75	14.367	1
802_806_5	802.c	0	PQ	6.25	3.5	14.367	1
802_806_6	802A.c	1290	PQ	6.25	3.5	14.367	1
802_806_7	802B.c	2150	PQ	6.25	3.5	14.367	1
802_806_8	806.c	2580	PQ	6.25	3.5	14.367	1

Table B.1: Distributed loads between buses 802 and 806

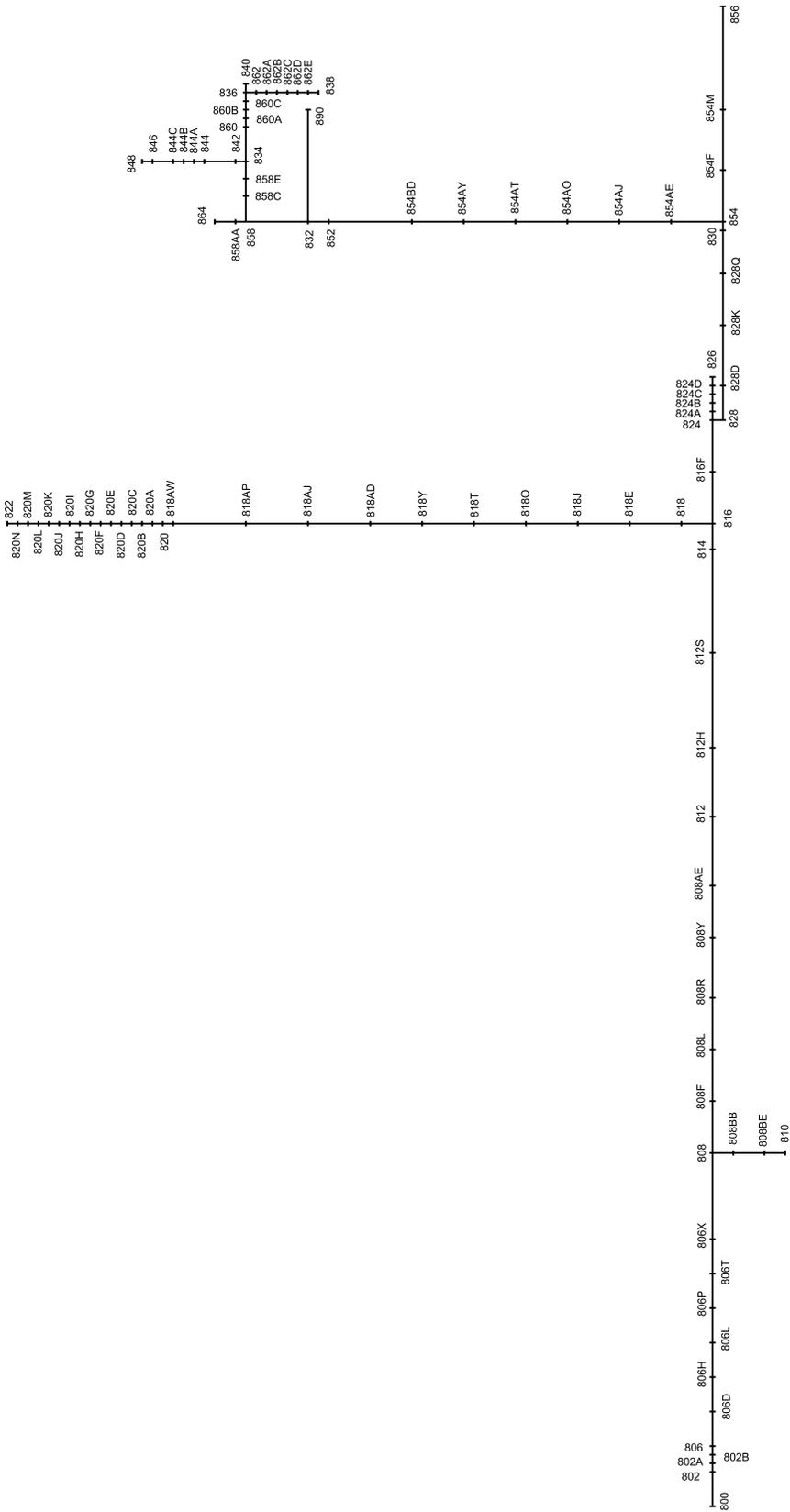


Figure B.1: Extended IEEE34 system

Name:	Bus	Distance from bus 808 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
808_810_1	808.b	0	I	4.0	2.0	14.367	1
808_810_2	808BB.b	1451	I	4.0	2.0	14.367	1
808_810_3	808BE.b	5320	I	4.0	2.0	14.367	1
808_810_4	810.b	5804	I	4.0	2.0	14.367	1

