AUTOMATIC ROAD NETWORK EXTRACTION FROM SATELLITE IMAGES

A THESIS

submitted by

KOLLI V S PRASADA REDDY

for the award of the degree

of

MASTER OF SCIENCE
(by Research)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS

SEPTEMBER 2000

Certificate

This is to certify that the thesis entitled **Computational Intelligent Techniques in Power System Security Analysis** submitted by **D. Devaraj** to the Indian Institute of Technology, Madras for the award of the degree of Doctor of Philosophy, is a bonafide record of research work carried out by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

(Prof. K. Ramar)

Place: Chennai-600036 (Prof. B. Yegnanarayana)

Date:

ACKNOWLEDGEMENTS

First and foremost I would like to express my sincere thanks to my guides Prof. B. Yegnanarayana and Prof. K. Ramar for their guidance, help and encouragement throughout this work. I am fortunate to work under two great researchers. I thank my doctoral committee members for their time and effort in reviewing the progress in this work. I would like to thank Dr S.Mohan and Dr. Swaroop for their helpful suggestions during the DC meetings.

I express my gratitude to Prof. B. Yegnanarayana and Prof. C. Venkaseshaiah for their elderly advice and timely help during the critical periods. Special thanks are due to Dr. Manish Sarkar for training me with good seeds during the initial period of this work. I am highly indebted to my co-research scholar Thilagar for his encouragement and for the technical and non-technical discussions which we had for hours.

I thank the Head of the Department of Computer Science and Engg., and Dr. Sukhendu Das for permitting me to work in the ANN lab. I thank Mr. Rajendran for making the systems and the Matlab software available in the hour of need.

I am thankful to P. Kiran and Prasad for their help at various times in preparing the documents in LaTeX. I thank my colleagues Prasanna, Surya, Nayeem, Srinivasa rao, Palanivel, Sharat, Kishore, Kiran, Vinod, Anil, Venkatesh, Jinu, Gupta and Viswam for their help and co-operation. My thanks are due to Sharat and Prasad who had given me a very good company in SAC, without which my stay here would not have been so much enjoyable.

I would like to thank my wife Viji for her patience and cooperation she extended during these four years. I thank my colleagues and old students of A.K. College of Engg., Krishnankoil, for their well wishes. Above all I thank my parents for their prayers and blessings, without which I would not have completed this work.

ABSTRACT

Keywords: Power system security, Contingency selection, Feature selection, Mutual information, Neural networks, Radial basis function network, Security enhancement, Genetic algorithm.

Due to the economical and environmental constraints, power systems all over the world are being forced to operate near to their capacity limits. Under this stressed mode of operation, a disturbance in the form of equipment outage may lead to cascading failures, and hence to system blackout. To mitigate these undesirable events and to maintain continuous supply of power to the consumers without disruption, real time monitoring of security of the system against probable disturbances is essential. This thesis identifies issues in security assessment of a power system and explores the use of computational intelligence tools like Artificial Neural Networks (ANN) and Genetic Algorithm (GA) in static-security analysis.

A common criterion for operational security is the ability to withstand the outage of any one piece of equipment, which is called "N-1 security". The security of the system is assessed by modeling the outage events or contingencies using a power flow program by simulating the contingencies one at a time. But in real-time environment it is impractical to analyze the effect of every conceivable contingency in detail. A systematic procedure is desired to identify automatically the critical contingencies to be studied in detail. This screening phase is called **contingency selection**. This thesis proposes ANN-based models for automatic contingency selection process. The ANN models are trained off-line using data collected from a number of contingency analysis studies. These trained networks can then be used to estimate the security level corresponding to different operating conditions.

Scaling ANN models to realistic dimensions needed for power system problems

is a major issue. Real power systems have thousands of inputs at the system level. If all the available measurement variables are used as inputs to an ANN, it results in increased size of network and larger training time. To make the ANN models useful for large scale power system problems, some dimension reduction is mandatory. In this work, **mutual information** between input variables and output is used to select the input features of the network.

Once the important contingencies have been identified, remedial actions should be taken to prevent the limit violations in the system, should any one of those contingencies occur. The common strategy followed is through rescheduling of power system controls in the pre-contingency state itself based on the result of security-constrained optimal power flow (OPF) program or by preparing a contingency plan in advance using optimal power flow program and executing it as soon as the contingency occurs. This thesis proposes a **GA-based technique** to solve the optimal power flow problem for preparing the contingency plan.

TABLE OF CONTENTS

ABBREVIATIONS and LIST OF SYMBOLS

ANN : Artificial Neural Network
$\mathbf{a}$: Vector of input variables
B_{ii} : Self-susceptance of bus i (pu)
B_{ij} : Mutual susceptance between buses i and j (pu)
D : Dimension of input variables
$\mathbf{d}$: Vector of desired output
GA : Genetic Algorithm
G_{ii} : Self-conductance of bus i (pu)
G_{ij} : Mutual conductance between buses i and j (pu)
H : Entropy
I : Mutual information
J_{MW} : MW performance index
K : Number of clusters
M : Dimension of selected features
N : Number of training patterns
N_b : Number of buses in the system
N_c : Number of contingencies in the contingency list
N_g : Number of generators
N_l : Number of lines in the system
OPF : Optimal Power Flow
P_i : Reactive powers injected into network at bus i (pu)
P_{di} : Real power demand at bus i
P_{gi} : Real power generated at bus i
P_l : Power flow in line l
P_l^{max} : Rating of line l in MW
Q_i : Reactive powers injected into network at bus i (pu)
Q_{di} : Reactive power demand at bus i
Q_{gi} : Reactive power generated at bus i
RBF : Radial Basis Function
s_k : Security index corresponding to contingency k
S_l^{post} : Post-contingency MVA flow in line l
S_l^{max} : Rating of line l in MVA
$\mathbf{u}$: Vector control variables
V_i : Voltage magnitude at bus i (pu)
X,Y : Random variables
$\mathbf{x}$: Vector of state variables
θ_{ij} : Voltage angle difference between buses i and j (rad)
ϕ_j : Activation function of the jth hidden node
$\boldsymbol{\mu_j}$: Vector of jth basis function center
σ_j : Width of jth basis function

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The primary aim of a power system is to provide uninterrupted supply of electricity to the consumers within the set limits of voltage and frequency. Power systems have been designed and built to operate efficiently under normal circumstances. However, disturbances in the form of equipment outages may occur in the system at any time during normal operation. These outages, called "contingencies", may occur, either due to internal causes or due to external causes such as lightning strike, thunder storm, or objects hitting transmission towers. These disturbances will have widespread impact on the operation of the system, leading sometimes to "blackout" of the system [39]. In order to maintain uninterrupted supply of power to the consumers, it is mandatory to study the impact of contingencies on the operation of a system, and to take remedial measures, if needed. These tasks need to be carried out on real-time basis through the power system control centers, which are equipped with real-time computers and high-speed data acquisition systems. Numerous papers and articles have been written on this topic of **security analysis** and a good overview of this topic can be found in the review articles

[9,117]. In this chapter, the basic concept of static-security assessment is explained and the main issues in real-time security analysis are highlighted.

1.1 Power System Under Steady-state

The steady-state operation of a power system is modeled by a set of equality constraints (called load flow equations) representing the power balance at each bus. The load flow equations are represented by the following expressions:

$$P_i - V_i \sum_{j=1}^{N_b} V_j \left[G_{ij} cos\theta_{ij} + B_{ij} sin\theta_{ij}\right] = 0,\ i = 1, 2, \cdots, N_b \tag{1.1}$$

$$Q_i - V_i \sum_{j=1}^{N_b} V_j \left[G_{ij} sin\theta_{ij} - B_{ij} cos\theta_{ij}\right] = 0,\ i = 1, 2, \cdots, N_b \tag{1.2}$$

where

N_b	number of buses in the system
P_i, Q_i	real and reactive powers injected into network at bus i (pu)
V_i	voltage magnitude at bus i (pu)
θ_{ij}	voltage angle difference between buses i and j (rad)
G_{ii}, B_{ii}	self-conductance and susceptance of bus i (pu)
G_{ij}, B_{ij}	mutual conductance and susceptance between buses i and j (pu)

For a derivation of the load flow equations refer Grainger and Stevension [46]. The load flow equations can be solved to determine the power flows and voltage levels throughout the transmission system. The common solution method used in the power industry is based on the Newton-Raphson algorithm [58, 114] or its derivatives.

Satisfying the load flow equations (1.1) and (1.2) with any set of numerical variables means that the physical characteristics of the power system under certain conditions are satisfied. But, many of these physically possible states do not make operational sense or are not operationally possible. In order to model the power system behavior more realistically, the operational constraints have to be included. The various operational constraints include capacity limits on generators, lower and upper bounds on bus voltages, thermal limits of transmission lines etc.

In summary, the operation of the power system can be expressed in the form of two sets of constraints:

$$\mathbf{g}(\mathbf{x}, \mathbf{u}) = 0 \qquad \text{(load constraints)} \tag{1.3}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0 \qquad \text{(operating constraints)} \tag{1.4}$$

where

$\mathbf{g}$, $\mathbf{h}$	function vectors
$\mathbf{x}$	vector of dependent variables
$\mathbf{u}$	vector of independent or control variables

During normal operation, all the equality and inequality constraints are satisfied, indicating that the generation is adequate to meet the existing demand of the load, and that no equipment is being overloaded. This state of the power system is called *operational-constraint state* [33]. An *unplanned outage* like loss of a transmission line or a generator causes changes in the flows and voltages on the remaining transmission equipment connected to the system and may leads to overload of an equipment. This if severe enough, can damage transmission and generation equipment of its own. However, all equipment in a power system is protected against a fault with protective-relays. Therefore, an overload which persists long enough on any equipment usually results in its being switched out of service. However, the outage of second piece of equipment due to relay action often results in yet more readjustment of power flows and bus voltages that can in turn cause more overloads and cause further removal of equipment, etc. This can cause an uncontrollable cascading series of overloads and equipment removal resulting in shutting down a large part of the system.

As transmission networks are open to the atmosphere, it is always under the threat of being affected by disturbances like lightning stroke, wind storm, ice etc. Though power system is designed for good tolerance, its response to any particular component failure will vary depending on system conditions. To maintain the supply of electrical energy to the consumers under disturbance requires evaluating the effect of contingencies on the present operating state using computer simulation and taking remedial measures, if needed.

1.2 Steady-state Security Assessment

Power system security is concerned with determining whether the system operating in a normal state can withstand the contingencies without violating the load constraints and the operating limit constraints or not. A power system operating state is considered to be *secure* if there is neither any occurrence of overloading of any equipment nor any instability in the system due to a set of *credible contingencies* (applied one at a time) [34]. The contingency list usually consists of unplanned outages of transmission equipment and generators. The time period of consideration is such that the fast-acting automatic-control devices like governor action and automatic voltage regulators have restored the system load balance, but the slow-acting controls and human operators have not responded.

The steady-state-security of a power system under the kth contingency is determined by solving for $\mathbf{x}^k$ in the load flow equations

$$\mathbf{g}^k(\mathbf{x}^k, \mathbf{u}^k) = 0 \tag{1.5}$$

where

$\mathbf{x}^k$	vector of dependent variables under contingency k
$\mathbf{u}^k$	vector of independent or control variables under contingency k

by a power flow algorithm by taking into account the changes in the system topology. The post-contingency control vector $\mathbf{u}^k$ is calculated based on the type of contingency. For a sizeable disruption of real power, such as the loss of a tie-line or a generator, the outputs of the remaining generator are adjusted on the basis of their individual speed-droop characteristics. Otherwise, only the swing bus absorbs the slack generation.

The line flows and bus voltages are then checked against their safe operating limits. Depending on whether the post-contingency operating condition satisfies or violates any one of the operating limits, the corresponding pre-contingency power system state is appropriately labelled secure or insecure.

A quick security assessment of power system operating state using real-time data will make possible timely detection of conditions which may lead to overloading of equipment before they actually occur. This will enable the system operators to determine and take proper corrective actions in case the system is operating in an insecure state.

1.3 Real-time Security Analysis

Power system operating state is never static as the system load demand is always changing. Thus some kind of security assessment must be undertaken from time to time, and corrective actions taken in case the system is in insecure state.

Fig 1.1 shows the structure of on-line security analysis system used in the power system control center [9]. The security analysis starts with real-time measurements of line power flows, line current flows, power injections and bus voltage magnitudes, as well as the status of the circuit breakers and switches. If the present operating state is found to be violating any of the limits, corrective action via emergency control may be executed to bring the system back to the operational-constrained state, whereas, if the system is found to be in the operational-constrained state, security analysis function is invoked for the next state contingencies. The main issues in on-line security analysis are:

Contingency selection : A contingency screening mechanism is used to quickly identify severe contingency cases from the list of contingencies. This is desirable as it is impractical and unnecessary to perform detailed analysis on the impact of every conceivable contingency. Speed and accuracy are main requirements of a contingency selection mechanism.

Contingency evaluation : After identifying the critical contingencies, contingency evaluation is performed using AC power flow studies on individual cases in the order of decreasing severity. The process is continued until no post-contingency violations are encountered, or a maximum number of cases has been covered, or until a specified time has elapsed. Evaluation of a case typically involves a conventional fast-decoupled power flow solution [115].

Security enhancement : Action must be initiated to ensure sufficient security margin, if the power system is found to have insufficient margin for one or more of the selected contingencies. This can be addressed by: (a) altering the pre-contingency system operating state to mitigate or eliminate the emergency resulting from the contingency, or (b) developing a control strategy that will alleviate the emergency, should it occurs. Optimal power flow program is the main tool used for security enhancement. Security enhancement tools help the utilities achieve a more secure and cost-effective operation.

Existing contingency selection methods suffer from either misclassification or slow execution. This is a computational burden for security analysis. Conventional optimization tools used for security enhancement may lead to a local minimum, and sometimes result in divergence in solving contingency-constrained optimal power flow problem. Hence there is need to explore other alternative approaches like machine learning and derivative-free optimization techniques [20, 124] to address the contingency selection and security enhancement tasks in real-time security analysis. In this context it is worth exploring the potential of computational intelligent tools [53, 55] for the security analysis of power systems.

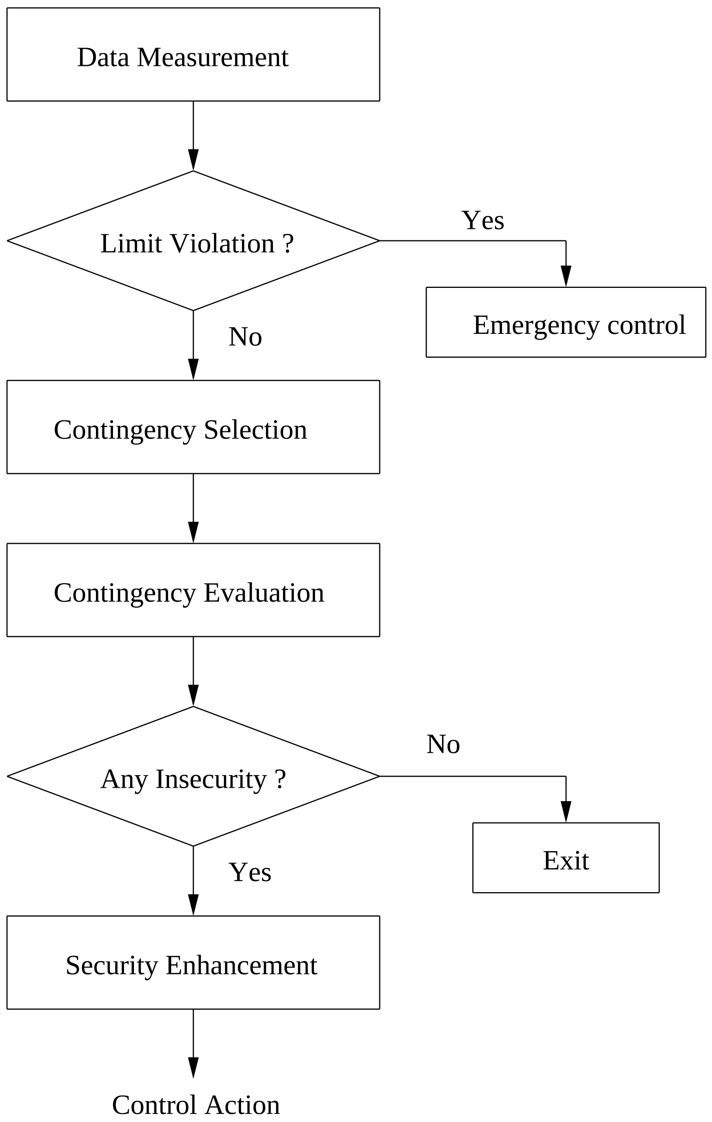

Fig. 1.1: Components of on-line security analysis.

1.4 Computational Intelligent Tools

Computational intelligence techniques explore the potential for creating intelligent systems by modeling the computational processes of biological organism. The constituents of this technique include: Fuzzy logic, neural networks and evolutionary computation such as genetic algorithms. Each of the constituents of computational intelligence offers uniquely useful means for problem solving. This section briefly review the basic concepts of neural networks and genetic algorithms, their potential applications to power system security analysis are outlined in the next section.

1.4.1 Artificial Neural Networks

An Artificial neural network is defined as a computing system made up of a number of simple, interconnected processing elements called *neurons*, which process information in parallel [47,132]. The neuron receives information through a number of input nodes, process it internally, and produces a response. The processing is usually done in two stages: First, the input values are linearly combined, then the result is used as the argument of a nonlinear *activation function*. The combination uses the *weights* attributed to each connection, and a constant *bias* term, represented by the weight of a connection with a fixed input equal to -1.

Neural networks have been found to be effective systems for learning functional mappings from a body of examples. This is done by adjusting connection weights and node threshold of the interconnected neurons according to a specific learning rule. If the network has properly learned the relationship it can perform *generalization*. Generalization is the ability of the network to respond correctly using its past experience to an input which it has never seen before.

The large amount of data typically available from the power industry, coupled with the ability of the neural network to learn the nonlinear relationships, suggest it as a viable candidate for solving significant power system engineering problems.

1.4.2 Genetic Algorithms

Natural evolution is an optimization process that can be simulated on a computer and used to good engineering purpose. Genetic algorithm emulates biological evolution to solve optimization problems [45]. Unlike traditional *hill-climbing* methods involving iterative changes to a single solution, genetic algorithms work with a population of solutions. A *fitness* value, derived from the problem's objective function is assigned to each member of the population. Individuals that represent better solutions are awarded higher fitness values, thus enabling them to survive more generations. Successive 'generations' of populations are created by the genetic operators-reproduction, crossover and mutation to yield fitter solutions which approach the optimal solution to the problem. Genetic algorithms are best suited to solve non-linear optimization problems and problems with large search space. The working principle of a simple genetic algorithm is given in Appendix D.

1.5 Proposed Approach for Security Analysis

The objective of this research work is to explore methods based on computational intelligent techniques for fast contingency ranking and to develop an operator assistant tool for security enhancement.

The problem of contingency selection is formulated as a supervised learning problem, and artificial neural network-based models are developed using the data generated through off-line simulation process. The trained networks are used to get the security information during real-time operation. Neural network-based models are proposed due to the following reasons:

1. Neural networks can learn the relationship between pre-contingency operating condition and post-contingency system status just by analyzing different operating conditions and their corresponding security levels.

2. Neural networks are extremely fast when in testing mode. This can be a major asset in real-time applications, especially when the direct applications of analytical tools is time consuming.

Rescheduling the power system controls to improve security can increase operating costs, increase duty on equipment, and can burden the operator with additional tasks. It is desirable to find the control actions that represent the optimal balance between security, economy, and other operational considerations. A genetic algorithm is proposed in this work for this task. The advantages of genetic algorithm over traditional techniques are:

(a) It places no restrictions such as differentiability and convexity on the objective function.

(b) It is less likely to lead to local minima.

1.6 Organization of the thesis

Chapter 2 review the various methods proposed in the literature for automatic contingency selection and static-security enhancement.

Chapter 3 deals with the problem of input-output parameter selection and training data generation through off-line contingency analysis.

Chapter 4 deals with the issue of feature selection in neural network-based contingency selection models. Mutual information between the input variables and the output quantity is used to address this feature selection problem.

Chapter 5 deals with the development and evaluation of radial basis function neural networks for contingency selection. Simulation results are presented for contingency ranking and selection in IEEE 30-bus system.

Chapter 6 formulates the feature selection problem as an optimization problem and genetic algorithm in combination with radial basis function network is used to select the input features of the network as well as the structure of the radial basis function network. This integrated approach helps to automate some of the processes in network development.

Chapter 7 presents the development of a genetic algorithm-based tool to aid the power system operators in taking corrective action to alleviate line overloads arising out of contingencies.

Chapter 8 concludes the thesis by summarizing the work and indicating the further directions in applying the computational intelligent tools for security analysis.

CHAPTER 2

Power System Security Analysis - A Review

2.1 Introduction

Contingency selection and security enhancement are the two major tasks in real-time security analysis. Contingency selection helps to speed up the process of security assessment by removing the harmless contingencies from the contingency list. During the past several years a great deal of research has been devoted to develop a method for automatic contingency selection to be used in real time security analysis. Methods proposed in the literature for automatic contingency selection are reviewed in this chapter. After reviewing the existing techniques for contingency selection, the proposed approach for contingency selection is outlined.

Once the potentially harmful contingencies are identified, action must be taken to avoid limit violation in the post-contingency period, should the contingency occur. Optimal power flow program provides recommendations to the power system operator to enhance the security level of the system. Techniques proposed in the literature to solve the optimal power flow problem are reviewed. Finally, a brief outline of the various issues addressed in this thesis in developing an operator assistant tool for security enhancement is presented.

2.2 Automatic Contingency Selection Problem

The analysis of contingencies for steady-state security assessment requires methods to predict the flows and voltages in the post-contingent period. A straightforward approach for contingency analysis in static-security assessment would involve modifying the base case admittance matrix to simulate the contingency, performing a power flow solution, and checking for violations in the calculated post-contingency state. However for large systems, even if only single component contingencies were considered and fast solution methods were used, such an exhaustive approach could be prohibitively time consuming.

One way to speed up the process of security assessment is to prepare a list of contingencies which are likely to give trouble to the system, and to conduct the load flow study on these selected cases alone. Historically, the contingency list to be evaluated by the security analysis program was created by the power system operators based on their experience and off-line simulation studies, and they can get along by running load flow for these cases alone. However, in real time, as the system conditions change, the contingencies which cause insecure operation may also change, and would be different from those predicted by off-line simulation studies. Another way to use a reduced list of cases to be studied is to have a calculation that would indicate the possible bad cases and run it as often as the contingency analysis is run. Thus the list of cases is dynamically built and the cases that are included in the list may change as conditions on the power system change. This process is called **Automatic Contingency Selection** (ACS).

2.3 Contingency Selection Methods

Contingency selection methods utilize a fast, approximate method to select the few contingencies that actually affect the security of the power system. This section review some of the existing contingency selection methods in the literature for static-security analysis.

2.3.1 Algorithmic contingency selection methods

The two main approaches to contingency selection are (a) Indirect methods and (b) Direct methods.

2.3.1.1 Indirect Methods

Indirect methods rely on explicit calculation of severity indices without evaluating the individual quantities. The main advantage of the indirect method is the speed at which the indices are calculated.

The first attempt in developing an indirect method for automatic contingency selection came in the form of sensitivity based ranking method [35]. The following scalar performance index (J_{MW}) was proposed to quantify the system stress in terms of line overloads [35]:

$$J_{MW} = \sum_{l=1}^{N_l} \left[\frac{P_l}{P_l^{max}} \right]^{2m} \tag{2.1}$$

where	P_l	power flow in line l
	P_l^{max}	rating of line l in MW
	N_l	no of lines in the system
	m	an integer exponent

The performance index is a scalar function of the network variables and describe some characteristics of the network operating point. Following a contingency, any line which is overloaded will make a contribution of greater than unity to the performance index, whereas a line whose flow is below its limit will make a less than unity contribution. The exponential normalized flows further increase the contributions of overloaded lines. Consequently, the impact of each contingency on the system is directly measured by the post-contingency performance index. Thus the performance index serves as a penalty function for limit violations.

The authors of [35] proposed a method for ranking transmission line contingencies by evaluating the sensitivities of the performance index with respect to outages. The sensitivities are the first derivative of the performance index with respect to a change in admittance for a line-outage or with respect to a change in net power output for a generator outage. The authors showed that through the use of Tellegen's theorem the first-order sensitivities of the performance index can be readily determined and efficiently implemented. The sensitivities can then be used to rank contingencies from the most important (largest positive performance index change) to the least important (largest negative performance index change). Later, Irisarri *et al.* [52] developed the first-order and second-order sensitivities of the performance index using elementary differential calculus rather than using

Tellegen's theorem.

The main disadvantage of sensitivity-based ranking method is that, efficient techniques for evaluating the sensitivities of performance index have been found for an exponent of 2 only (i.e., $m = 1$). This formulation causes the so called "masking effect" where, for example, many lines at moderate loading give the same performance index value as one line with an overload and others at light load (See discussion to [35]). This masking problem can be somewhat reduced by applying the performance index separately to different subnetworks of the power system, but with correspondingly reduced computing speed advantage.

The future prospects for the indirect approach do not seem promising, unless breakthroughs occur that enable evaluation of sensitivities of performance index with higher values of exponent to be used while preserving computational efficiency [117].

2.3.1.2 Direct Methods

Direct methods calculate the appropriate severity indices using the individual monitored quantities (line flows, bus voltages, reactive generation). This implies that these quantities have to be first calculated. The knowledge of the individually monitored quantities permits the calculation of severity indices of any desired complexity without significantly affecting the numerical performance of the direct methods.

Distribution factors method: This method calculates the fractional distribution of the power flowing in the outage equipment onto all other transmission lines as constant factors dependent only on the network impedances [128]. The effect of any contingency can then be quickly calculated by using these distribution factors. This technique is fast but has the following limitations: (a) Only branch MW (real power) flows are calculated and there is no knowledge of MVAR (reactive power) flows or bus voltage magnitudes. (b) The calculated values are only within about 5% accuracy.

1P-1Q method: Albuyeh *et.al* [2] proposed one iteration of the fast decoupled power flow to calculate the performance index for every outage. Outages are ranked according to the magnitudes of their corresponding performance indices. To avoid the masking problem the summation in Equation 2.1 is carried out on overloaded lines only. This 1P-1Q method is very fast in calculating the performance index values, but there is danger in using a non-iterative approach, as some of the overloads may not be apparent after only a single iteration.

Concentric relaxation method: In many instances, if a contingency causes a branch overload, the location of the overload is adjacent to the contingency. This localized nature of the outage is exploited in the concentric relaxation method proposed by Zaborszky *et.al* [134] to obtain an approximate load flow solution. The power system is divided into two parts: the affected part and the unaffected part. To make this division, the buses at the end of the outage line are marked as tier zero. The buses that are one transmission line from tier zero are labelled tier one and so on. This process is continued until all buses in the network are included. Starting with the first tier buses around the contingency, the network equations are solved by a modified Gauss-Seidal iterative technique, assuming fixed voltages at the other buses. The flows into the second tier buses are then computed and compared against their pre-contingency values. If the change is significant, the second tier buses will be included, and the above process repeated. The relaxation is continued by successively bringing in more buses tier by tier, until a point is reached where the changes in the boundary flows are no longer significant. The method suffered from the following weaknesses:

(a) Lack of consistent criteria for the selection of buses to be included in the small network.

(b) Need to solve a number of different systems of increasing size resulting from concentric expansion of the small network.

A simplified version of concentric relaxation employs a fixed number of tiers around the contingency [62] as the best compromise between speed and accuracy, but loses the inherent adaptive advantages of the method.

Bound estimate algorithm [43]: This method is based on the computation of the bound estimates of the effect of a contingency on line MW flows. The method does this through the application of various filters which are used sequentially. Each filter divides the input contingencies into three groups: Critical, non-critical and uncertain. The uncertain contingencies from one filter are used as input to the next filter, the final filter being the DC load flow. The feasibility of this approach was demonstrated on small problems [43] and consequently, it is difficult to establish its speed for large size problems. Another limitation of this method is that it addresses only the real power flow issue during contingencies.

Efficient bounding methods: Brandwajn [13, 14] proposed an efficient bounding method for contingency analysis. For a contingency involving the outage of a line $p-q$, the power system is divided into three subsystems as follows:

N1 – the subsystem immediately surrounding the outage line

N2 – the external subsystem

N3 – the set of boundary buses that separate N1 and N2

The subsystems appear as shown in Fig 2.1.

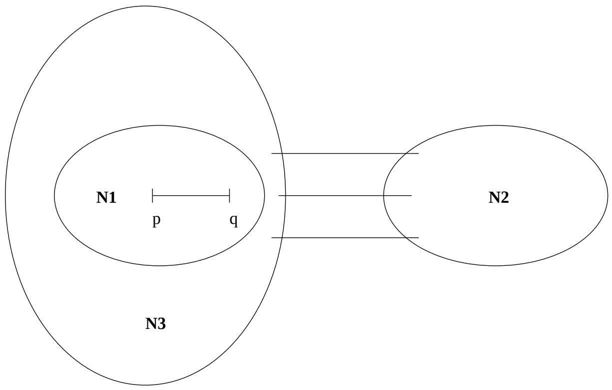

Fig. 2.1: Partition of network into subsystems

The bounding method is based on the fact that we can make certain assumptions about the line flow changes in lines in N2, given the injections in N1 and the maximum phase angle appearing across any two buses in N3. For each contingency, the algorithm automatically selects a boundary to establish a small subnetwork N1 that includes all topological changes caused by the contingency. The subnetwork N1 is expanded until it encompasses all possible overloaded circuits. Finally, only subnetwork N1 is solved and checked for limit violation.

All the algorithmic contingency selection methods mentioned above have some positive properties but they also have their specific flaws which drastically reduce the effectiveness of these methods. Efforts made to improve the performance (in terms of speed and accuracy) of the existing algorithmic methods have not produced any promising results. The following subsections summarizes non-algorithmic methods for contingency selection.

2.3.2 Artificial intelligence techniques in contingency selection

With the advancement in artificial intelligence technology, expert systems have been proposed for contingency selection [97, 104]. Most of the actions an operator in a power system control center takes in daily operation are based on qualitative understanding. This knowledge of the human operator is utilized in the form of rules in addition to the algorithmic methods for the contingency selection process in an expert system. The method provides powerful inference and explanation capabilities. However, knowledge acquisition, validation and maintenance of the expert system are quite difficult. Besides, the expert system might take time to search through the large knowledge base to get the conclusion, and thus sometime unable to meet on-line application requirement.

2.3.3 Pattern Recognition Approach for Security Assessment

Pattern recognition methods were applied to overcome the computational burden associated with the conventional security assessment methods [85,86,103]. Pattern recognition methods shift the burden of elaborate on-line calculations to off-line analysis. A general pattern recognition scheme consists of two stages: Training and testing. The training stage consists of an off-line learning process for the pattern classifier design. In the second stage, this pattern classifier is used for classification of new patterns. The basic concepts of applying pattern recognition method to power system security assessment is briefly given below.

The operating state of a power system is characterized by the variables such as power injections, loads, voltage magnitudes, phase angles at all the buses, line flows, etc. These variables can be obtained from the measured data through state estimation or power flow solutions. In pattern recognition terminology, a power system operating state can be termed as a *pattern*, and the variables characterizing the state constitute the state vector in the n-dimensional Euclidean space R^n. Thus each operating state of the power system can be represented as a point in R^n.

The pattern vector whose components represent the significant variables of the system is evaluated at many representative operating conditions, some of which are secure and others not. This set of class labeled patterns is called the training set. If a hyperplane is generated which separates the R^n such that all secure patterns fall on one side and all insecure patterns on the other side of the hyperplane, then this hyperplane can be considered as a decision surface or a security classifier. For generating an ideal security classifier all possible power system operating states should be considered. This is not practicable. Instead a large set of operating states, covering the whole spectrum of power system operating states, is considered. With the help of an off-line contingency evaluation method all the patterns are labelled as secure or insecure.

With the conventional pattern recognition technique, the computational effort required to design the classifier becomes prohibitively large, and the mapping to be learned also becomes increasingly complicated with increasing size of the power system.

2.3.4 Artificial Neural Network-based Approach

Artificial Neural Network (ANN) as a means of pattern recognition have recently attracted much attention among the researchers [12, 88, 93, 131]. Generally ANN approaches are proposed for the processes in which there exist no well-defined mathematical models which describe their behavior, making them unsuitable for application of analytical methods. But in the case of security assessment, there exists a set of nonlinear algebraic equations describing formally the behavior of the power system and the main motivation for the use of neural networks is to reduce the computational burden associated in determining the security information in real-time.

ANN models with different topologies perform different pattern recognition tasks [132]. The pattern mapping capability of neural network is exploited for contingency selection. The networks are trained to capture the relationship between pre-contingency operating condition and the post-contingency security status of the system. Most of the papers on ANN-based contingency selection have used multilayer feedforward neural networks trained by backpropagation algorithm to learn the input-output relationship. They can be classified into two groups, according to the number of networks used for contingency selection. In the first group are the ones that use only one network to handle all the contingencies in the network [70, 89, 102]. While developing the models for large power systems, in which the number of contingencies to be considered is large and the number of quantities to be monitored is very high, a monolithic network, also known as *all-contingency-one network*, either may not converge or may take large amount of time to converge during training [93, 123]. Further, with limited data set the generalization capability of the network will deteriorate due to the "curse of dimensionality" problem [12]. These difficulties prevent the application of this approach for large power system applications.

One way to avoid the scaling problem associated with the above approach, is to develop separate networks for each contingency. The advantages of this *one-contingency-one-network* approach are as follows:

(a) The architecture of the networks will be less complex, and hence, is easier and faster for training. The generalization capability may also be enhanced due to the reduction in complexity.

(b) Different feature sets can be used to train different networks. This flexibility allows us to exploit the localized nature of contingencies in the form of input dimension reduction. Therefore, the training time decreases and generalization ability may increase. Also, by using the reduction of input data, there is significant reduction in the demand for measurements within the power systems.

Following the concept of one-contingency-one network, Weerasooriya *et.al* [123] proposed a neural network classifier to predict voltage violations following single line-outages. Real and reactive power injections at all buses and the voltage magnitude at all buses were selected as possible inputs to the ANN. The input patterns of the networks were generated by conducting optimal power flow study for a large number of randomly generated load patterns. The generated patterns are labelled as secure or insecure based on the result of the contingency analysis. The key features to train the networks were selected based on the interclass distance measure proposed in [86] for feature selection in statistical classifiers. The authors have demonstrated the suitability of this approach for voltage security assessment in AEP 8-bus system. The classifier determines whether the presented input pattern is secure or insecure only, and the ranking of the contingencies based on the severity is not possible. Furthermore, the interclass distance measure used for feature selection, assumes Gaussianity in the input domain. If a variable has normal or Gaussian distribution, its distribution is completely characterized by its mean and variance. If this assumption is not true, serious errors might occur in feature selection.

Ghosh and Chowdhury [44] designed a multilayer feedforward network for line flow contingency ranking in IEEE 118-bus system. To generate the training data, the total load on the system was increased uniformly upto 300 MW from the base case value, in steps of 25 MW. For each load pattern, a training set was prepared by running an optimal power flow with contingency analysis for each line outage. They proposed a new severity index for line flows as the output of the network. The following parameters were selected as input of the network: (a) Pre-contingency line MVA in the outage element between buses p and q (S_0^{Pre}), (b) Product of pre-contingency bus voltage magnitude ($V_p \times V_q$), (c) Difference of the pre-contingency bus voltage angles of the buses p and q ($\theta_p - \theta_q$) and (d) A pre-contingency line MVA margin index (M_{MVA}). Through correlation technique they chose the first and last parameter as the input features of the ANN and they showed that the convergence of the network is better when trained with selected features than with all the parameters. Though the generating units are dispatched to run under optimal economic condition, it is not unusual for the system to run under non-optimal economical operating conditions. Under these conditions, the input parameters mentioned above may not be sufficient to estimate the output severity index accurately. Also, methods based on correlation coefficient can measure the linear relationship alone and cannot measure arbitrary relations between the variables.

2.4 Proposed Work on Contingency Selection

The present work focuses on developing neural network-based models following one-contingency-one-network strategy for contingency selection. The neural network-based approach is composed of a learning stage and a recall stage. During the learning stage, the networks are trained in a supervised manner using the available input-output patterns obtained from the system or derived from off-line simulation. During application, the network receives an input vector from the power system and estimates the corresponding post-contingency security index. This estimation is based on the knowledge the network acquired during training.

In developing a suitable model for contingency selection, the following issues are addressed:

(a) **Input output parameter selection:** This involves choosing appropriate input attributes to represent the state of the power system, and a quantity to reflect the security status of the system following an outage. A security index is introduced to reflect the system severity in the post-contingency period. The input parameter of the network is chosen among the set of state variables based on their estimation accuracy in predicting the security index.

(b) **Generation of training data:** In automatic learning methods, training data is the only available information to build the model, and so it should represent the range of the operating conditions of the system. The training data is generated by conducting a number of contingency analyses in off-line. The main tool used for contingency analysis is the Newton-Raphson load flow program.

(c) **Selection of suitable subset of features:** Real power systems have t-housands of variables at the system level. If all the measured variables are used as input to the ANN, the structure of the network will be complex and the training will take long time. To reduce the number of inputs to a neural network, it is necessary to select the key variables and discard the attributes that do not carry significant information to predict the security status. This will enable us to exploit the local nature of many security problems. This dimension reduction is carried out for each contingencies.

(d) **Model configuration and validation:** Once the appropriate features of the model are identified, the model is configured to capture the underlying relationship between the input and output using the training data. The trained models are tested with a separate set of input-output data to assess its generalization capability.

Fig 2.2. gives the overall picture of the learning stage of the proposed ANN-based model. First a large number of training data is generated through off-line power system simulation. Input features are selected to reduce the dimensionality of the input vector, using a feature selection technique. Selected input features are normalized and applied to the neural network along with the corresponding output, to learn the input-output relationship.

Development of the ANN-based model for an application requires lot of human intervention in the form of selecting the number of hidden nodes and number of input features. To reduce the burden in designing the neural network-based model, an attempt is made to select the number of hidden nodes and the input features automatically with the help of genetic algorithm.

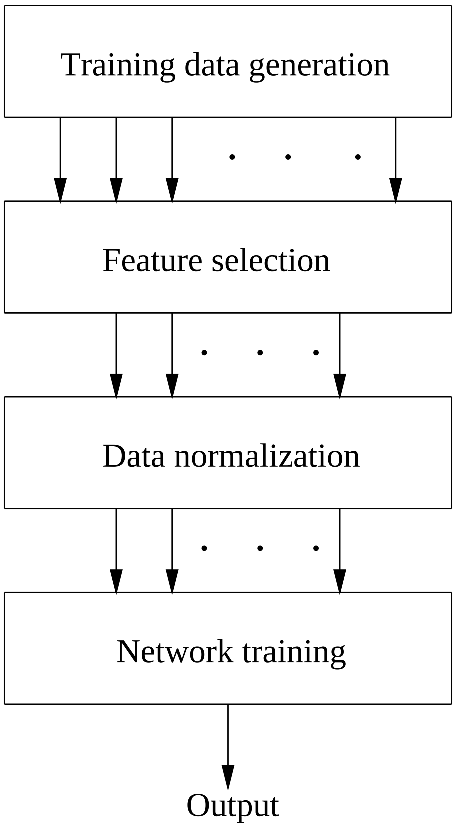

Fig. 2.2: Schematic representation of the learning stage.