Table B.2: Distributed loads between buses 808 and 810

Name:	Bus	Distance from bus 818 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
818_820_1	818.a	0	Z	3.4	1.7	14.367	1
818_820_2	818E.a	5405	Z	3.4	1.7	14.367	1
818_820_3	818J.a	10318	Z	3.4	1.7	14.367	1
818_820_4	818O.a	15231	Z	3.4	1.7	14.367	1
818_820_5	818T.a	20144	Z	3.4	1.7	14.367	1
818_820_6	818Y.a	25058	Z	3.4	1.7	14.367	1
818_820_7	818AD.a	29971	Z	3.4	1.7	14.367	1
818_820_8	818AJ.a	35867	Z	3.4	1.7	14.367	1
818_820_9	818AP.a	41763	Z	3.4	1.7	14.367	1
818_820_10	818AW.a	48641	Z	3.4	1.7	14.367	1

Table B.3: Distributed loads between buses 818 and 820

Name:	Bus	Distance from bus 820 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
820_822_1	820.a	0	PQ	8.44	4.38	14.367	1
820_822_2	820A.a	490.7	PQ	8.44	4.38	14.367	1
820_822_3	820B.a	1472	PQ	8.44	4.38	14.367	1
820_822_4	820C.a	2454	PQ	8.44	4.38	14.367	1
820_822_5	820D.a	3435	PQ	8.44	4.38	14.367	1
820_822_6	820E.a	4416	PQ	8.44	4.38	14.367	1
820_822_7	820F.a	5398	PQ	8.44	4.38	14.367	1
820_822_8	820G.a	6379	PQ	8.44	4.38	14.367	1
820_822_9	820H.a	7361	PQ	8.44	4.38	14.367	1
820_822_10	820I.a	8342	PQ	8.44	4.38	14.367	1
820_822_11	820J.a	9324	PQ	8.44	4.38	14.367	1
820_822_12	820K.a	10305	PQ	8.44	4.38	14.367	1
820_822_13	820L.a	11286	PQ	8.44	4.38	14.367	1
820_822_14	820M.a	12268	PQ	8.44	4.38	14.367	1
820_822_15	820N.a	13249	PQ	8.44	4.38	14.367	1
820_822_16	822.a	13740	PQ	8.44	4.38	14.367	1

Table B.4: Distributed loads between buses 820 and 822

Name:	Bus	Distance from bus 816 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
816_824.1	816.b.c	0	PG	2.5	1.0	24.9	1
816_824.2	824.b.c	10210	PG	2.5	1.0	24.9	1

Table B.5: Distributed loads between buses 816 and 824

Name:	Bus	Distance from bus 824 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
824_826.1	824.b	0	I	6.67	3.33	14.367	1
824_826.2	824A.b	378.7	I	6.67	3.33	14.367	1
824_826.3	824B.b	1136	I	6.67	3.33	14.367	1
824_826.4	824C.b	1894	I	6.67	3.33	14.367	1
824_826.5	824D.b	2651	I	6.67	3.33	14.367	1
824_826.6	826.b	3030	I	6.67	3.33	14.367	1

Table B.6: Distributed loads between buses 824 and 826

Name:	Bus	Distance from bus 832 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
832_858.1	832.a	0	Z	3.5	1.5	24.9	1
832_858.2	832.b	0	Z	3.5	1.5	24.9	1
832_858.3	832.c	0	Z	3.5	1.5	24.9	1
832_858.4	858.a	4900	Z	3.5	1.5	24.9	1
832_858.5	858.b	4900	Z	3.5	1.5	24.9	1
832_858.6	858.c	4900	Z	3.5	1.5	24.9	1

Table B.7: Distributed loads between buses 832 and 858

Name:	Bus	Distance from bus 858 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
858_834.1	858.a.b	0	PQ	1.0	0.5	24.9	1
858_834.2	858.b.c	0	PQ	3.75	2.0	24.9	1
858_834.3	858.c.a	0	PQ	3.25	1.75	24.9	1
858_834.4	858C.a.b	3401	PQ	1.0	0.5	24.9	1
858_834.5	858C.b.c	3401	PQ	3.75	2.0	24.9	1
858_834.6	858C.c.a	3401	PQ	3.25	1.75	24.9	1
858_834.7	858E.a.b	5344	PQ	1.0	0.5	24.9	1
858_834.8	858E.b.c	5344	PQ	3.75	2.0	24.9	1
858_834.9	858E.c.a	5344	PQ	3.25	1.75	24.9	1
858_834.10	834.a.b	5830	PQ	1.0	0.5	24.9	1
858_834.11	834.b.c	5830	PQ	3.75	2.0	24.9	1
858_834.12	834.c.a	5830	PQ	3.25	1.75	24.9	1