2.5 Security Enhancement Techniques

Network overload can be alleviated by re-routing power flows in the system. This involves the rescheduling of power system controls via the automatic generation control/economic dispatch channels. Where present, phase shifter angles and HVDC link flows can also be controlled. When these measures are inadequate, load shedding must be carried out. This work focuses on generation rescheduling and phase shifter control only. In [72, 73], the linearized relationship between power flow in the overloaded lines and the generated power has been utilized to reschedule the power generation. By recognizing that transmission line overloads can impinge on the reactive-power balance, a computationally simple algorithm has been developed in [59]. Though these approaches are fast in rescheduling the power generations, it may lead to overloading in the other lines. For secure operation of the system without any limit violation, complete modeling of the system through Optimal Power Flow (OPF) algorithm is necessary [77]. The following subsection briefly reviews the various optimization techniques proposed in the literature to solve the OPF problem.

2.5.1 Solution Methods for Optimal Power Flow

The solution methods for OPF problem are based on general optimization principles and techniques specially adapted to exploit the structure of the power system problem. The major components of any optimization process are

- The minimum-seeking approach
- The approach for handling equality constraints
- The approach for handling inequality constraints

2.5.1.1 Mathematical Programming Methods

A number of mathematical programming based methods have been proposed to solve the OPF problem. The minimum seeking mechanism in most earlier OPF methods was first order descent [3,30]. These methods, employing first-derivative gradient information in determining the direction and length of the next step, were found to converge unpredictably. They had a tendency to zigzag or stop the floor of a valley was reached [117].

Sun *et.al.* [118] proposed Newton approach, which exploits the sparsity technique and solves active/reactive optimization problem. However, the Newton OPF has problems in dealing with inequality constraints.

Another important nonlinear optimization method proposed for OPF problem is the successive quadratic programming [15]. At each step, the objective function is approximated as quadratic and the constraints as linear. Then the resulting quadratic programming problem is solved. This is repeated successively until the nonlinear problem is solved.

One of the fully developed methods now in common use is the linear programming-based method [4,116]. In the linear programming approach, the smooth nonlinear curves are approximated as piece-wise linear by the algorithm. As the solution progresses, the sizes of the linear segments are gradually reduced, resulting in improved accuracy.

In general, the conventional calculus-based optimization algorithms rely on convexity to obtain the optimal solution. Hence the input-output characteristics of thermal generators are usually approximated by quadratic functions or piecewise quadratic functions. However, the OPF problem is in general non-convex, and as a result, many local minima may exist. In the case that the input-output curves of the generators are highly nonlinear and contain discontinuities due to the effects of 'valve points', the fuel cost curves of the generators are non-monotonic and the conventional methods have difficulties in determining the global optimum solution for OPF problem [120]. Further, under contingency conditions conventional optimization methods may result in divergence in solving OPF problems.

2.5.1.2 Genetic Algorithm-based Method

Genetic Algorithm (GA) offers a promising alternative in solving the optimal power flow problem [60]. GA doesn't require any space limitations such as smoothness, convexity or unimodality of the function to be optimized. This feature makes it suitable for many real world applications including optimal power flow problem.

In the traditional binary-coded genetic algorithm, the decision variables of the problem are represented by a fixed length string of binary bits (0,1). For problems with more than one decision variable, each variable is usually represented by a substring. All substrings are then concatenated together to form a bigger string. For example, if a problem has two decision variables u_1 and u_2, and if four and six bits binary coding are chosen for the first and second variables, respectively, then a solution will be represented as follows:

$$(u_1,\, u_2) = 1\ 0\ 1\ 0 \mid 1\ 0\ 0\ 1\ 1\ 0$$

The resolution of the solution depends upon how many bits are used to represent the variables. While evaluating an individual in the population, the binary strings are first decoded into decimal form and then the fitness value is calculated using the decoded value. The relationship between the actual parameter value (u_i) and the decoded string can be expressed as,

$$u_i = u_i^{min} + \left[\frac{(u_i^{max} - u_i^{min})}{(2^{l_i} - 1)} \right] d_i \tag{2.2}$$

where l_i is the string length used to code the i the parameter, d_i is the decimal equivalent of the binary string representing the parameter and u_i^{min} and u_i^{max} are the lower and upper bounds of the variable.

The coding of real-valued variables in finite-length strings causes a number of difficulties [22]. One difficulty is the Hamming cliffs associated with certain strings, from which the transition to a neighboring solution requires alteration of many bits. Hamming cliffs in the binary coding cause artificial hindrance to a gradual search in the continuous search space. Another difficulty is the inability to achieve any arbitrary precision in the optimal solution. The more the precision required, the larger the string length. For large strings, the population size requirement is also large, thereby increasing the computational complexity of the algorithm. Since fixed mapped coding is used to code the variables, the variable bounds must be such that they bracket the optimum variable value. In many problems, this information is not usually known a priori, this may cause some difficulty in using binary-coded GAs for those problems.

2.6 Issues in GA-Based Optimal Power Flow Problem

This thesis proposes a GA-based method to solve the optimal power problem which can be used for power system rescheduling. The following issues are addressed in developing the GA-based optimal power flow program.

(a) **Representation of solution:** In the traditional GA, the solution variables are encoded as binary strings. For the problems with continuous variables this is equivalent to discretization of the search space. This discretization results in a number of difficulties during the GA run, as explained in section 2.5.1.2. Instead, a natural representation of the solution variables is proposed.

(b) **Evaluation of the solutions:** The power system operator has the control to vary a few variables in the system only. Some of these independent variables called control variables will be represented as the solution variables in the GA population. From these candidate solutions (control variables), the dependent variables (state variables) of the system have to be derived and those candidate solutions which result in violation of upper or lower limits of the state variables have to be penalized to discourage the infeasible solutions.

(c) **Genetic operators:** The simple crossover and the bit mutation operators can effectively deal with the binary representation only. New crossover and mutation operators which can effectively deal with the non-binary problem representation have to be investigated.

CHAPTER 3

Data Generation for Security Study

3.1 Introduction

Building a neural network-based model for contingency selection requires large amount of training data. Each training data is an example contingency analysis, made of: (a) A description of a particular operating condition of the power system, and (b) An information on the system robustness when facing a contingency under study. The data can be collected either from actual power system operation or through off-line power system simulation. This chapter presents the details of generation of input and output data through off-line power system simulation for IEEE 14-bus and IEEE 30-bus test systems. Finally, the result of input parameter selection problem based on the generated training set is presented.

3.2 Security Index for Line-flow Severity

The information on the robustness of the system in facing the contingencies can be of two kinds : Discrete classification (secure/insecure), continuous security margin based on security index. A scalar security index (s_k), similar to the one introduced by Ejebe and Wollenberg [35] is considered here to represent the robustness of the system:

$$s_k = \sum_{l=1}^{N_l} \left[\frac{S_l^{post}}{S_l^{max}} \right]^{2m} ; k = 1, 2, \cdots, N_c \tag{3.1}$$

where S_l^{post} post-contingency MVA flow in line l

S_l^{max} rating of line l in MVA

m an integer exponent

N_l no of lines in the system

N_c no of contingencies in the contingency list

This security index reflects the loading condition of the system in the post-contingency state. A small exponent in the above equation would result in "masking effect". To avoid this undesirable effect, one can consider only the overloaded lines into account in the computation of security index as was proposed in [2]. Such an approach may be effective if security index is directly derived from the load flow results. But while using automatic learning tools like artificial neural networks to capture the underlying relationship between system operating condition and post-contingency security index, the network may not be able to capture the complete relationship if security index is set to zero for non-violation cases. Another approach for overcoming the masking problem is to set a large value for m. Proper value of m for a system will need some experimentation which would eliminate or reduce the masking effect.

3.3 Training data generation

The generation of the appropriate training data for the ANN-based model is a very important factor for its design. The training data must represent the complete range of operating conditions of the system under consideration. The training data is generated in two steps. (a) A range of situations is directly generated at random out of a base case. (b) For each situation, robustness of the system is determined through contingency analysis. The range of system situations is determined by the variation of the parameters which have a strong effect on the power system security. Many system variables have a direct effect on system security, and these variables include: load demand, generator output, reactive power injection by capacitor/reactor at different locations, transformer taps etc. Ideally, variations of all these parameters should be considered for generating the training data. However, for simplicity and to keep the training data to a minimum, only the load demand is altered during generation of training data.

First, a range of operating conditions is generated by randomly perturbing the load at all buses between 70% to 140% of their base case value and by adjusting the output of each generator in proportion to the output in the base case condition. For each load-generation pattern thus generated, the power flow equations given by Equations 1.1 and 1.2 are solved using Newton-Raphson algorithm to obtain the pre-contingency operating state. For each operating condition, the contingencies specified in the contingency list are simulated sequentially, and their security index values evaluated by solving the power flow equations after modification. The outages of the generating units need a modification of the power generation of remaining generators. For the outage of a line, the base-case admittance matrix has to be modified. While simulating the line outage contingencies, it is assumed that the control variables ($\mathbf{u}$) do not change. But the state variables ($\mathbf{x}$) change due to the physical behavior of the power system.

3.4 Test Systems

Two test systems, IEEE 14-bus system and IEEE 30-bus system were selected for carrying out various security related studies in this work. The system data for IEEE 14-bus and IEEE 30-bus systems are given in Appendix A and Appendix B respectively. The description of the generation of input and output data for these two test systems is presented below.

As shown in Fig 3.1, IEEE 14-bus system has has 5 generator buses, 9 load buses and 20 transmission lines. Generators are connected to buses 1 and 2, and synchronous condensers are connected buses 3, 6 and 8. Bus number 1 is the slack bus. Bus number 7 is a connecting to bus and no loads are connected to this bus. Tap changing transformers are connected to lines 4-7, 4-9 and 5-6. The assumed values of MVA ratings of the transmission lines are given in Table A.3, and the generating bus voltages are given in Table A.1. For this test system 19 single line-outages (which do not cause islanding) were selected for contingency analysis. The base case real power output of generator at bus number 2 is taken as 40 MW [84]. A total of 750 input-output pairs were generated, with 500 for training and another 250 for testing. A Matlab-based power system toolbox [135] was used to conduct the load flow studies. It was noticed from the analysis that the security

Table 3.1: Contingency list for IEEE 14-bus system

Contingency No.	Outage line (between buses)	Overloaded lines (between buses)
1	1-2	1-5, 2-5, 4-5
2	2-3	2-4, 2-5, 3-4
3	2-4	1-5
4	2-5	1-2, 2-4, 2-5
5	4-5	2-4
6	5-6	4-7, 7-9, 9-10, 10-11

index values are very low for 13 out of 19 contingencies. These contingencies are not considered essential for model development. Hence models are developed for the remaining six cases only. Table 3.1 gives the selected contingencies along with the lines in which MVA flow has been violated. From this table, it is evident that the effect of most of the contingencies is localized in nature except for the line outage 5-6 (which is a tie-line) whose impact is widespread.

IEEE 30-bus system (Fig 3.2) has 6 generator buses, 24 load buses and 41 transmission lines. Tap setting transformers are connected to the lines 6-9, 6-10, 4-12 and 28-27. The optimal power flow solution given in [3] is taken as the base case condition of the system. For this test system, 40 single line outages (which do not cause islanding) were considered for security analysis. A total of 1000 input-output pairs were generated following the procedure given in the previous

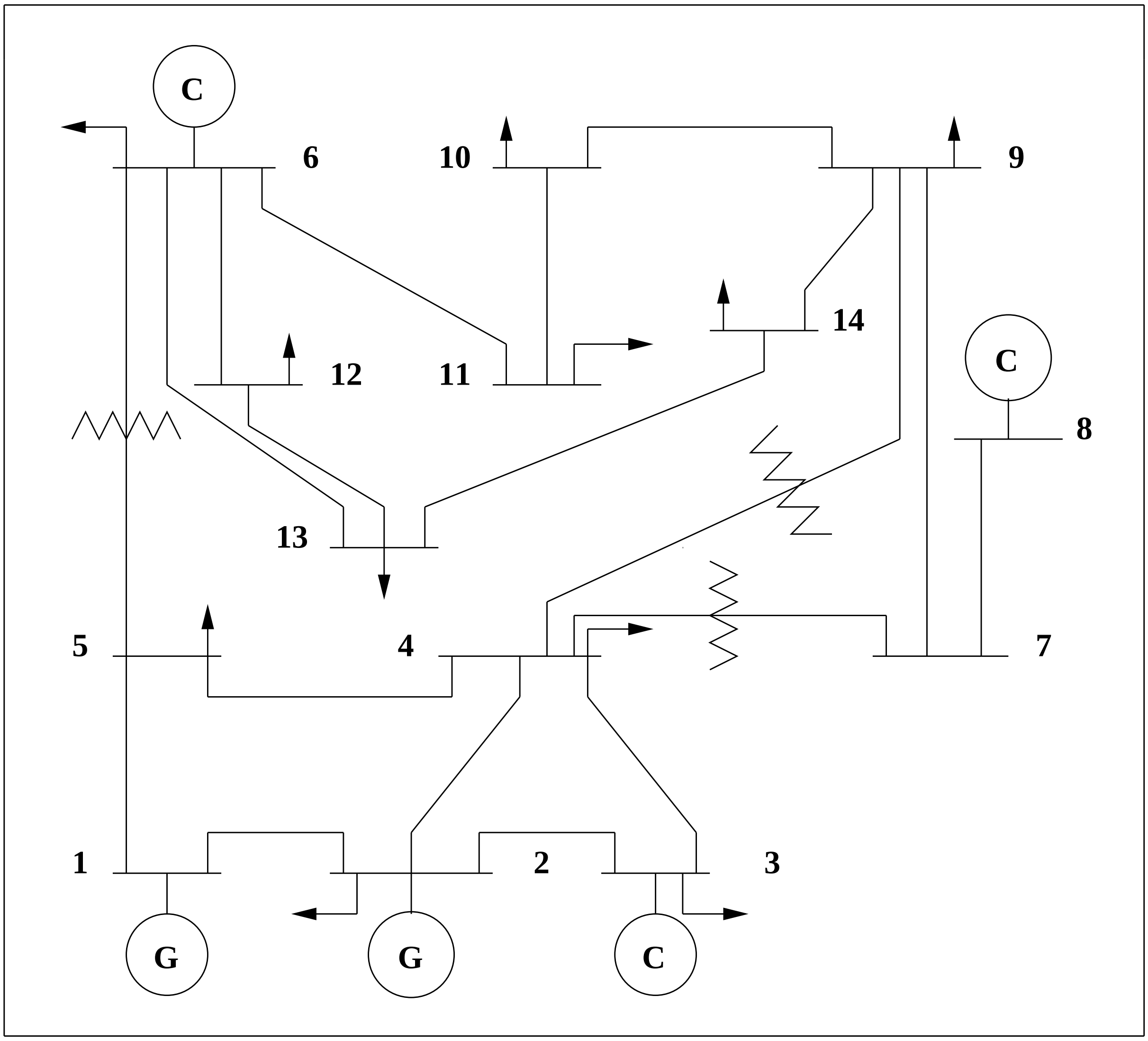

Fig. 3.1: IEEE 14-bus test system.

Table 3.2: Contingency list for IEEE 30-bus system

Contingency No.	Outage line (between buses)	Overloaded lines (between buses)
1	1-2	1-3, 3-4, 4-6
2	1-3	1-2, 2-4, 2-6
3	3-4	1-2, 2-4, 2-6
4	2-5	2-4, 2-6, 4-6, 5-7, 6-7
5	4-6	1-2, 2-6
6	6-7	1-2, 2-5
7	9-10	6-10, 12-15, 16-17, 15-18, 25-27
8	19-20	15-18
9	27-28	12-15, 10-21, 15-23, 22-24, 23-24, 24-25, 25-27

section, with 750 for training and another 250 for testing. Table 3.2 gives the list of contingencies selected for model development along with the lines in which MVA flow has been violated after the line-outage. From this table, it is evident that the effect of most of the contingencies is localized in nature except for the line outage 28-27, whose impact is widespread and is felt even in the fourth tier elements (lines 12-15 and 15-23).

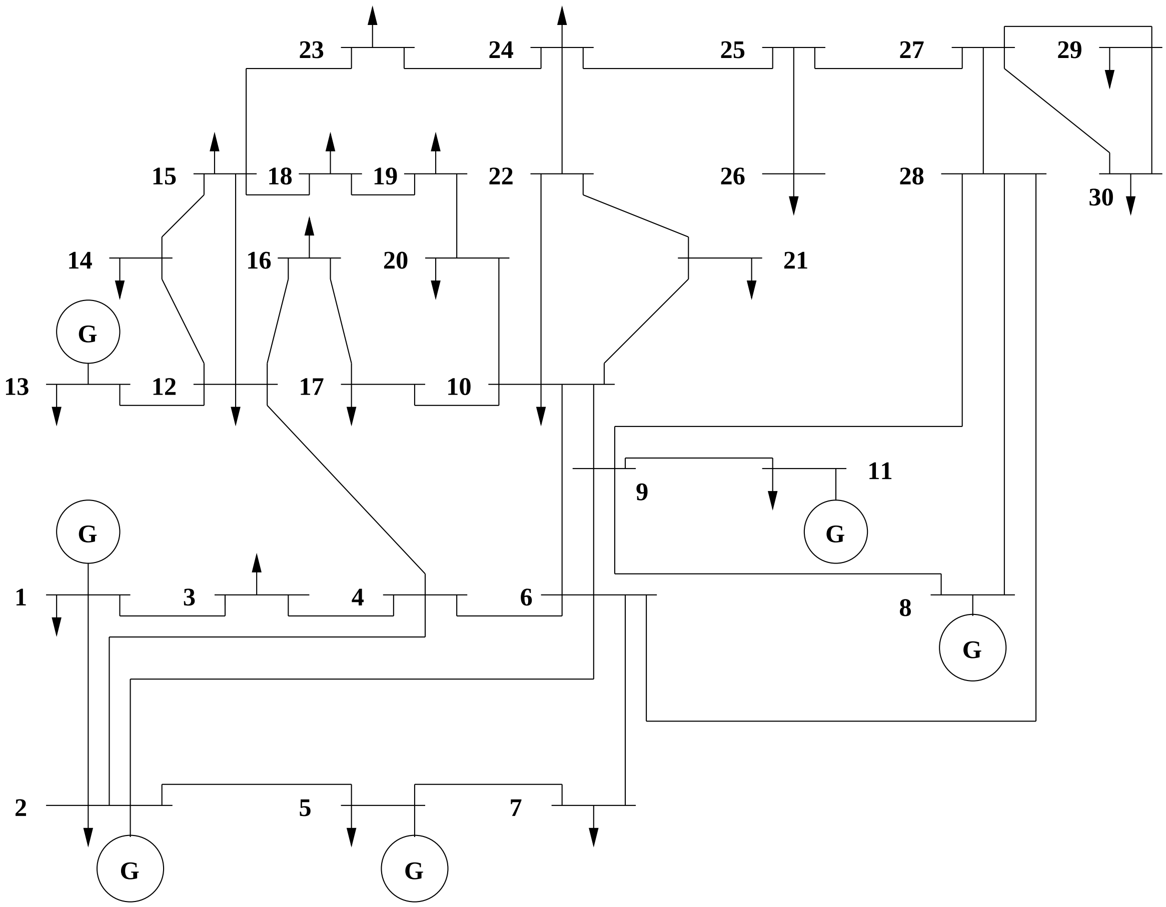

Fig. 3.2: IEEE 30-bus test system.

3.5 Normalization of the Data

During training of the neural network, higher valued input variables may tend to suppress the influence of smaller ones. Also, if the raw data is directly applied to the network, there is a risk of the simulated neurons reaching saturated conditions. If the neurons get saturated, then the changes in the input value will produce a very small change or no change in the output value. This affects the learning to a great extent. To minimize the effects of magnitudes among inputs as well as to prevent saturation of the neuron activation function, the input data is normalized before being presented to the neural network. One way to normalize the data x is using the expression:

$$x_n = \frac{(x - x_{min}) \times \text{range}}{(x_{max} - x_{min})} + \text{starting value of the normalized input} \qquad (3.2)$$

where x_n is the normalized value and x_{min} and x_{max} are the minimum and maximum values of the data. The raw data is normalized between the range -1.0 and +1.00 or 0.0 and +1.0 depending on the nature of the neuron activation function used.

In the case of output variable, if it assumes values close to the boundary of the activation function it causes difficulty in training. A way to overcome this difficulty is to normalize the variable between say -0.9 and +0.9 or 0.1 and 0.9.

3.6 Input Parameter Selection

For the application of machine learning approaches, it is important to properly select the input variables. The importance of this variable selection process is evident from the fact that ANNs are supposed to learn relationships between input and output variables on the basis of input-output pairs provided during training. If all the relevant variables are not used as input, ANN could not be expected to accurately predict the output variables.

In the ANN-based contingency selection models, the input variables represent the operating state of the power system under consideration. The state variables which are important from the point of view of estimating the security of the system are:

(a) Voltage magnitude and phase angle at each bus

(b) Real and reactive power injection at each bus

(c) Power flows in various lines

The advantage of using bus injections and line flows is that they are available as measurements while complex bus voltages must be estimated. But the final choice of the input variables depends on their prediction accuracy in estimating the post-contingency security index.

Experiments were conducted to assess the estimation capability of bus injection and line MVA flows using feedforward neural networks trained by backpropagation algorithm (Appendix C). The input-output dataset generated for the IEEE

Table 3.3: Estimation performance for different features

Input variable	Training time (sec)	Testing error (mse)
Bus injection	130	2.43×10^{-4}
Line MVA flow	75	1.46×10^{-4}

14-bus test system was used to develop the network. All-contingency-one-network approach was followed in the network development. Neural network tool box from Matlab [24] was used to develop the feedforward networks. Hyperbolic sigmoidal functions were used in the hidden layer nodes and linear functions were used in the output nodes. The networks were trained with the training data set using conjugate gradient algorithm [76] with the error goal of 0.0001. Conjugate gradient algorithms are faster in training than the standard gradient descent algorithms. Trial and error procedure was followed to select the suitable number of neurons in the hidden layer. After training, the network was tested with the test data set to assess the generalization capability of the developed network. The results of these experiments are presented in Table 3.3.

Though both the variables resulted in the successful development of networks for security index estimation, it is found that the networks trained with bus injection take more time for training. This is mainly due to the increased dimensionality of the bus injection vectors than the MVA flow vectors. Also, line MVA flows explicitly carries the information about the loading condition of the system in the pre-contingency state, which influences post-contingency security index. Hence it was decided to use line MVA flows as input for subsequent studies in this work.

3.7 Summary

In this chapter, a security index based on line MVA flow has been introduced to describe the robustness of the system in facing the contingencies. A set of input-output patterns which describe the security status of the system for various operating conditions has been generated using off-line contingency analysis. Experiments were conducted to analyze the ability of bus injections and pre-contingency line flows as features to estimate the post-contingency security index. It was found that the networks with line MVA flow as input learns faster compared to bus injection as input feature.

CHAPTER 4

Input Feature Selection

4.1 Introduction

The ability to scale neural network applications to realistic dimensions for power system problem is a major issue. Real power systems have thousands of variables at the system level. If all the measured variables are used as inputs to neural network, it results in large size of the network and hence larger training time. To make the neural network approach applicable for large scale power system problems, some dimensionality reduction is mandatory. Also, networks involving too many input variables suffer from *curse of dimensionality* [12]. A network with fewer inputs has fewer adaptive parameters to be determined, and these are more likely to be properly constrained by a data set of limited size, leading to a network with better generalization properties.

There are two different approaches to achieve dimensionality reduction: Feature extraction and feature selection. Feature extraction, linearly or nonlinearly transforms the original set of features into a reduced one. Well known feature extraction methods are Principal Component Analysis (PCA) and Discriminant analysis [32]. On the other hand, feature selection selects a subset of features from the initial set of available features. By reducing the number of features considered by a network, both techniques can improve estimation speed and efficiency. The choice between feature selection and feature extraction depends on the application domain. Feature selection leads to savings in measurement cost and the selected features retain their original physical interpretation. Hence, feature selection is more relevant to power system applications and is followed in this thesis.

This chapter proposes a **mutual information**-based technique to select the input features of the neural networks. The effectiveness of the proposed method is demonstrated through feature selection in security index estimation network.

4.2 Feature Selection Problem

The problem of feature selection is defined as follows: Given a set of D features, select a subset of size M that leads to the smallest estimation error. The most straightforward approach to the subset selection problem is the exhaustive search of all feature subsets. The total number of subsets (N_s) that need to be considered is

$$N_s = \binom{D}{M} = \frac{D!}{M!(D-M)!} \tag{4.1}$$

While it is true that the best subset can only be obtained through the exhaustive search procedure, the amount of search involved is often prohibitive.

A number of suboptimal feature selection methods have been proposed in the literature for practical applications. These approaches involve searching for an optimal subset of features based on some criteria of interest. They can be classified into two categories depending on whether or not feature selection is done independently of the learning algorithm used to train the network. If feature selection is performed independently of the learning algorithm, the technique is said to follow a *filter* approach (Figure 4.1a). The data is first analyzed, for instance, using statistical techniques, to determine which variables describing the data set are relevant for the given task. The relevant variables are used to train the network. No information from the subsequent performance of the network is used in selecting the features. In contrast, *wrapper* method incorporates the learning algorithm in the feature selection process (Figure 4.1b). Optimal feature selection is done by a search in the space of possible features. This means that subset of features are generated and evaluated. The evaluation is done by training and testing the network using each generated feature subsets. Wrapper approach often achieve better results than filter approach due to the fact that, it is tuned to the specific interaction between the learning algorithm and the training data. However, it tends to be much slower than filter method.

For power system security evaluation, Pang *et al.* [85, 86] proposed a filter approach to select the input features for their statistical security classifier. They used the *class separability* of the variables as the criterion to select the features. The following index F_i provides a measure of class separability (Secure/Insecure) with respect to the ith variable.

$$F_i = \left| \frac{m_i^S - m_i^I}{(\sigma_i^S)^2 + (\sigma_i^I)^2} \right|, \; i = 1, 2 \cdots, D \tag{4.2}$$

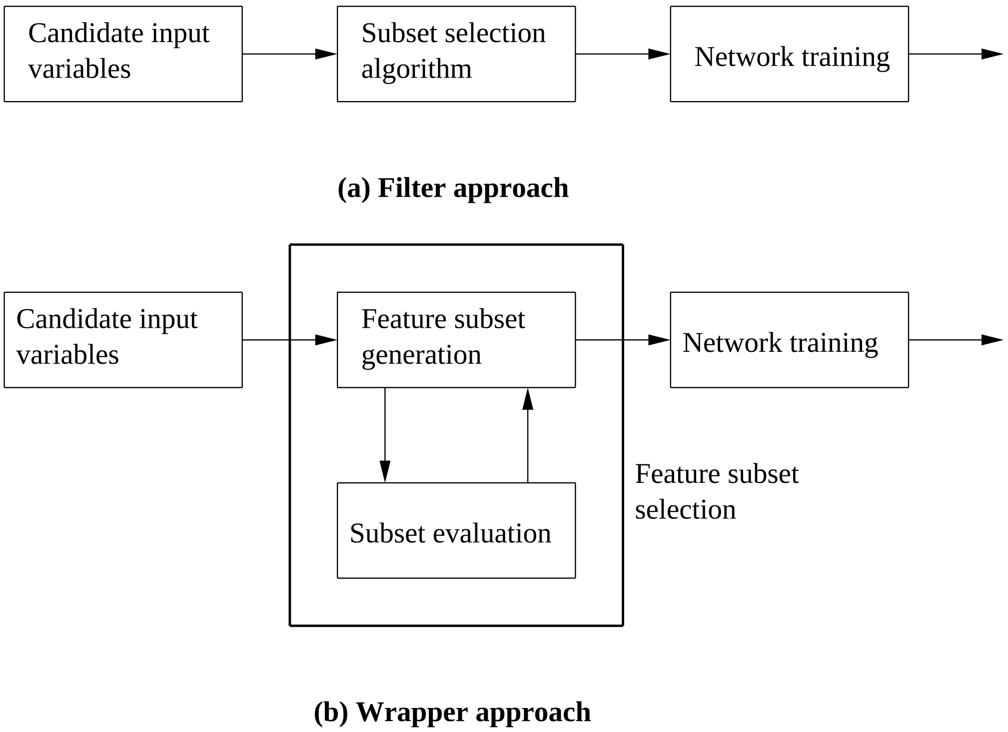

Fig. 4.1: Filter and wrapper approaches to feature selection.

where m_i^S and m_i^I are the mean of the variable i for the secure and insecure classes, and $(\sigma_i^S)^2$ and $(\sigma_i^I)^2$ their variances respectively. Variables with higher values of F_i carry more information about class separability than others. In [123] and [70] also the authors have used the same interclass-distance measure for feature selection in their neural network-based contingency selection models. Ghosh and Chowdhury [44], used the correlation coefficient between the input variables and the output severity measure to select the input features for their feedforward neural network model. The interclass-distance measure assumes Gaussianity in the input domain. If a vector has normal or Gaussian distribution, its distribution is completely characterized by its mean and variance. If the distribution of the input variables are non-Gaussian, errors might occur in the interclass distance-based feature selection. Also, methods based on linear dependence (like correlation) cannot measure arbitrary relations between the independent and dependent variables. A method based on mutual information is presented in the next section for selecting the input features of the network.

4.3 Mutual Information-based Feature Selection

4.3.1 Concept of Mutual Information

The degree of average uncertainty about a random experiment is described by *entropy*. Let a random experiment be defined by a discrete random variable Y with possible values $y_i, i = 1, 2, \cdots, N_y$, and let the probability distribution function of Y be $Prob(Y = y_i) = P_i$. Then the entropy of the random experiment is defined by the expression [92],

$$H(Y) = -\sum_{i=1}^{N_y} P_i \log P_i \tag{4.3}$$

If we have some additional information X regarding that random experiment, it will result in reduction in initial entropy. The *conditional entropy* of the random experiment after knowing the information X is

$$H(Y|X) = -\sum_{j=1}^{N_x} P_j (\sum_{i=1}^{N_y} P(y_i|x_j) \log P(y_i|x_j)) \tag{4.4}$$

where P_j is the probability distribution function of X with possible values x_j, $j = 1, 2, \cdots, N_x$, and $P(y_i|x_j)$ is the conditional probability for y_i given x_j. In general, the conditional entropy will be less than or equal to the initial entropy. The amount by which the entropy (uncertainty) is decreased is, by definition, the *mutual information* $I(Y;X)$ between Y and X:

$$I(Y;X) = H(Y) - H(Y|X) \tag{4.5}$$

The mutual information is therefore the amount by which the knowledge provided by X decreases the average uncertainty about the random experiment represented by the variable Y. Mutual information is a symmetrical measure. That is, the amount of information gained about Y after observing X is equal to the amount of information gained about X after observing Y. For the contingency selection problem under consideration, X corresponds to the pre-contingency line flows and Y corresponds to the post-contingency security index.

4.3.2 Computation of Mutual Information

To compute mutual information, the probability distribution of variables is needed which in practice is not known, and the best we can do is to use the histogram of the data. The steps involved in computing the mutual information from the histogram of the training data are given below:

Step 1 Arrange the output patterns (security index) corresponding to a line-outage in the descending order and divide the arranged patterns into N_y classes equally.

Step 2 Calculate the initial entropy of the output Y, when there is no information regarding the input variable using (4.3).

Step 3 Sort the patterns in the first input variable, X_1 (power flow in line 1-2) in the descending order and divide the sorted patterns into N_x groups equally.

Step 4 Compute the conditional entropy of Y, given the input variable X_1 using (4.4).

Step 5 Determine the mutual information of Y with respect to X_1 using (4.5).

Step 6 Repeat Steps 3 to 5 for the remaining variables also.

The above procedure is illustrated through the calculation of mutual information for variables in line outage 1-2 of the IEEE 30-bus system.

First to calculate the initial entropy, the output patterns are divided into 3 classes, y_1, y_2 and y_3, with y_1 containing the first 250 patterns having the largest security index values and so on. Now each class has the same number of load patterns and the probability that a certain pattern belongs to a class y_i is (250/750). Therefore the initial entropy is

$$H = 3 \times (250/750)\log(750/250) = 1.0986$$

To calculate the conditional entropy of Y, given the ith input variable X_i, the input patterns in X_i are divided into five groups, a_i, $i = 1, 2, \cdots, 5$, with x_1 containing those patterns with largest power flow and so on. Now, each group contains 150 patterns. Tables 4.1 and 4.2 shows the input-output distribution for the variables X_1 and X_{15} respectively. From Table 4.1, it is observed that when power flow in line 1 is high (group x_1), there is a growing tendency for the corresponding security index to fall in the class with higher values of security

Table 4.1: Distribution of the patterns for variable X_1

Group	y_1	y_2	y_3
x_1	150	0	0
x_2	100	50	0
x_3	0	146	4
x_4	0	54	96
x_5	0	0	150
Conditional entropy = 0.2826			

index values (class y_1) and so on. This indicates that there is a strong relationship between input variable A_1 and security index values. The mutual information in this case is found to be 0.8160. In contrast to this, the input patterns (X_{15}) in Table 4.2 are uniformly distributed, which indicates that the relationship between the input variable X_{15} and the output security index is not strong. The mutual information value in this case is found to be 0.0022.

For illustration, the mutual information between all the input variables and the security index for line-outage 1-2 is shown through a bar chart in Fig 4.2. From this figure it is evident that only a few variables (variables 1 to 9 and 13) are having significant value of mutual information and the remaining variables have very less value. The variables which have high value of mutual information are the ones which represent the pre-contingency power flows on the lines near the outage line. This shows that line flows in the vicinity of the outage line posses more information about the security level of the system.

Table 4.2: Distribution of the patterns for variable X_{15}

Group	y_1	y_2	y_3
x_1	47	50	53
x_2	46	49	55
x_3	51	51	48
x_4	50	48	52
x_5	56	52	42
Conditional entropy = 1.0962			

4.3.3 Feature Selection Through Mutual Information

The observation that useful information will result in reduction in system entropy provides the basis for feature selection. If the information regarding a certain system variable results in significant reduction in the system entropy, then this variable must have significant impact on the security index which is to be estimated. Therefore, this variable will be selected as a feature for security index estimation. On the other hand, the system variables which result in minor reduction in the system entropy will be regarded as having minor effects on the security index and will not be selected as feature.

Once the mutual information value of input variables is evaluated, the variables are ranked, with the variable having the highest mutual information value at the top and so on. The optimum number of features can be selected by consequent training of the neural networks using a progressively increasing number of features

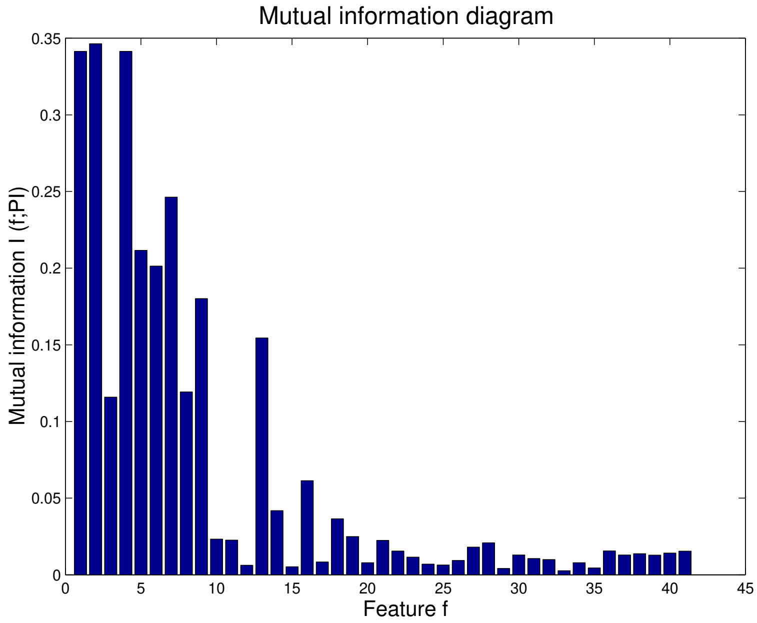

Fig. 4.2: Mutual information for variables in model 1-2.

until the minimum required accuracy is obtained.

4.4 Experimental Results

To demonstrate the effectiveness of the proposed feature selection method, a feed-forward neural network was developed to estimate the security index corresponding to line-outage 1-2. As a first step, the first four variables in the ranked list ($S_{1-2}, S_{1-3}, S_{3-4}$ and S_{2-5}) were taken as the input features and the network was trained with the training data set generated in the previous chapter. Conjugate gradient algorithm was used to train the network. The network was trained until it reaches the mean square error value of 0.0001. After training, the network was tested using the test data set to evaluate the generalization performance. Trial and error procedure was followed to select the number of neurons in the hidden layer. The optimum performance of the network was obtained with 8 neurons in the hidden layer. The performance of the network during training and testing phase is summarized in Table 4.3. From this result, it is evident that the proposed method has effectively identified the important features for security index estimation. Figure 4.3 shows the plot of the network output and the actual security index for the 250 test patterns. The network output is close to the actual value in almost all the cases. This shows that the network has properly learned the input-output relationship with the selected features and is able to estimate the security index corresponding to different operating conditions. The complete details of feature selection and network development for all nine models will be presented in the next chapter.

For comparison, an unified neural network with all the pre-contingency line MVA flows as the input and the security indices corresponding to the 9 selected contingencies as output was developed. The performance of this network during training and testing phase is given in Table 4.4. Comparing Tables 4.3 and 4.4, it is evident that the network developed with the selected features (four in number) has achieved the same generalization performance as that of the network developed with all input variables. But, with selected input features the training time has reduced considerably.

Table 4.3: Performance of the network for line outage 1-2

Selected features (MVA flow)	No. of hidden nodes	Training time (sec)	Testing error (mse)
1-2, 1-3, 3-4, 2-5	8	9.1	9.6×10^{-5}

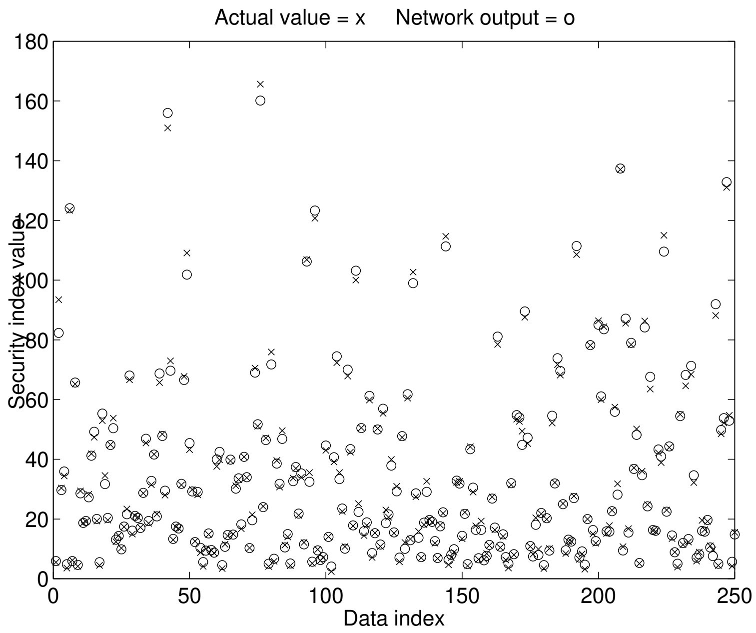

Fig. 4.3: Plot of the network output and actual value.

Table 4.4: Performance of the unified network

Input features	No. of hidden nodes	Training time (sec)	Testing error (mse)
Line MVA flows	20	250	1.57×10^{-4}

4.5 Summary

A key issue in neural network-based approach is identifying a representative set of features from which to develop the network for a particular task. This chapter has addressed the problem of feature selection through mutual information between the input variables and the output security index. Experimental results indicate that the proposed method is effective in identifying the important features of the network. Both in the learning and application, the performance of the network is significantly improved by reducing the input dimension.

CHAPTER 5

Contingency Selection Using Radial Basis Function Networks

5.1 Introduction

Contingency selection is performed to identify the contingencies that cause limit violation in the post-contingency period. The main task in contingency selection is the computation of the post-contingency security index for each contingency. Computation of security index by analytical methods takes long time, as load flow analysis has to be run for every single contingency to calculate the post-contingency line flows. In recent years there has been an increasing interest in applying artificial neural network models for contingency selection [44, 89, 100, 119].