Table B.8: Distributed loads between buses 858 and 834

Name:	Bus	Distance from bus 834 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
834_860_1	834.a.b	0	Z	1.6	0.8	24.9	1
834_860_2	834.b.c	0	Z	2.0	1.0	24.9	1
834_860_3	834.c.a	0	Z	11.0	5.5	24.9	1
834_860_4	834A.a.b	336.7	Z	1.6	0.8	24.9	1
834_860_5	834A.b.c	336.7	Z	2.0	1.0	24.9	1
834_860_6	834A.c.a	336.7	Z	11.0	5.5	24.9	1
834_860_7	834B.a.b	1010	Z	1.6	0.8	24.9	1
834_860_8	834B.b.c	1010	Z	2.0	1.0	24.9	1
834_860_9	834B.c.a	1010	Z	11.0	5.5	24.9	1
834_860_10	834C.a.b	1683	Z	1.6	0.8	24.9	1
834_860_11	834C.b.c	1683	Z	2.0	1.0	24.9	1
834_860_12	834C.c.a	1683	Z	11.0	5.5	24.9	1
834_860_13	860.a.b	2020	Z	1.6	0.8	24.9	1
834_860_14	860.b.c	2020	Z	2.0	1.0	24.9	1
834_860_15	860.c.a	2020	Z	11.0	5.5	24.9	1

Table B.9: Distributed loads between buses 834 and 860

Name:	Bus	Distance from bus 860 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
860_836_1	860.a.b	0	PQ	3.0	1.5	24.9	1
860_836_2	860.b.c	0	PQ	1.0	0.6	24.9	1
860_836_3	860.c.a	0	PQ	4.2	2.2	24.9	1
860_836_4	860A.a.b	446.7	PQ	3.0	1.5	24.9	1
860_836_5	860A.b.c	446.7	PQ	1.0	0.6	24.9	1
860_836_6	860A.c.a	446.7	PQ	4.2	2.2	24.9	1
860_836_7	860B.a.b	1340	PQ	3.0	1.5	24.9	1
860_836_8	860B.b.c	1340	PQ	1.0	0.6	24.9	1
860_836_9	860B.c.a	1340	PQ	4.2	2.2	24.9	1
860_836_10	860C.a.b	2233	PQ	3.0	1.5	24.9	1
860_836_11	860C.b.c	2233	PQ	1.0	0.6	24.9	1
860_836_12	860C.c.a	2233	PQ	4.2	2.2	24.9	1
860_836_13	836.a.b	2680	PQ	3.0	1.5	24.9	1
860_836_14	836.b.c	2680	PQ	1.0	0.6	24.9	1
860_836_15	836.c.a	2680	PQ	4.2	2.2	24.9	1

Table B.10: Distributed loads between buses 860 and 836

Name:	Bus	Distance from bus 836 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
836_840_1	836.a.b	0	I	9.0	4.5	24.9	1
836_840_2	836.b.c	0	I	11.0	5.5	24.9	1
836_840_3	840.a.b	280	I	9.0	4.5	24.9	1
836_840_4	840.b.c	280	I	11.0	5.5	24.9	1

Table B.11: Distributed loads between buses 836 and 840

Name:	Bus	Distance from bus 862 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
862_838_1	802.b	0	PQ	4.0	2.0	14.367	1
862_838_2	862A.b	486	PQ	4.0	2.0	14.367	1
862_838_3	862B.b	1458	PQ	4.0	2.0	14.367	1
862_838_4	862C.b	2430	PQ	4.0	2.0	14.367	1
862_838_5	862D.b	3402	PQ	4.0	2.0	14.367	1
862_838_6	862E.b	4374	PQ	4.0	2.0	14.367	1
862_838_7	838.b	4860	PQ	4.0	2.0	14.367	1

Table B.12: Distributed loads between buses 862 and 838

Name:	Bus	Distance from bus 842 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
842_844_1	842.a	0	PQ	4.5	2.5	14.367	1
842_844_2	844.a	1350	PQ	4.5	2.5	14.367	1

Table B.13: Distributed loads between buses 842 and 844

Name:	Bus	Distance from bus 846 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
846_848_1	846.b	0	PQ	11.5	5.5	14.367	1
846_848_2	848.b	530	PQ	11.5	5.5	14.367	1