Feedforward neural networks trained by backpropagation algorithm is widely used for many applications including power system security assessment Provided that the network structure is sufficiently large, any continuous function can be approximated to within an arbitrary accuracy by carefully choosing parameters in the network [50]. But the shortcoming of this network is that it takes long time for training. Also, feedforward networks with sigmoidal nonlinearities have no inherent ability to indicate when they are functioning outside the domain over which they were trained. Even though training is done off-line, short training time is preferred as one may have to retrain the networks on a regular basis as the topology or the system condition changes. For many reasons one cannot be assured that the cases that would be presented to the network during the application phase are completely and accurately represented by the training set. This is particularly true for power system problems, as it is not possible to simulate all possible operational conditions. Also, the system may be trained on synthetic data derived from the power system simulation which does not exactly match its operation.

This chapter proposes radial basis function networks for security index estimation. Radial basis function networks take less time for training [26] and the distance-based activation function used in the hidden layer helps to identify when they are working outside the domain over which they were trained. The effectiveness of the proposed method is demonstrated through contingency selection in IEEE 30-bus test system.

5.2 Radial Basis Function Networks

The design of Radial Basis Function (RBF) network may be viewed as a curve fitting problem in high dimensional space [47]. According to this viewpoint, learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data. Correspondingly, generalization is equivalent to the use of this multidimensional surface to interpolate the test data.

5.2.1 Structure of RBF network

Radial basis function network consists of three layers of nodes: Input, hidden and output nodes. Fig 5.1 shows the structure of the RBF network with n inputs and a scalar output. The input nodes pass the input variables to the hidden layer unaltered, and the first layer connections are not weighted. The hidden nodes are the radial basis function units, and for most applications the number of hidden nodes is higher than the number of input nodes. Radial basis functions are typically of thin plate spline, Gaussian, multi-quadratic or inverse multi-quadratic functions [12]. It has been shown theoretically and practically that the performance of the radial basis function network does not critically depend on the choice of the functions [78] and generally Gaussian functions are used. The activation function of the hidden node (ϕ_j) is given by,

$$\phi_j(\mathbf{a}) = \exp\left(-\frac{\| \mathbf{a} - \boldsymbol{\mu}_j \|^2}{2\sigma_j^2}\right), \; j = 1, 2, \cdots, h \tag{5.1}$$

where $\mathbf{a}$ input vector

$\boldsymbol{\mu}_j$ vector of jth basis function center

σ_j width of jth basis function

h number of hidden nodes

Fig 5.2 shows a typical Gaussian function response with σ=1 and $\mu = 0$ in the range $-3 \leq \| a - \mu \| \leq 3$ in one dimension. Each radial basis function unit has a significant activation over a specific region determined by $\boldsymbol{\mu}_j$ and σ_j; Thus each radial basis function represents a unique local neighborhood in the input space.

The connections in the second layer are weighted and the output nodes are linear summation units. The value of the output node y is given by

$$y(\mathbf{a}) = \sum_{j=1}^{h} w_j \phi_j(\mathbf{a}) + w_0 \tag{5.2}$$

where w_j are the weights from hidden unit j to output node and w_0 is the bias term.

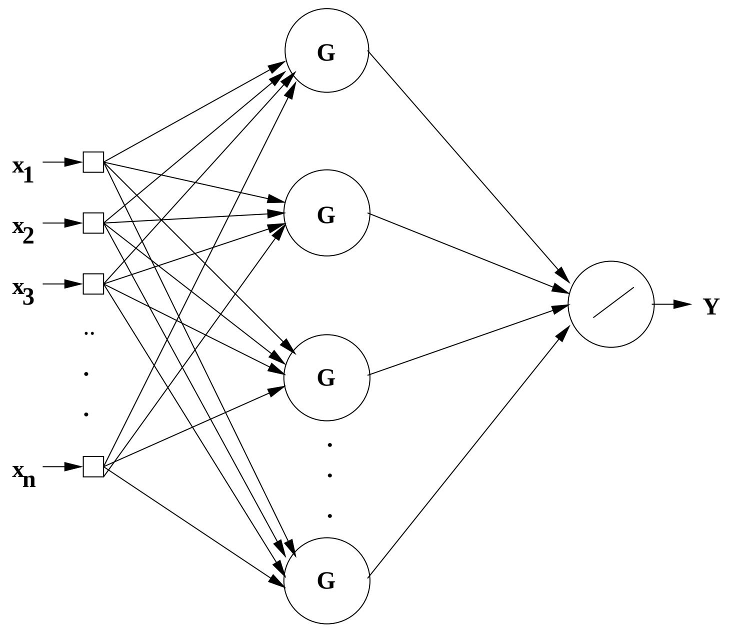

Fig. 5.1: Radial basis function neural network.

5.2.2 Training Algorithm for RBF Network

The major task in RBF network development is identification of the centers of the basis functions and the weights of the output layer based on the given training

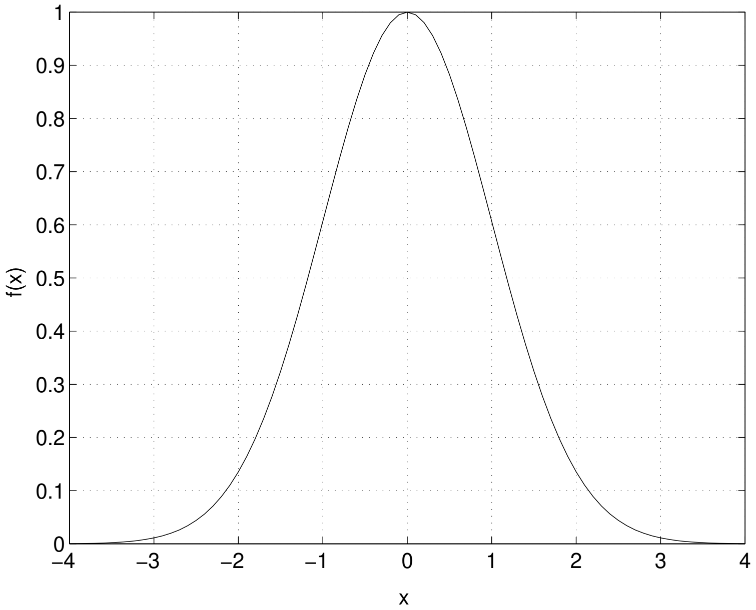

Fig. 5.2: Gaussian activation function.

data. For fixed centers, the weights of the output connections can be calculated through linear least square regression [55], as the output node is the simple summation unit. Since the regression correlates the desired network outputs with the hidden node activations, each input pattern ($\mathbf{a}^n$) is passed through the first layer of the network to the hidden nodes to produce a corresponding hidden node activation value, $\phi(\mathbf{a}^n)$, where n is the index of the training pattern. The objective is to find the set of weights $\mathbf{w}= [w_1, w_2, \cdots, w_H]^T$ that minimizes the squared norm of the error

$$\| \mathbf{d} - \mathbf{w}\boldsymbol{\phi} \| \tag{5.3}$$

The solution of the above problem is given by

$$\mathbf{w} = \boldsymbol{\phi}^{+}\boldsymbol{d} \tag{5.4}$$

where $\mathbf{d} = [d_1, d_2, \cdots, d_N]^T$ is the desired output for the N training samples and $\boldsymbol{\phi}^+$ is the pseudo inverse of the matrix

$$\boldsymbol{\phi} = \begin{bmatrix} \phi(\mathbf{a}^1;\boldsymbol{\mu}_1) & \phi(\mathbf{a}^1;\boldsymbol{\mu}_2) & \cdots & \phi(\mathbf{a}^1;\boldsymbol{\mu}_H) \\ \phi(\mathbf{a}^2;\boldsymbol{\mu}_1) & \phi(\mathbf{a}^2;\boldsymbol{\mu}_2) & \cdots & \phi(\mathbf{a}^2;\boldsymbol{\mu}_H) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{a}^N;\boldsymbol{\mu}_1) & \phi(\mathbf{a}^N;\boldsymbol{\mu}_2) & \cdots & \phi(\mathbf{a}^N;\boldsymbol{\mu}_H) \end{bmatrix}$$

Therefore the key issue in RBF network design is the determination of the the centers of the basis functions.

In the design method of RBF networks, the number of hidden nodes is equal to the number of training samples N and the centers of the basis function are located at the sample values $\mathbf{a}^n$, $n = 1, 2, \cdots, N$. This method can generate a RBF network with its output exactly equal to the desired outputs and free of errors when fed with an input vector $\mathbf{a}$ existing in the training set. However, this interpolation procedure may not be a good strategy for training of RBF networks for certain classes of tasks because of poor generalization to new data for the following reasons: When the number of data points in the training sample is much larger than the number of degrees of freedom of the underlying physical process, and we are constrained to have as many radial basis functions as data points, the problem is over determined. Consequently, the network may end up fitting misleading variations due to noise in the training data, thereby resulting in poor generalization performance. A suboptimal solution to the given problem is obtained by using fewer basis functions h $H < N$. Moody and Darken [78] proposed a three step procedure for learning in RBF networks, which is followed in this work.

Under this scheme the training in RBF networks is decomposed into three sequential stages. The first stage of the learning consists of determining the unit centers μ_{ij} by the K-means clustering algorithm [32]. The algorithm partition the data points $\mathbf{a}^n$, $n = 1, 2, \cdots, N$, into K disjoint clusters C_j, $j = 1, 2, \cdots, K$, containing N_j data points, in such a way as to minimize the sum-of-squares clustering function given by

$$J = \sum_{j=1}^{K} \sum_{n \in C_j} \| \mathbf{a}^n - \boldsymbol{\mu}_j \|^2 \tag{5.5}$$

where μ_j is the mean of the data points in cluster C_j. The algorithm iteratively determines the cluster centers μ_j as follows:

Step 1. Initialize the cluster center $\boldsymbol{\mu}_j$, $j = 1, 2, \cdots, K$ by randomly selecting K data points from among all of the data points.

Step 2. Generate a new partition by assigning each pattern to its closest cluster center.

Step 3. Compute new cluster centers as the centroids of the clusters.

Step 4. Compute the cost function according to Equation 5.5. Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold, otherwise go to step 2.

In practice, the use of K-means clustering will require experimenting with different numbers of clusters. The number of clusters is selected by the user. Next, we determine the unit widths σ_j using a heuristic approach that ensures the smoothness and continuity of the fitted function. The width of any hidden node is taken as the maximum Euclidean distance between the identified centers. Finally, the weights of the second layer connections are determined by linear regression, using a least-squares objective function.

Many practical applications show that the RBF networks are superior over other neural network approaches in the following senses [65, 66]. First, the RBF networks are capable of approximating nonlinear mappings effectively. Second, the training time of the RBF networks is quite low compared to that of other neural network approaches such as the multi-layer perceptron, because training of the two layers of the networks is decoupled. Third, the node activations in the hidden layer decrease with distance from the unit center and the network output will go to zero as the distance between a test point and the unit centers increases. The rate of this decrease is controlled by the scaling parameter σ. This gives the network the ability to signal cases far from training data.

One of the difficulties with RBF networks, however, also stems from the localized nature of the hidden units. It concerns the way in which such a network addresses the curse of dimensionality. The number of basis functions required to approximate a given function increases exponentially with the dimension of the input data. As well as increasing the computation time, a large number of basis functions leads to a requirement for large number of training patterns in order to ensure that the network parameters are properly determined. The problem is particularly severe if there are input variables which have significant variance but which play little role in determining the appropriate output variables.

5.2.3 Data Normalization

The first stage of RBF network learning is the identification of the cluster centers through k-means clustering algorithm, which uses Euclidean distance as a measure of dissimilarity. Distance norms are sensitive to variations in the numerical ranges of the different features. For example, the Euclidean distance assigns more weighting to features with wide ranges than to those with narrow ranges. To overcome this problem, input data is normalized before presenting it to the clustering algorithm. The input data is normalized between 0 and 1 using

$$x_n = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{5.6}$$

where x_{min} and x_{max} are the minimum and maximum of the variable x. Similarly, output data is also normalized between 0 and 1.

5.3 Results and Discussion

The effectiveness of the proposed approach is demonstrated through contingency selection in IEEE 30-bus system using the training and test data set generated in Chapter 3. As noted earlier, separate networks are developed for each contingency. The input to the networks are the 41 pre-contingency line flows and output is the post-contingency security index. The details of network development and performance evaluation for the nine contingencies listed in Table 3.2 are presented below.

5.3.1 RBF Network Development

First the mutual information between the input variables and the output is calculated following the procedure given in Section 4.3.2. Then the variables are ranked based on the mutual information value. The optimum number of features for each network is selected by subsequent training of the neural networks using a progressively increasing number of features until the minimum required accuracy is obtained. A small number of features is preferred since the complexity of the network increases rapidly with the number of features used. The selected features after normalization are presented to the network. Gaussian functions were used in the hidden layer of the RBF network. Twenty iterations of the clustering algorithm followed by linear regression is performed to estimate the parameters of the network. After training, the generalization performance of the networks are evaluated with the 250 test data.

Table 5.1 shows the selected features for the nine models. As expected, the selected feature subset consists of the pre-contingency line flow of the outage line along with the other variables in all the nine models, as the loading level of the outage line in the normal state will have a direct bearing on the security status of the system in the post-contingency period. Also, most of the selected features are the variables which represent the line flows in the vicinity of the outage line. This shows that the loading condition of the transmission lines near the outage line plays a major role in deciding the security status of the system in the post-contingency period. It is seen that, starting from 41 variables, estimation is done using a maximum of seven variables (for line-outages 19-20 and 28-27).

Table 5.1: Input Features for Security Index Estimation

S.No.	Outage line (between buses)	Selected features (Power flow between buses)
1	1-2	1-2, 1-3, 3-4, 2-5
2	1-3	1-2, 1-3, 3-4, 4-6
3	3-4	1-2, 1-3, 3-4, 4-6
4	2-5	1-2, 2-5, 5-7, 6-7
5	4-6	1-2, 1-3, 3-4, 2-6, 4-6, 9-11
6	6-7	1-2, 2-5, 5-7, 6-7
7	9-10	6-10, 9-10, 12-15, 16-17, 10-21, 10-22
8	19-20	1-3, 9-10, 12-15, 15-18, 18-19, 19-20, 10-20
9	28-27	22-24, 28-27, 27-29, 27-30, 29-30, 6-28

Table 5.2: Result of RBF Networks

S.No.	Line outage	No of basis functions	Training time (sec)	Testing error (mse)
1	1-2	15	0.50	$4.4 \cdot 10^{-5}$
2	1-3	15	0.50	$9.7 \cdot 10^{-6}$
3	3-4	15	0.50	$1.6 \cdot 10^{-5}$
4	2-5	20	0.70	$5.2 \cdot 10^{-5}$
5	4-6	20	0.75	$2.6 \cdot 10^{-4}$
6	6-7	20	0.65	$1.5 \cdot 10^{-4}$
7	9-10	25	1.00	$3.5 \cdot 10^{-4}$
8	19-20	35	1.40	$3.9 \cdot 10^{-4}$
9	28-27	25	0.95	$5.8 \cdot 10^{-4}$

Different numbers of basis functions were investigated in order to achieve the best performance. The number of basis functions which gave best performance for each of the nine models are presented in Table 5.2 along with the results of the testing phase. More number of basis functions used in case of models 9-10, 19-20 and 28-27 shows that the number of basis functions required increases with the dimension of the input vector. All the networks reached a reasonable level of accuracy within 20 iterations itself and their training time is also reasonably short.

Table 5.3 presents the security index values estimated for an operating condition when one line at a time is outage, with the ranking of the contingencies given

Table 5.3: Security Index Estimation

Line outage	RBF output	Load flow result
1-2	15.76 (3)	16.00 (3)
1-3	6.32 (4)	6.40 (4)
3-4	5.65 (5)	5.63 (5)
2-5	40.79 (2)	40.27 (2)
4-6	1.58 (8)	1.56 (8)
6-7	1.75 (7)	1.72 (7)
9-10	2.33 (6)	2.26 (6)
19-20	0.42 (9)	0.47 (9)
28-27	48.54 (1)	44.40 (1)

in the parenthesis. For comparison, security indices computed using the traditional method i.e. Newton-Raphson algorithm are also given. The contingencies are ranked using the estimated security indices; The most severe contingencies being those that have a higher security index value. It can be seen from Table 5.3 that the security index values are almost same for both methods. Also, for both methods, the contingencies are having the same ranking. This shows that training of the RBF networks has been successful and the correct estimation of performance index has been achieved by the RBF network even for previously unseen data.

5.3.2 Comparison with Multilayer Perceptron Network

In this section, the performance of the proposed RBF network-based approach is compared with the commonly used neural network architecture, multilayer perceptron (MLP) networks. Three layer networks with hyperbolic sigmoidal activation function in the hidden layer and linear activation function in the output layer were developed. The networks were trained to reach the same error level achieved by the RBF networks during training. The input features of the networks are the same as that used for their RBF network counterpart. The networks were trained with conjugate gradient algorithm. After training, the networks are tested with the test data. The results of training and testing are presented in Table 5.4. Comparing Tables 5.2 and 5.4 it is found that RBF networks take less time for training. Apart from that RBF network exhibits better generalization performance than the MLP network in most of the cases. but they require more number of hidden nodes as compared to MLP networks due to the local nature of the basis functions used in the hidden layer. Increased number of hidden nodes will increase the execution time of the networks during the application phase.

5.3.3 Performance of ANN-based Ranking

The objective of a ranking method is to filter and order the critical contingencies without performing AC load flow for each contingency. The effectiveness of a ranking method is determined by the following two characteristics: Ordering accuracy and computation time.

Table 5.4: Results of MLP Networks

S.No.	Line outage	No of hidden nodes	Training time (sec)	Testing error (mse)
1	1-2	8	9.1	$9.6 \cdot 10^{-5}$
2	1-3	6	5.3	$9.6 \cdot 10^{-5}$
3	3-4	6	5.4	$9.6 \cdot 10^{-5}$
4	2-5	8	17.7	$9.3 \cdot 10^{-5}$
5	4-6	8	18.3	$3.0 \cdot 10^{-4}$
6	6-7	6	15.5	$1.5 \cdot 10^{-4}$
7	9-10	6	16.0	$4.4 \cdot 10^{-4}$
8	19-20	6	16.2	$2.0 \cdot 10^{-4}$
9	28-27	6	16.1	$6.2 \cdot 10^{-4}$

5.3.3.1 Ordering Accuracy

The ordering accuracy indicates how correctly a method places the contingencies at the right positions in the ranking. The percentage ordering accuracy (R) is defined by [68]

$$R = \frac{\mathrm{M}^{max} - \mathrm{M}}{\mathrm{M}^{max}} \times 100 \tag{5.7}$$

where M is the mean misrank value, and M^{max} the maximum mean misrank value which are defined by

$$\mathrm{M} = \frac{\sum_{j=1}^{N_c} |O_j^{act} - O_j|}{N_c} \tag{5.8}$$

where O_j^{act} and O_j represent the actual order number and the ranking based on the estimated value respectively for contingency j and

$$\mathrm{M}^{max} = \frac{2}{N_c}[1 + 3 + 5 + \cdots + (N_c - 3) + (N_c - 1)], \text{if } N_c \text{is even} \tag{5.9}$$

$$= \frac{2}{N_c}[2 + 4 + 6 + \cdots + (N_c - 3) + (N_c - 1)], \text{if } N_c \text{is odd} \tag{5.10}$$

For the contingency ranking considered here, $N_c = 9$, and therefore $\mathrm{M}^{max} = 4.444$.

Using above expression, the percentage ordering accuracy was computed for all the 250 samples in the test set by comparing the ranking of the RBF network output and the actual ranking based on the AC load flow result. Figure 5.3 shows the histogram of the percentage ordering accuracy obtained for all the 250 samples. The satisfactory performance of the proposed approach in ranking the contingencies is evident from the above 90 % ordering accuracy obtained for most of the samples in the test set.