Table B.14: Distributed loads between buses 846 and 848

Name:	Bus	Distance from bus 844 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
844_846_1	844.b	0	PQ	5.0	2.4	14.367	1
844_846_2	844A.b	455	PQ	5.0	2.4	14.367	1
844_846_3	844B.b	1365	PQ	5.0	2.4	14.367	1
844_846_4	844C.b	2275	PQ	5.0	2.4	14.367	1
844_846_5	846.b	3640	PQ	5.0	2.4	14.367	1
844_846_6	844.c	0	PQ	4.0	2.2	14.367	1
844_846_7	844A.c	455	PQ	4.0	2.2	14.367	1
844_846_8	844B.c	1365	PQ	4.0	2.2	14.367	1
844_846_9	844C.c	2275	PQ	4.0	2.2	14.367	1
844_846_10	846.c	3640	PQ	4.0	2.2	14.367	1

Table B.15: Distributed loads between buses 844 and 846

Name:	Bus	Distance from bus 806 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
806_808.1	806D.a	4395	PQ	4.0	2.0	14.367	1
806_808.2	806T.a	20022	PQ	4.0	2.0	14.367	1
806_808.3	806L.b	12208	PQ	4.0	2.0	14.367	1
806_808.4	806X.b	23928	PQ	4.0	2.0	14.367	1
806_808.5	806H.c	8312	PQ	4.0	2.0	14.367	1
806_808.6	806P.c	16115	PQ	4.0	2.0	14.367	1

Table B.16: Added distributed loads between buses 806 and 808

Name:	Bus	Distance from bus 808 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
808_812.1	808.a	0	PQ	4.0	2.0	14.367	1
808_812.2	808L.a	12336	PQ	4.0	2.0	14.367	1
808_812.3	808F.b	6415	PQ	4.0	2.0	14.367	1
808_812.4	808Y.b	25164	PQ	4.0	2.0	14.367	1
808_812.5	808.c	0	PQ	4.0	2.0	14.367	1
808_812.6	808R.c	18257	PQ	4.0	2.0	14.367	1
808_812.7	808AE.c	31086	PQ	4.0	2.0	14.367	1

Table B.17: Added distributed loads between buses 808 and 812

Name:	Bus	Distance from bus 812 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
812_814.1	812	0	I	12.0	8.0	24.9	3- Δ
812_814.2	812H	8423	I	12.0	8.0	24.9	3- Δ
812_814.3	812S	19325	I	12.0	8.0	24.9	3- Δ
812_814.4	814	29730	I	12.0	8.0	24.9	3- Δ

Table B.18: Added distributed loads between buses 812 and 814

Name:	Bus	Distance from bus 816 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
816_824.1	816.b.c	0	I	2.5	1.0	24.9	1
816_824.2	824.b.c	10210	I	2.5	1.0	24.9	1
816_824.3	816F.b.c	6033	I	2.5	1.0	24.9	1

Table B.19: Added distributed loads between buses 816 and 824

Name:	Bus	Distance from bus 854 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
854_852_1	854.a	0	PQ	4.0	2.0	14.376	1
854_852_2	854AE.a	5475	PQ	4.0	2.0	14.376	1
854_852_3	854AT.a	20406	PQ	4.0	2.0	14.376	1
854_852_4	852.a	36830	PQ	4.0	2.0	14.376	1
854_852_5	854.b	0	PQ	4.0	2.0	14.376	1
854_852_6	854AO.b	15429	PQ	4.0	2.0	14.376	1
854_852_7	854AY.b	25383	PQ	4.0	2.0	14.376	1
854_852_8	852.b	36830	PQ	4.0	2.0	14.376	1
854_852_9	854.c	0	PQ	4.0	2.0	14.376	1
854_852_10	854AJ.c	10452	PQ	4.0	2.0	14.376	1
854_852_11	854BD.c	30360	PQ	4.0	2.0	14.376	1
854_852_12	852.c	36830	PQ	4.0	2.0	14.376	1

Table B.20: Added distributed loads between buses 854 and 852

Name:	Bus	Distance from bus 854 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
854_856.1	854F.b	6319	PQ	2.0	1.0	14.376	1
854_856.2	854M.b	13123	PQ	2.0	1.0	14.376	1
854_856.3	856.b	23330	PQ	2.0	1.0	14.376	1

Table B.21: Added distributed loads between buses 854 and 856

Name:	Bus	Distance from bus 858 (ft)	Type	kw	kVAR	Voltage (kV)	Phases
858_864.1	858AA.a	405	PQ	1.0	0.5	14.376	1
858_864.2	864.a	1620	PQ	1.0	0.5	14.376	1

Table B.22: Added distributed loads between buses 858 and 864