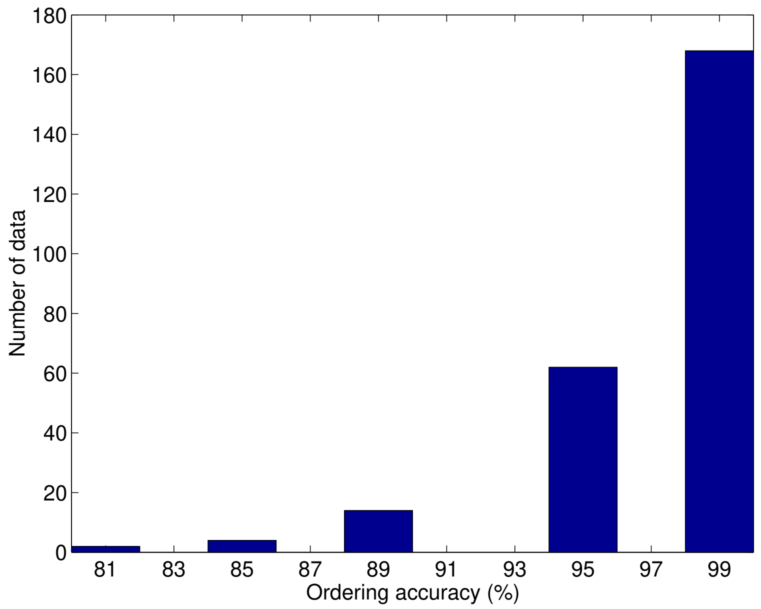

Fig. 5.3: Histogram of ordering accuracy.

5.3.3.2 Computation Time

Computational efficiency is one of the main reason for using of ANNs for real-time security assessment. The computation time indicates the efficiency of a ranking method. Table 5.5 shows the time taken to calculate the security index values of all the nine contingencies for an operating condition and the time taken to estimate the same from the trained neural networks. From this table it is found that both RBF network and the multilayer feedforward network take less time to estimate the security index values as compared to the time take to compute the security index values by conducting AC load flow. Among the neural networks RBF network takes more time, for the following reasons:,

Table 5.5: Time taken to calculate security index values

Method	Time taken (sec)
AC load flow	0.36
RBF network	0.01
FF network	0.0086

(a) RBF network requires more number of hidden nodes than the feedforward networks with sigmoidal activation function in the hidden layers to approximate a given function.

(b) The hidden layer of the basis function network involves computation of the nonlinear basis function values for each input vector in order to determine the outputs of the hidden units. These computations generally take much more time than for the linear basis function computations in a multilayer feedforward network.

5.4 Summary

This chapter has presented a neural network-based approach for contingency selection in static-security assessment. Radial basis function networks have been proposed to learn the nonlinear relationship between the pre-contingency operating condition and the post-contingency security indices. Radial basis function networks have been preferred over feedforward neural networks trained by backpropagation algorithm in view of their fast learning, and their ability to indicate when they are functioning outside the domain over which they were trained. Performance of the proposed radial basis function network-based approach in conjunction with the mutual information-based feature selection has been demonstrated through contingency selection in IEEE 30-bus test system. The proposed contingency selection scheme is accurate and very fast, so it can be implemented for on-line applications in power system control centers.

One contingency-one network approach has been followed in network development. The networks are trained and operate separately, but they collectively solve the contingency selection problem. Use of separate networks for each contingency considerably reduces learning time and memory requirement. The simple networks can operate in parallel, which enables the proposed model to act as a part of real-time control system.

CHAPTER 6

Automatic Feature Selection Using Genetic Algorithm

6.1 Introduction

The mutual information based feature selection followed in the previous chapter is computationally efficient, but it may not lead to an optimal feature subset selection, as the dependencies among the features are not taken into account. Further this procedure requires lot of human intervention in the form of selecting progressively increasing number of features and developing the networks with the selected features. During network development also, time consuming trial and error approach is to be followed to select the optimal network structure (number of basis functions in case of RBF networks). Though these operations are done off-line, the person responsible for development of these networks cannot spend too much time on these tasks. So, some sort of automatic mechanism is needed during the network development phase to make the neural network approach practicable for actual implementation.

This chapter proposes a genetic algorithm-based technique to select the input features of the network and the optimal number of basis functions to be used in the RBF network to approximate the given function.

6.2 Feature Selection Through Optimization

As defined in Section 4.2, the problem of feature selection is stated as the selection of a subset of size M from a given set of D variables, that leads to the smallest estimation error. For accurate and automatic feature selection, the subset selection problem is to be formulated as an optimization problem and some search technique is applied to select the optimal feature subset with respect to the criteria of interest. A number of authors have explored the use of *heuristics* for subset selection, often in conjunction with branch and bound search (See [54] for survey), a technique that is well known in combinatorial optimization. This approach can be applied to select the near-optimal subset. But it requires the criterion function used to evaluate a feature subset to be a monotonic function [80]. That is, the performance of a feature subset should improve whenever a feature is added to it. But in many practical pattern recognition problems, the monotonicity assumption is not satisfied.

Genetic algorithms, which belong to the class of randomized heuristic search techniques, offer an attractive approach to find near-optimal solutions to such optimization problems [101]. Feature subset selection techniques that employ genetic algorithms do not require the restrictive monotonicity assumption. They also readily lend themselves to the use of multiple selection criteria. Using genetic algorithm for feature selection for the design of neural networks involves running a genetic algorithm for several generations. In each generation, evaluation of a feature subset requires training the corresponding neural network and computing its accuracy in estimation. This evaluation has to be performed for each of the individuals in the population. This necessitates the use of fast learning methods to evaluate the feature subsets. In [101], the authors have used K-nearest neighbor classifier to evaluate the feature subsets. In this work we use radial basis function network for this task.

6.3 Feature Selection Using Genetic Algorithm

Genetic algorithms are search and optimization methods for solving problems by using principles inspired by natural genetics [45]. Unlike traditional 'hill-climbing' methods involving iterative changes to a single solution, GAs work with a population of solutions, which is 'evolved' in a manner analogous to natural selection. It works in binary search spaces, i.e., the search space consisting of binary strings, and is well suitable for combinatorial optimization problems. The working principle of GA is briefly outlined in Appendix D.

Fig 6.1. shows the schematic representation of the GA-based feature selection scheme. Initially, GA generates a population of feature subsets randomly. These candidate subsets are evaluated by training a radial basis function network with the training data set for a few iterations. If a fixed architecture is used in the RBF network, to evaluate the feature subsets, it may result in poor network training. For example, if the network has fewer basis functions than necessary, the learning

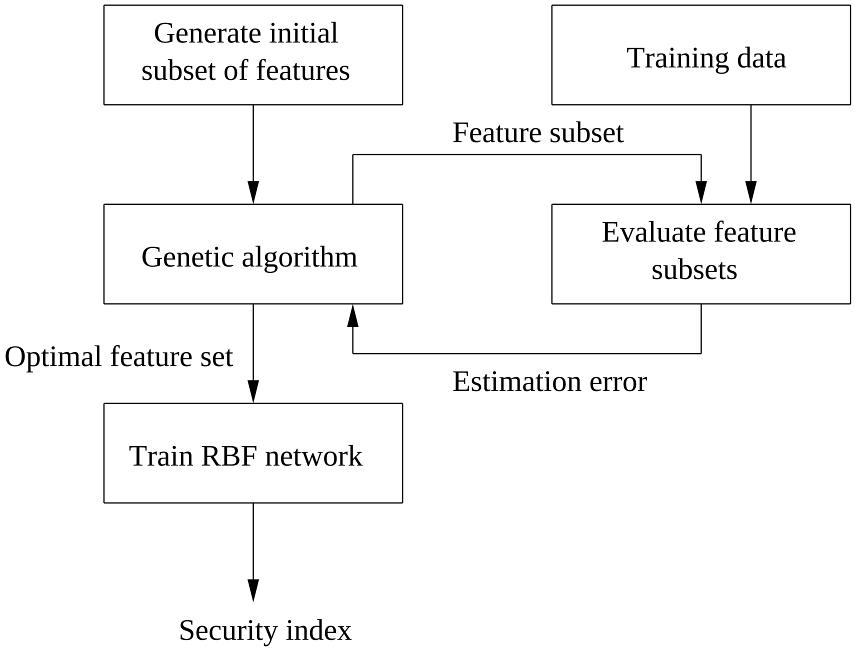

Fig. 6.1: GA-RBF network scheme for network development.

algorithm will fail to find the input-output mapping. If the network has more basis functions than necessary, it can result in over-fitting of the training data leading to poor generalization. In either case, it would make it difficult to evaluate the usefulness of a feature subset. So, the number of basis functions required to approximate the underlying relationship is also taken as a decision variable and evolved along with the feature subset. Thus each candidate solution in the GA population has a subset of input features and the number of basis functions to be used during the network simulation. The number of basis functions is determined randomly from a uniform distribution. The accuracy obtained in estimating the security index is then returned to the GA to calculate the fitness of each candidate solution. Then the candidate solutions are allowed to undergo the genetic operations, namely, reproduction, crossover and mutation. The whole procedure is repeated until the number of generations becomes greater than some prespecified constant. Finally, the input variables in the candidate solution with the highest fitness is considered as the input features and the number of basis functions used in that solution as the optimum value of basis function for network development.

Once the optimal feature subsets and the number of basis functions to be used to develop the radial basis function network is identified, the network can be trained further to reach a particular level of accuracy in security index estimation.

6.4 Implementation of Genetic Algorithm

While applying GA to a search problem, there are three important issues to be addressed. Firstly, how are the candidate solutions to the optimization problem represented, in order to allow the genetic algorithm to be easily applied? Secondly, how is the optimality of the solutions assessed? Finally, how is the genetic operators applied to existing solutions to yield new solutions? These issues are addressed in the following subsections.

6.4.1 Representation of Solution

Encoding the problem in terms of a genetic model is the most important step in the GA approach. Selection of a subset of a given set is perhaps one of the problems that is more amenable to the GA approach. For the feature selection problem considered here, there are N_l candidate variables representing power flows in all the N_l transmission lines. An individual in the GA population is represented by a string of N_l elements ("1" or "0"). The "1" in the i^{th} position of the string means that the i^{th} variable is a significant variable, while the "0" refers to insignificant variable. The insignificant variables are to be pruned out.

Each subset of features is evaluated according to its estimation accuracy on a set of training data using an RBF network. When dealing with a specific application, the performance of the RBF network mainly depends on the number of basis functions used in the hidden layer. So the number of basis functions required to approximate the underlying relationship is also taken as a decision variable and evolved along with the feature subset. This basis function number is represented as an integer here.

6.4.2 Evaluation Function

An evaluation function is required to assign a figure of merit (fitness) to each new solution, which should represent the quality of the solution that the individual represents. In the network development problem considered here, each individual is evaluated by training a radial basis function network with a training data set for a few iterations. The number of basis basis functions of the network is taken as equal to the integer quantity represented by the individual and the input features are the subset of features encoded by the individual. Based on the error in estimation and the dimension of the feature subset each individual is evaluated using the following fitness function (F):

$$F = \frac{1}{E} + W \cdot (D - M) \tag{6.1}$$

where D is the number of candidate variables, M is the number of input features selected, W is a weighting factor and E is the mean square error in estimation defined by

$$E = \frac{1}{N}\sum_{i=1}^{N}(o_i - d_i)^2 \tag{6.2}$$

In this expression, N is the number of training data, and o_i, d_i are the i^{th} obtained and desired outputs. GA searches for an individual that result in minimum estimation error with minimum number of input features.

6.4.3 Genetic Operators

Tournament selection along with *elitist strategy* is used to select the individuals to produce successive generations. Two point crossover is applied to generate the new offsprings. Since individuals in the population contain two different types of information (feature bit string and number of basis functions), mutation operator has to be applied little differently from the normally used mutation operator. For the feature bit string, the usual binary mutation operator is applied and each time the basis function is chosen to be mutated, it is increased or decreased by one.

6.5 Simulation Results

6.5.1 Test Case 1

The proposed GA-based method for network development was applied to develop the networks for security index estimation corresponding to the six single line outages (Table 3.1) identified as potentially severe cases in Chapter 3 for the IEEE 14-bus system. The input-output data generated for this system was used to train and test the networks.

Pre-contingency line flows in all the 20 lines are the candidate input variables of the networks. The security index corresponding to a contingency is the output of the network. The initial population of the GA was generated randomly. The parameters of the GA was set as follows:

Population size : 30

Crossover rate : 0.8

Mutation rate : 0.01

Size of tournament : 2

Elite copies : 2

These parameters were determined empirically through a set of preliminary experiments and they are held constant across all runs.

Each individual in the population represents a candidate solution to the feature selection problem. The candidate feature subsets are evaluated by training the radial basis function network for 5 iterations. The three step learning method explained in Section 5.2.2 is followed here also to train the networks. The GA is

Table 6.1: Result of GA-RBF network

S.No	Line outage (between buses)	Selected features ($S_i; i = 1 \cdots 20$)	No of basis functions	Testing error
1	1-2	1,4	17	$1.69 \cdot 10^{-5}$
2	2-3	2,6	28	$4.13 \cdot 10^{-6}$
3	2-4	2,5,7	28	$6.12 \cdot 10^{-6}$
4	1-5	1,3,5,11,12,16,20	25	$4.57 \cdot 10^{-5}$
5	4-5	1,2,3,7,9,10,16	28	$4.81 \cdot 10^{-5}$
6	5-6	8,9,10,17	22	$3.59 \cdot 10^{-4}$

run for ten generations to select the optimal feature subsets. The optimal subset of features along with number of basis functions used for all the six model are presented in Table 6.1. Once the input features and the number of basis functions are selected, the RBF networks are trained for 20 iterations with the training data. After training, the generalization capability of the network is evaluated using the test data set. The performance of the networks during testing phase is given in the last column of Table 6.1. Starting from 20 candidate variables the GA-based method is able to develop the networks with minimum number of features. This shows the effectiveness of the GA-based method for optimal feature selection.

The security index values estimated by the networks for two test cases are given in Table 6.2 and Table 6.3, with the ranking of the contingencies given in the parenthesis. For comparison, the actual values of security index, calculated from AC load flow study, are also given. The result shows the agreement between

Table 6.2: Security index estimation result-1

Line outage	Security index	
(between buses)	RBFN output	Load flow result
1-2	13.29 (1)	13.40 (1)
1-3	12.02 (2)	11.93 (2)
3-4	1.17 (6)	1.15 (6)
2-5	6.86 (3)	6.86 (3)
4-6	1.63 (5)	1.63 (5)
6-7	2.48 (4)	2.53 (4)

the actual ranking and the ranking based on the outputs of the RBF networks trained with the GA selected input features. This shows that the training of the RBF network is complete and correct estimation of the security index has been achieved by the RBF network, even for previously unseen data.

6.5.2 Test Case 2

After successfully applying the proposed GA-based method for network development in IEEE 14-bus system, it was decided to apply the same procedure to develop the networks for the nine contingencies listed in Table 3.2 for the IEEE 30-bus system. For each network there are 41 pre-contingency MVA flows available as candidate input variables of the networks. Some times during simulation the GA-based approach selected unnecessarily a few variables which represent the

Table 6.3: Security index estimation result-2

Line outage	Security index	
(between buses)	RBFN output	Load flow result
1-2	1.68 (3)	1.87 (3)
1-3	1.16 (4)	1.08 (4)
3-4	0.62 (5)	0.61 (5)
2-5	2.71 (2)	2.43 (2)
4-6	0.44 (6)	0.54 (6)
6-7	11.52 (1)	10.24 (1)

power flows in the lines far away from the location of the outage line. This resulted in poor performance of the networks during testing phase. Also with the increase in string length, GA took long time to converge. Hence, it was decided to follow a two step approach for network development in this case. In the first step, mutual information-based technique (filter approach) is applied to screen a specified number of variables from the original list. Then the GA-based approach (wrapper) is applied to identify the near optimal feature subsets from the screened variables. This is expected to result in accurate and faster search.

Table 6.4 shows the list of ten variables identified by the mutual information-based filter for the nine models. As expected, most of the screened variables represent the power flows in the lines in the vicinity of the outage line. Table 6.5 presents the final feature subsets along with the performance of the networks developed using these features. The selected features are not the same as the ones

Table 6.4: Selected variables after initial screening

Line outage (between buses)	Selected features (MVA flows)
1-2	1-2,1-3,2-4,3-4,2-5,2-6,4-6,5-7,6-7,9-11
1-3	1-2,1-3,2-4,3-4,2-5,2-6,4-6,5-7,6-7,9-11
3-4	1-2,1-3,2-4,3-4,2-5,2-6,4-6,5-7,6-7,9-11
2-5	1-2,1-3,3-4,2-5,2-6,4-6,5-7,6-7,6-9,9-11
4-6	1-2,1-3,2-4,3-4,2-5,2-6,4-6,5-7,6-7,9-11
6-7	1-2,1-3,3-4,2-5,2-6,4-6,5-7,6-7,6-9,9-11
9-10	6-9,6-10,9-11,9-10,4-12,12-14,12-15,12-16,16-17,10-21
19-20	1-3,2-4,3-4,2-6,9-10,12-15,15-18,18-19,19-20,10-20
28-27	15-23,22-24,23-24,25-27,28-27,27-29,27-30,29-30,8-28,6-28

selected by the mutual information-based feature selection in the previous chapter (Table 5.1). The effectiveness of the two stage approach for feature selection in large scale problems is evident from the low estimation error achieved by the networks during the testing phase.

6.6 Summary

The input features for neural network are generally selected by humans, who selects the features based on trial and error. Since the inner workings of such networks are so poorly understood, it is unlikely that the best features are actu-

Table 6.5: Performance of the networks with two stage feature selection

Line outage (between buses)	Selected features (Power flow between buses)	Testing error (mse)
1-2	1-2,1-3,3-4,4-6	3.38×10^{-6}
1-3	1-2,1-3,3-4,2-6,4-6	7.57×10^{-6}
3-4	1-2,3-4,2-6,4-6,6-7	8.97×10^{-6}
2-5	3-4,2-5,5-7	2.04×10^{-5}
4-6	1-3,2-4,3-4,2-6,4-6,6-7,9-11	2.26×10^{-4}
6-7	1-2,2-5,6-7,6-9,9-11	7.26×10^{-5}
9-10	6-10,9-10,12-14,12-15,16-17	3.36×10^{-4}
19-20	1-3,3-4,2-6,12-15,15-18,19-20	2.7×10^{-4}
28-27	22-24,25-27,28-27,27-29,29-30,8-28	5.6×10^{-4}

ally selected. Also, trial and error is needed in the optimal selection of network structure. The work presented in this chapter attempts to automate some of the process in network development such as feature selection and network structure selection through the combined use of genetic algorithm and radial basis function network. The resulting system improve the accuracy of the developed network and relieve the burden from the human.

To make the GA-based approach applicable for feature selection in large scale power system problems, a two stage approach, initial screening based on mutual information followed by GA for optimal feature selection is proposed.

CHAPTER 7

Security Enhancement by Optimal Power Flow

7.1 Introduction

With the contingency selection scheme built upon the security index defined by equation 3.1, a small set of "potentially harmful" contingencies is filtered out from the list of contingencies. After this screening process, remedial actions should be taken, to prevent the network overloads, should any one of those contingencies occur. There are two common strategies to avoid limit violations in the contingency state, *preventive control* and *post-contingency corrective action.* Under preventive-control strategy, actions are taken in the base case itself irrespective of the occurrence or non-occurrence of the contingencies. This makes the system to satisfy all the operational constraints even after the occurrence of a selected number of contingencies [3]. This is a conservative approach and it does not take advantage of the corrective capabilities of the network after the contingency has occurred [77]. These capabilities include generation rescheduling, phase shifting control, switching, etc. On the other hand, under corrective action strategy, control actions are not taken until the contingency actually occurs. But a *contingency* *plan* is prepared in advance, for the identified severe cases. This is a viable option for steady-state security, as limit violations can be tolerated for a reasonable period of time allowing corrective action to be taken. This work focuses on this later strategy only.

7.2 Role of Optimal Power Flow in Line Overload Alleviation

Transmission line overload can be alleviated by re-routing power flows in the system. A change in line flow can be caused by an appropriate change in phase angles and magnitudes of bus voltages, which are usually referred to as state variables. The state variables can in turn be modified by a variation in generated powers.

In [72,73], the linearized relationship between power flow in the overloaded lines and the generated power has been utilized to reschedule the power generation. By recognizing that transmission line overloads can impinge on the reactive-power balance, a computationally simple algorithm has been developed in [59]. Though these approaches are fast in rescheduling the power generations, it may lead to overloading in the other lines. For secure operation of the system without any limit violation, complete modeling of the system through load flow equations and operational constraints is necessary [77]. The mathematical formulation of the optimal power flow algorithm suitable for power system rescheduling is presented in the next section.

The other method of changing the line flow patterns in the system is through adjustment of phase angles of phase shifting transformer [107]. A phase shifting transformer can be represented by an admittance y in series with an ideal autotransformer having a complex turns ratio (k) as shown in Fig. 7.1.

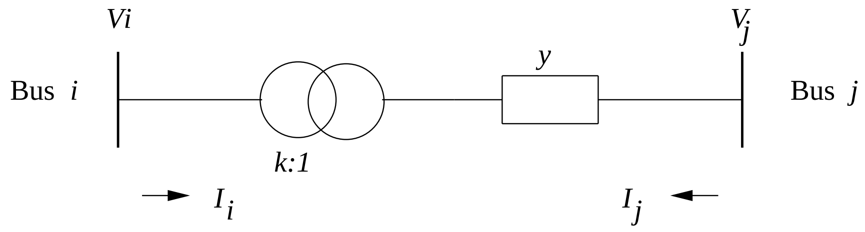

Fig. 7.1: Representation of phase shifting transformer.

The admittance matrix of the phase shifter is given by

$$\mathbf{Y} = \begin{bmatrix} \frac{y}{k^2} & \frac{y}{k^*} \\ \frac{-y}{k} & y \end{bmatrix}$$

For the complete derivation of the admittance matrix refer Dhar [27].

7.3 Mathematical Formulation of Optimal Power Flow Problem

The principal goal of an optimal power flow program is to provide the electric utilities with suggestions (set points) to optimize the current power system state with respect to various objectives such as minimization of production costs or system active power losses, maximizing the degree of security of a network or even combination of some of these. Optimal power flow program consists of three parts: The set of equality constraints representing the power system model for static computations, the set of inequality constraints representing real-world and practical operational constraints whose violation is not acceptable in the power system, and the objective function.

7.3.1 Objective Function

Optimal power flow algorithm uses an objective function and assigns a single numerical value to each possible solution, and the solution with the lowest value is the optimal solution. Typical objective functions used in OPF are expressions for the total fuel costs or expressions for transmission losses. For the first case, all system control variables are used in minimizing the objective and for the second case, only the control variables without direct costs, that is all except MW generation are considered for optimization.

In the more common cost minimization objective, the most critical factor is modeling of the input-output curves (cost/hr against MW) of the generating units. This not only has an effect on overall optimality, but is intimately bound with the solution method used. Usually, this curve is obtained from data points taken during “heat run” tests, when input and output data is measured as the unit is slowly varied through its operating range. Large steam turbine generators have a number of steam admission valves that are opened in sequence as the unit output is increased. Whenever an additional valve is first opened, the throttling losses increase rapidly and the incremental heat rate rises suddenly. A typical cost curve for a thermal unit with three valves is shown in Fig. 7.2 in dotted lines. The cost curve is very complex and the conventional gradient-based optimization algorithms cannot deal with cost curves of this form. Hence the cost curves are usually approximated and a convex, smooth cost curve (drawn in thick line) can be assumed to be the cost curve model of the generator. With approximated cost curve model, the fuel cost of the generating unit (F_i) is expressed by the following quadratic equation [128]

$$F_i = a_i P_{gi}^2 + b_i P_{gi} + c_i \ \$/\text{hr} \tag{7.1}$$

where P_{gi} is the real power generated by the ith generating unit and a_i, b_i and c_i are the fuel cost coefficients. The minimization of the total fuel cost (F_T) of all the N_g generators

$$F_T = \sum_{i=1}^{Ng} F_i \ \$/\text{hr} \tag{7.2}$$

to supply the given demand is the objective function of the optimal power flow problem.

During corrective control action, the prime task of the power system operator

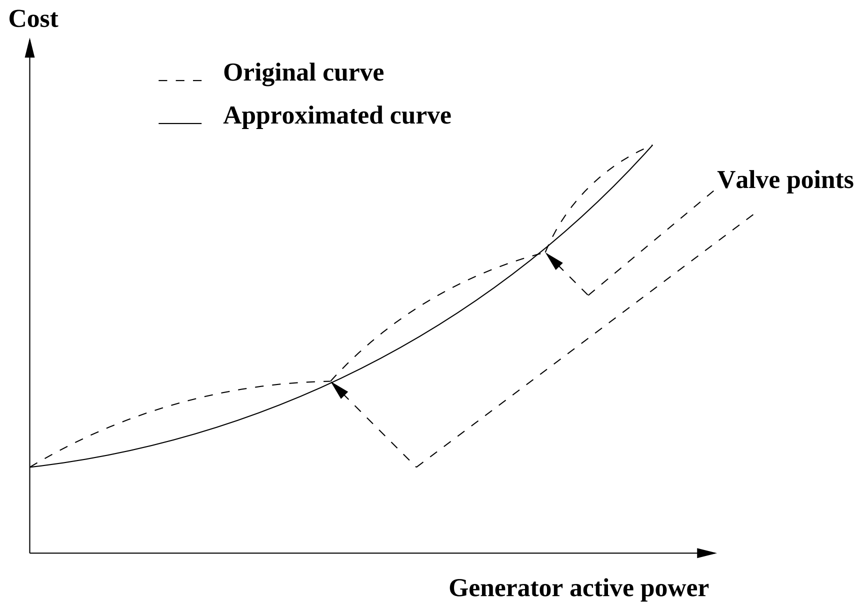

Fig. 7.2: A typical cost curve for thermal power generation

would be to remove the line overload. Hence, minimum line overload (F_o) can be selected as the objective function of the OPF problem when used for corrective control action.

$$F_o = \sum_{\in L_o} (S_l - S_l^{max}) \tag{7.3}$$

where S_l is the MVA flow in line l, S_l^{max} is the MVA rating of the line and L_o is the set of overloaded lines following a contingency.

Apart from the overload minimization objective considered above, the other important objective for corrective control calculation is the "minimum deviation from a specified point" [116]. This is defined as the weighted sum of the control variables from their target values.

$$f = \sum_{i=1}^{N_c} \alpha_i \cdot \Delta u_i \tag{7.4}$$

where N_c is the total number of control variables, Δu_i is the change in control variables from the initial point and α_i is the relative weighting of the ith control variable. This objective function heavily penalizes large excursions from the initial point, and thus perhaps from economic operation, as usually power systems are working under economic operating condition. It discourages the overloads from being relieved by large changes on a small number of units. This is a multiple objective problem comprising the weighted sum of dissimilar quantities. While using this objective function with a mix of control variable types like generator powers and voltages the relative weightings should be properly chosen, to avoid unnecessary scheduling of some of the control variables. For example, with the above objective, if generator powers (in MW) and voltages (in per-unit) are weighted equally, then the OPF program finds it just as expensive to schedule the power of controlled generator by 0.1 MW as to change its voltage by 0.1 per-unit. If the violation is largely MW branch overload, the result is an unnecessary rescheduling of voltages.

7.3.2 Constraints of the OPF Model

For reliable operation of the power system, it has to satisfy the equality constraint corresponding to the power flow model and a large number of operational and constraints. Mathematical formulation of these constraints is presented here.

(a) Power Flow Model as Equality Constraint Set of the OPF:

In a power system with N_b buses, at each bus i, the sum of the total injected real and reactive power and the specified power must be equal to 0. The specified power is equal to the difference between the power generation and the load. These constraints are mathematically represented as,

$$(P_{gi} - P_{di}) - V_i \sum_{j=1}^{N_b} V_j \left[G_{ij} cos\theta_{ij} + B_{ij} sin\theta_{ij}\right] = 0,\ i = 1, 2, \cdots, N_b \quad (7.5)$$

$$(Q_{gi} - Q_{di}) - V_i \sum_{j=1}^{N_b} V_j \left[G_{ij} sin\theta_{ij} - B_{ij} cos\theta_{ij}\right] = 0,\ i = 1, 2, \cdots, N_b \quad (7.6)$$

where,

P_{gi}, Q_{gi}	real and reactive power generated at bus i (pu)
P_{di}, Q_{di}	real and reactive power demand at bus i (pu)
V_i	voltage magnitude at bus i (pu)
θ_{ij}	voltage angle difference between buses i and j (rad)
G_{ii}, B_{ii}	self-conductance and susceptance of bus i (pu)
G_{ij}, B_{ij}	mutual conductance and susceptance between buses i and j (pu)

The set of equations formed by Equation 7.5 for all system buses except the slack bus and Equation 7.6 for all load buses constitute the power flow equation. The power flow equations determine the steady-state conditions of the power system network for a specified generation and load pattern. Calculates voltages, phase angles, and flows across the entire system. When solving the power flow equations iteratively, in successive solutions one will have a mismatch between the specified and the injected power, so Equation 7.5 and 7.6 will not be satisfied. Hence, a tolerance is specified for the power flow solutions.

(b) Lower and upper voltage magnitude limits:

Too high or too low voltage magnitudes could cause problems to the end user power apparatus or instability in the power system. The voltage magnitude constraint is expressed as,

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{7.7}$$

(c) Lower and upper generator active power limits:

The active power of a generator is physically limited in each generator.

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max} \tag{7.8}$$

(d) Lower and upper generator reactive power limits:

The reactive power of a generator is an important measure of voltage magnitude quality, e.g. a low voltage indicates a local shortage of reactive power.

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \tag{7.9}$$

The upper and lower reactive power limits are often not given as numeric values but as functions of the active power generator.

(e) Maximum line MVA limit:

The maximum MVA values for transmission lines and transformers are given due to limitations of the branch material. Excessive power would damage the transmission elements.

$$S_l \leq S_l^{max} \tag{7.10}$$

(f) Lower and upper transformer phase angle limit:

These limits come from the fact that the phase angle variation of phase shifting transformer is limited by the physical construction of each transformer.

$$\phi_i^{min} \leq \phi_i \leq \phi_i^{max} \tag{7.11}$$

7.3.3 Classification of OPF Variables

The variables associated with OPF problem can be separated into categories: Control variables ($\mathbf{u}$) and state variables ($\mathbf{x}$). The control variables correspond to quantities that can be arbitrarily manipulated, within their limits, in order to minimize an objective function. These include generator real power outputs, generator bus terminal voltages, transformer tap ratio and transformer phase shift angles. The state variables correspond to quantities that are set as a result of the control variables, but must be monitored. The state variables include load bus voltage magnitudes and phase angles of all the buses.

7.3.4 Summary of OPF Problem

In summary, the optimal power flow problem can be stated as the minimization of the objective function given by Equation 7.2 or any other objective function subject to the constraints (7.7) to (7.11). In terms of the control and state variable representation, the OPF problem can be written in compact form as follows:

$$\text{Minimize} \quad f(\mathbf{x}, \mathbf{u}) \tag{7.12}$$

$$\text{subject to} \quad \mathbf{g}(\mathbf{x}, \mathbf{u}) = 0, \tag{7.13}$$

$$\mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0 \tag{7.14}$$

where f($\mathbf{x}$, $\mathbf{u}$) is the objective function. Equation 7.13 represents the *load constraints* of the system, and Equation 7.14 represents the *operational constraints* of the system, which impose maximum or minimum operating limits on variables associated with the component parts of the system. The former are usually treated as *hard*, and the latter may be regarded as *soft*.

7.3.5 Contingency Constraints

For secure operation, in addition to satisfying the equality and inequality constraints given by the equations (7.5-7.10) in the base case, the system should satisfy these constraints in the post-contingency state also. Thus for each outage case k, the following additional constraints are formulated.

$$\mathbf{g}^{(k)}(\mathbf{x}^k, \mathbf{u}^k) = 0; \text{for } k = 1, 2, \cdots, N_s \tag{7.15}$$

$$\mathbf{h}^{(k)}(\mathbf{x}^k, \mathbf{u}^k) \geq 0; \text{for } k = 1, 2, \cdots, N_s \tag{7.16}$$

where N_s is the number of contingencies selected by the contingency selection algorithm. Equations 7.15 and 7.16 represent the load constraints and operational constraints respectively, under contingency k.

With the inclusion of contingency constraints, the optimal power flow problem becomes a multiple network problem, comprised of the base case network and each contingency network. To obtain an optimal solution these individual network problems must be formulated as a multiple network problem and solved in an integrated fashion. The main reason for this integrated approach is that any base case control action will affect all contingency states. In the special case of *corrective control action* considered here, base case control actions are absent and hence the multiple network problem becomes a set of single network problems one for each contingency [9], which can be solved by the regular optimal power flow program.

7.4 Solving the OPF Problem for Corrective Control

Corrective control calculation can be viewed as a search which systematically explores the control options, maintaining the principle of optimality throughout until a feasible set of actions is found. Network overload can be alleviated by re-routing power flows in the system. This involves the rescheduling of power system controls via the automatic generation control. Where present, phase shifter angles can also be controlled. A phase shifter regulates the flow of active power by varying its phase angle ϕ [27].

The control variables associated with generator output power and generator terminal voltages are continuous. For example, the constraint $P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max}$ allows P_{gi} to take on all the values within the specified range. But the variables associated with phase shifting transformer are discrete. The discrete variables cannot be incorporated directly into the general optimal power flow programs. One possible solution for this problem is to round the optimal setting found assuming continuous variable. However, three problems arise from this approach. First, there is no guarantee that the rounded solution is the optimal solution. Second, the solution may become infeasible after rounding, i.e., some constraints may be violated. Finally, this methodology will not work properly for discrete variables that have very large step sizes such as switched capacitor banks.

The next section presents a GA-based technique to solve the OPF problem, in which the discrete variables of the phase shifting transformer can be directly included along with the other continuous type variables.

7.5 Genetic Algorithm Implementation for OPF Problem

Optimal power flow for corrective control action is a mixed integer non-linear optimization problem. Conventional calculus-based methods have difficulties in determining the global solution of the optimal power flow problem. Also difficulties are encountered in incorporating the discrete variables related to the phase shifting transformers. A random search technique like genetic algorithm [45] can be applied to solve the optimal power flow problem. Fig 7.3 shows the flow chart of the GA-based algorithm for solving the optimal power flow problem.

The traditional binary-coded genetic algorithm has number of difficulties in dealing with continuous search spaces, as explained in Section 2.5.1.2. We propose a hybrid genetic algorithm approach to solve the optimal power flow problem. The implementation details of this algorithm are presented below:

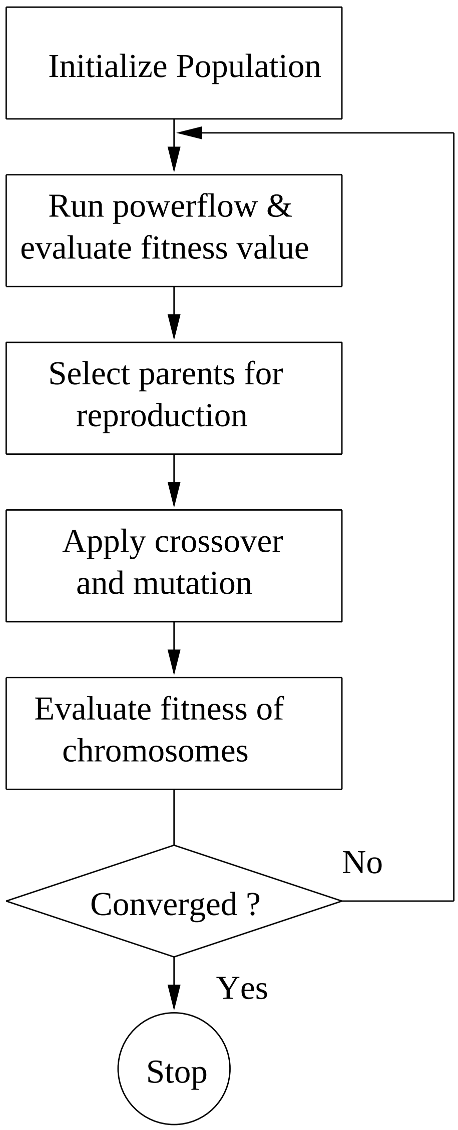

Fig. 7.3: Flow chart of GA-OPF algorithm

7.5.1 Problem representation

Each individual in the population represents a candidate OPF solution. We consider the generator active power generation, generator terminal voltage and phase angle of the phase shifting transformer as control variables of the OPF problem. We thus have $2N_g + N_{ph}$ control variables, where N_g is the number of generators in the system and N_{ph} is the number of phase shifting transformers available for phase angle control. Out of these, the real power generation at the slack bus can be calculated only after knowing the transmission losses in the system. The remaining $(2N_g + N_{ph} - 1)$ variables are the independent variables of the system and they form the chromosome of the GA population. These variables are represented in their natural form i.e. generator active power P_{gi} and generator terminal voltages V_{gi} are represented as real numbers whereas, the phase angle of the phase shifting transformer, being a discrete quantity is represented as an integer. With this representation, a typical chromosome of the OPF problem looks like the following.

$$\underbrace{97.5}_{P_{g2}}\underbrace{110.8}_{P_{g3}}\ldots\underbrace{300.8}_{P_{gn}}\underbrace{0.975}_{V_{g1}}\underbrace{1.017}_{V_{g2}}\ldots\underbrace{0.980}_{V_{gn}}\underbrace{-2}_{\phi_1}\underbrace{3}_{\phi_2}$$

The use of floating point numbers in the GA representation has a number of advantages over binary encoding. The efficiency of the GA is increased as there is no need to convert the solution variables to the binary type. Moreover, less memory is required and there is no loss in precision by discretization to binary or other values. Also, there is greater freedom to use different genetic operators.

7.5.2 Initialization of the population

For any GA, it is necessary to initialize the population. The most common method is to randomly generate solutions for the entire population. For the OPF problem under consideration, each of the individual is initialized randomly using a uniform random number distribution within the feasible range of the control variables.

7.5.3 Evaluation function

GA searches for the optimal solution by maximizing a given fitness function, and therefore an evaluation function which provides a measure of the quality of the problem solution must be provided. In the OPF problem under consideration, the objective is to minimize the total fuel cost (7.1) satisfying the constraints (7.5 - 7.10).

7.5.3.1 Calculation of Fuel Cost

As represented by equation 7.1, the cost of power generation is a function of individual active power generations. The active power generations at the generator buses are taken from the GA solution as they are the independent variables. To obtain the power generation at the slack bus generator, equations 7.5 and 7.6 are solved through a power flow algorithm using the control variable set provided by the GA. Once the active power generations of all the generators are known, the total fuel cost can be evaluated.

7.5.3.2 Treatment of Constraints

The limits on control variables are implicitly satisfied by the problem representation itself. However, the computed real power at the slack bus, reactive power at generator buses, the voltage magnitude at load buses, and finally the computed power flows in all the transmission lines are not guaranteed to satisfy the operating limits. These constraints on state variables are handled through penalty functions method [23]. An overall cost function, consisting of the original cost function plus additional penalty costs for any violated constraints is then optimized. In general, penalty functions should be chosen to satisfy the following requirements [106]:

- The penalty function should be progressive, so that a more severe violation of a constraint attracts a higher penalty cost - allowing the GA to be 'guided' towards feasibility.
- The penalty factors for each violated constraint should be summed to form the overall penalty cost, so that a well-behaved penalty cost surface is produced.

With the inclusion of penalty terms, the overall objective function becomes,

$$\begin{aligned} \min F = \sum_{i=1}^{Ng} F_i \quad &+ \quad K_s \mid P_{gs} - P_{gs}^{limit} \mid + K_v \sum_{j=1}^{N_l} \mid V_j - V_j^{limit} \mid \\ &+ \quad K_q \sum_{j=1}^{N_g} \mid Q_j - Q_j^{limit} \mid + K_f \sum_{l=1}^{N_l} \mid S_l \mid - S_l^{max} \end{aligned} \tag{7.17}$$

where P_{gs} is the slack bus generation and K_s, K_v, K_q and K_f are the penalty factors for the slack bus active power limit violation, voltage bus voltage limit violation, generator reactive power limit violation and line flow rating violation, respectively. In the above equation, x_i^{limit} of a quantity (x_i) is defined by the following expression:

$$x_i^{limit} = \begin{cases} x_i^{max}, & \text{if } x_i > x_i^{max} \\ x_i^{min}, & \text{if } x_i < x_i^{min} \end{cases} \tag{7.18}$$

The first term on the right hand side of Equation 7.17 corresponds to the objective function value and the remaining terms are penalty functions for the different state variables. For the candidate solutions with no limit violation, the penalty function values become zero and the value of cost function is equal to the objective function value. On the other hand, if there is any violation in the lower or upper limit of the state variables, a corresponding penalty value, based on the amount of violation is added to the objective function.

The success of the penalty function approach lies in the proper choice of the penalty factors. Using the above penalty function approach, one has to experiment to find a correct combination of penalty factors K_s, K_v and K_q. However, in order to reduce the number of penalty parameters, the constraints are normalized and only one penalty factor is used.

In conventional mathematical programming the use of penalty costs can lead to an ill-conditioned overall problem [23]. While applying penalty values that are large enough to guarantee feasible solutions the overall cost function surface may get distorted, which are then problematic for a mathematical programming method. This drawback of the penalty factor approach is less significant for GAs, since they do not impose any particular requirements on the mathematical properties of overall cost function surface.

During the GA run, GA searches for a solution with maximum fitness function value. Hence, the minimization objective function given by Equation 7.17 is transformed to a fitness function (f) to be maximized as,

$$f = \frac{k}{F} \tag{7.19}$$

where k is a large constant. This is used to amplify $(1/F)$, the value of which is usually small, so that the fitness values of the chromosome will be in a wider range.

7.5.4 Genetic Operators

With hybrid representation, the evaluation procedure and reproduction operator remain the same as in the binary-coded GA, but crossover operation is done separately for real and integer variables. Simple crossover explained in Appendix D is applied for the integer part. The blend crossover operator (BLX-α) [38] based on the theory of interval schemata is employed in this study for real variables. For the values $u_i^{(j)}$ and $u_i^{(k)}$ of the variable u_i in the parents j and k, this operator creates a new point uniformly at random from a range extending an amount $\alpha \mid u_i^{(j)} - u_i^{(k)} \mid$ on either side of the region bounded by two parents.

Fig 7.4 shows the BLX-α crossover operation for the one dimensional case. In this figure, u_1 and u_2 represent two parents from a particular variable. The values of e_1 and e_2 are given by the expressions:

$$e_1 = u_1 - \alpha \times (u_2 - u_1) \tag{7.20}$$

$$e_2 = u_2 - \alpha \times (u_2 - u_1) \tag{7.21}$$

It is to be noted that e_1 and e_2 must be between u_{min} and u_{max}, the variable's lower and upper bounds respectively.

In a number of test problems, Eshelman and Schaffer [38] observed that $\alpha = 0.5$

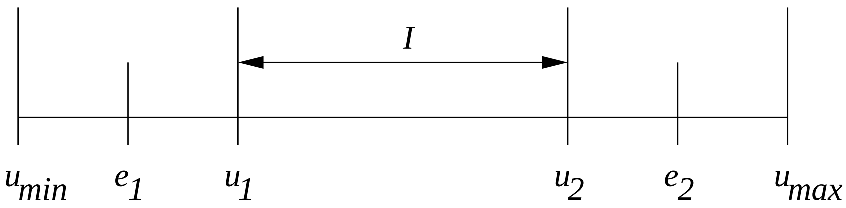

Fig. 7.4: BLX-α crossover operator.

provides good results. One interesting feature of this type of crossover operator is that the created point depends on the location of both parents. If both parents are close to each other, the new point will also be close to the parents. On the other hand, if parents are far from each other, the search is more like a random search.

The mutation operator is used to inject new genetic material into the population. Mutation randomly alters a variable with a small probability. "Uniform mutation" operator is applied to the real variables. It randomly selects one of the variables, u_i, from a parent and sets it equal to an uniform random number between the variable's lower (u_{min}) and upper (u_{max}) bounds. For the integer part, the randomly generated floating point number is truncated to the nearest integer.

7.6 Simulation Results

The proposed GA-based approach for solving OPF problem is tested on IEEE 30-bus system (Fig. 3.2). The network consists of 6 generator buses, 21 load-buses and 41 branches. The line parameters and the base load of this system are given in Appendix B. Four cases were considered: (a) Optimal scheduling of the power system under normal state, (b) Generation rescheduling to alleviate line overloads under base load condition, (c) Rescheduling with multiple-objective function and (d) Line overload alleviation with phase shifter control. The GA-based algorithm was implemented using Matlab code, and the program was executed on a Pentium III processor computer.

7.6.1 Case 1: Optimal scheduling for the base case

In this case the contingencies are not considered and the GA-based algorithm was applied to find the optimal scheduling of the power system for the base case loading condition given in [3]. Generator active power outputs and the generator bus terminal voltages were taken as the decision variables. The objective function is the minimization of the total fuel cost. The unit cost curves are represented by quadratic functions. The generator active and reactive power limits and generator cost coefficients are listed in Table B.1.

The initial population of the GA was randomly generated between the variable's lower and upper limits. The population size was taken to be 30. The performance of a GA is generally dependent on the GA parameters used, in par-

Table 7.1: Summary of the results with P_c=0.9 and P_m=0.01.

Minimum cost (over ten runs)	802.83 $/hr
Average cost (over ten runs)	804.64 $/hr
CPU time (for one run)	30 sec

ticular, the crossover and mutation probabilities, P_c and P_m, respectively. The performance of GA for various crossover and mutation probabilities in the range of 0.6-1.0 and 0.001-0.1, respectively, was therefore evaluated. Due to the probabilistic nature of the GA, ten independent runs of GA were made. The total number of generations in each run was fixed as 25. The best results of the GA was obtained with P_c=0.9 and P_m=0.01 and they are summarized in Table 7.1. The minimum cost obtained with the GA is near to the minimum cost of 802.43 $/hr, reported in [3] for the same case using gradient method. The optimal values of decision variables along with the real power generation of slack bus generator and the reactive power generation of all the generating units are provided in Table 7.2, for the best GA run. From this table it is evident that the power generation of all the generators are well within their limit. Also, it was found that all the state variables satisfy the lower and upper limits.

Fig 7.5 shows the variation of the fuel cost during the GA run for the best case. It can be seen that the fuel cost reduces rapidly in the first 10 generations of the GA. During this stage, the GA mainly concentrates on finding feasible solutions to the problem. Then the value decreases slowly, and settles down near the optimum

Table 7.2: Summary of the base case solution

Bus	Generation		Voltage
No.	Real	Reactive	magnitude
1	177.40	-11.11	1.048
2	48.78	23.91	1.033
5	21.32	24.92	1.001
8	20.38	21.61	1.006
11	13.01	17.83	1.060
13	12.19	32.23	1.075

value with most of the individuals in the population reaching that point. It shows that the GA converges well in the application of optimal scheduling problem.

For comparison, a binary-coded GA was applied to solve the same problem. Ten bits were used to encode each generator active power output and six bits were used to encode the voltages, with a total string length of 86 bits for an individual in the GA population. Tournament selection along with two-point crossover and bit mutation were applied during the GA run. The algorithm took an average of 65 seconds to reach the optimal point. This is more than twice the time taken by the real-coded GA to reach the optimal point. It shows that real-coded GA is faster in convergence than the binary-coded version for the problems with continuous search space.

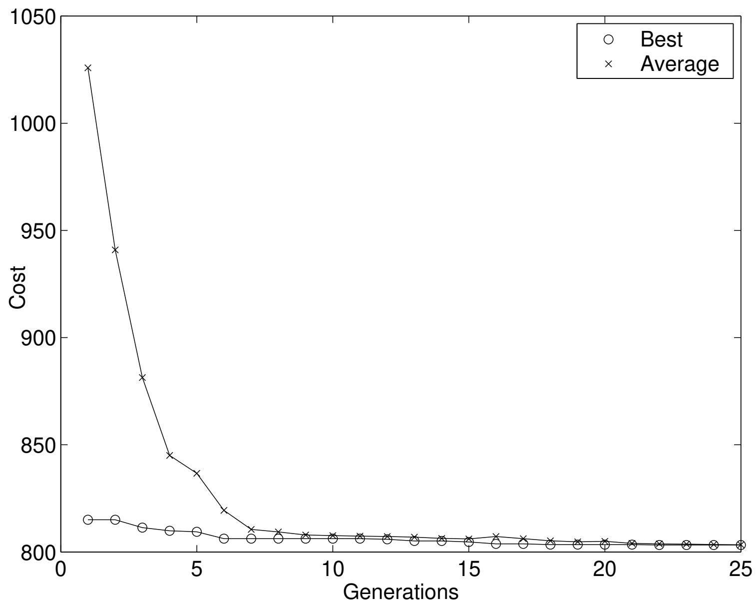

Fig. 7.5: Convergence of the GA-OPF algorithm

7.6.2 Case 2: Generation Rescheduling for Overload Alleviation

In this case, the GA-based algorithm was applied to reschedule some of the power system controls to alleviate line overloads in the post-contingency state. The single line-outages under the base case loading condition given in [3] for the IEEE 30-bus system is considered here for discussion. Table 7.3 gives the summary of contingency analysis under base case condition. From this table it is found that line outages 1-2, 1-3, 3-4 and 2-5 leads to line overloading in the post-contingency period, and they need corrective control action to alleviate the line overload.

The GA-based OPF algorithm was applied to alleviate the line overload with generator active power output as the control variable. The control parameter setting which gave the best results in Case 1 was used here also. Minimum overload

Table 7.3: Summary of contingency analysis for the base load

Outage line	Overloaded lines	Line flow (MVA)	Line flow limit (MVA)
1-2	1-3	191.30	130
	3-4	182.49	130
	4-6	110.13	90
1-3	1-2	143.66	130
3-4	1-2	180.93	130
	2-6	65.98	65
2-5	2-6	77.30	65
	5-7	81.28	70
4-6	Nil	–	–
6-7	Nil	–	–
9-10	Nil	–	–
19-20	Nil	–	–
27-28	Nil	–	–

Table 7.4: Line flows after rescheduling

Outage line	Overloaded lines	Line flow (MVA)		Line rating (MVA)
		Before	After	
1-2	1-3	191.30	129.84	130
	3-4	182.49	121.42	130
	4-6	110.13	76.67	90
1-3	1-2	143.66	123.88	130
	1-2	180.93	125.68	130
3-4	2-6	65.98	56.67	65
	2-6	77.30	64.96	65
2-5	5-7	81.28	68.70	70

was taken as the objective function of the GA. The GA-based algorithm converged in the first few generations itself in all the four cases. Table 7.4 presents the power flows before and after the rescheduling operation in the overloaded lines for all the four selected contingencies. From this table it is evident that the overloading of the transmission lines has been completely alleviated with the rescheduling of generator active power output alone. But, this may not be possible in all the cases, especially with higher loading conditions, in which one has to look for other control options like phase shifter control etc.

7.6.3 Case 3: Rescheduling with Multiple-objective Function

The main task of the power system operator during corrective control action would be to quickly move the power system to a normal operating state without considering the economic operation for the moment. To achieve this, the rescheduling problem is formulated as a multi-objective optimization problem with both minimum violation as well as the minimum deviation of the control variables from the initial point as the objectives. The GA-based algorithm was applied to this multi-objective problem with active power generation as the decision variable as in the previous case. The optimal values of the active power generation obtained in this case along with the results obtained in Case 2 (minimum violation objective) are given in Table 7.5 for outages 1-2 and 2-5. From this table, it is found that the overload has been alleviated with minimum control variable movement in Case 3, as compared to Case 2. Overload alleviation with minimum control movement will help the power system operator to restore the system to the normal state in a short period of time.

7.6.4 Case 4: Overload Alleviation With Phase Shifter Control

In this case, real and reactive load at all the buses were increased to 1.25 times the base case value. For this loading condition, conventional linear programming-based optimal power flow program failed to converge for line outages 1-2 and 2-5 with generator active power and generator terminal voltages as control variables with cost minimization objective. The performance of the GA-based optimal pow-

Table 7.5: Comparison of control variable deviation

Bus No.	Outage 1-2		Outage 2-5	
	Case 2	Case 3	Case 2	Case 3
1	-65.24	-61.44	-24.82	-1.76
2	+15.68	+24.03	-19.28	-27.62
5	+16.16	+8.57	+18.57	+12.88
8	+9.08	+7.61	+8.5	+1.39
11	+1.93	+8.38	+4.75	+4.83
13	+9.52	+0.72	+6.14	+6.47
Total deviation	117.59	110.75	82.05	54.96

Table 7.6: Optimal solution with phase shifter control

Variables	P_1	P_2	P_5	P_8	P_{11}	P_{13}	V_1
	172.32	48.53	49.94	34.50	29.28	36.50	1.047
Variables	V_2	V_5	V_8	V_{11}	V_{13}	ϕ_{1-2}	ϕ_{1-3}
	1.039	0.951	0.997	1.089	1.046	0	-4

er flow algorithm to search the control actions for line-outage 2-5 is presented here.

The GA-based algorithm was applied to alleviate the line overloads with generator active power, generator terminal voltages and phase angle of the phase shifting transformer as the control variables. The phase shifters were installed between buses 1-2 and 1-3. The limits of the phase shifters were taken as $\pm 10°$. The GA-based algorithm took around 100 seconds to converge to the feasible solution. The control variables corresponding to this feasible solution are given in Table 7.6. The result shows the effectiveness of the GA-based algorithm to solve the difficult optimization problems.

7.7 Summary

This chapter has proposed a flexible optimization tool using genetic algorithm to aid the power system operators. As the implementation of this algorithm is simple it can be easily modified to make it suitable for different applications. The application of this tool for scheduling the power system during normal operation and to schedule the power system controls during contingencies has been demonstrated through a number of test cases with different objective functions.

The problem of discretization in the representation of the generator loadings and generator bus voltage magnitudes in the binary-coded GA has been alleviated by employing floating point numbers to represent the generator loadings and generator bus voltage magnitude. A modified form of crossover and mutation operations to deal with the real variables has been presented. Compared to conventional GA, the proposed algorithm occupies less computer space and takes less CPU time and is well suitable for large-scale optimization problems.

CHAPTER 8

Summary and Conclusion

8.1 Summary of the Thesis

In this thesis, the problem of static-security analysis has been studied. The objectives are (a) To identify the issues in real time security analysis, (b) To address these issues using computational intelligent tools. The aim is to develop a neural network-based model for automatic contingency selection using the training data generated through off-line simulation process and to develop an operator assistance tool for security enhancement using genetic algorithm.

Contingency selection through analytical technique takes long time, as a load flow study has to be run for each contingency. An Artificial Neural Network (ANN)-based contingency selection technique has been proposed in this thesis for real time security assessment. One-contingency-one-network approach which dedicates separate networks to deal with each contingency is followed during network development. The networks are developed based on the input-output data generated through off-line contingency analysis. The developed networks can give the output result fastly, corresponding to the input. Radial Basis Function (RBF) network has been proposed to learn the input-output mapping. RBF network takes less time for training and it has the intrinsic ability to indicate when it is extrapolating.

A major task in the ANN-based approach is the identification of the proper input features of the network. In this thesis, the feature selection problem has been addressed through two approaches. In the first approach, the mutual information between the input variables and the output quantity is used as the criterion to select the input features. This approach with no-feedback from the learning algorithm used to train the neural network is computationally less complex but requires lot of human interaction. In the second approach, genetic algorithm in combination with radial basis function network is used to select the input features of the network as well as to select the structure of the network. Though this approach is computationally more complex than the previous approach, it needs less human interaction and it can select the near-optimal feature set.

A Genetic Algorithm (GA)-based optimal power flow algorithm has been proposed to prepare contingency plans for the selected contingencies. The problem of discretisation in the representation of the control variables in the conventional genetic approach has been alleviated by employing floating point numbers to encode the control variables. The proposed GA-based approach is more effective in finding the optimal solution than the calculus-based methods under contingency conditions. Also, the GA-based approach is capable of obtaining the *optimal infeasible* solution which is especially appropriate for OPF problems in the contingency state. Hence, the proposed GA-based algorithm can be used as an operator assistance tool to prevent the eventualities arising out of contingencies.

8.2 Contributions of this Work

The major contributions of this work are highlighted below:

1. Formulating the contingency selection problem as a supervised learning problem, and developing neural network-based model to address this task.

2. Development of security information database to study and develop the networks for contingency selection.

3. Proposing radial basis function networks to learn the relationship between the pre-contingency operating state and the post-contingency security index.

4. Proposing mutual information between the pre-contingency operating state and the post-contingency security index to select the input features of the radial basis function network.

5. Use of genetic algorithm to automatically select the input features of the network as well as to choose the number of basis functions to be used to map the input-output relationship.

6. Application of real-coded genetic algorithm to solve the optimal power flow problem in the normal and contingency states.

8.3 Conclusions of the Thesis

The following are some important conclusions of the study

1. Security index estimation using the proposed approach is accurate and fast. Therefore, the neural network-based model presented in this thesis can be used for fast contingency ranking for eventual use in control centers.

2. As the training time of radial basis function networks is shorter, it is well suited for power system applications, as one may have to retrain the networks whenever major topological changes occur in the system.

3. The genetic algorithm-based technique can be used for network design, as it automates some of the processes in network development like feature selection and selecting the optimal structure of the network.

4. For large scale applications, the mutual information-based algorithm can be used as a first stage feature selector, from which the optimal feature set can be searched by the GA-based algorithm.

5. The proposed GA-based technique can be used for optimal contingency planning, as it can handle multiple objectives in the corrective scheduling problem.

8.4 Directions for Future Work

The present work has considered the assessment and enhancement of steady-state security only. For a comprehensive security analysis, other phenomena like voltage collapse and transient instability arising out of contingencies can also be included in the security analysis package.

The developed neural network-based model requires retraining of the networks whenever major structural changes occurs in the system. To avoid that, networks with topological generalization capability should be explored.

The proposed genetic algorithm-based method although faster than the conventional genetic algorithm, still takes lot of time for convergence. To speed up the convergence of the algorithm, local search methods like gradient information can be used during the genetic algorithm run.

The present work has considered the security enhancement through re-scheduling of generator outputs only. For line overload alleviation, other means like control of Flexible AC Transmission System (FACTS) devices can be explored.

APPENDIX A

Test System 1: IEEE 14-bus System

Table A.1: Generator data

Bus No.	Voltage magnitude	Reactive power limits	
		Minimum	Maximum
2	1.045	-40	50
3	1.010	0	40
6	1.070	-6	24
8	1.090	-6	24

Table A.2: Transformer data

Between buses	Tap setting
4-7	0.978
4-9	0.969
5-6	0.932

Table A.3: Impedance and line-charging data

Line No.	Line	R p.u.	X p.u.	B (total) p.u.	Rating MVA
1	1-2	0.01938	0.05917	0.02640	250
2	2-3	0.04699	0.19797	0.02190	150
3	2-4	0.05811	0.17632	0.01870	100
4	1-5	0.05403	0.22304	0.02460	250
5	2-5	0.05695	0.17388	0.01700	75
6	3-4	0.06701	0.17103	0.01730	100
7	4-5	0.01335	0.04211	0.00640	150
8	5-6	0.00000	0.25202	0.0000	100
9	4-7	0.00000	0.20912	0.0000	75
10	7-8	0.00000	0.17615	0.0000	50
11	4-9	0.00000	0.55618	0.0000	50
12	7-9	0.00000	0.11001	0.0000	75
13	9-10	0.03181	0.08450	0.0000	40
14	6-11	0.09498	0.19890	0.0000	40
15	6-12	0.12291	0.25581	0.0000	40
16	6-13	0.06615	0.13027	0.0000	40
17	9-14	0.12711	0.27038	0.0000	30
18	10-11	0.08205	0.19207	0.0000	30
19	12-13	0.22092	0.19988	0.0000	30
20	13-14	0.17093	0.34802	0.0000	30

Table A.4: Load Data

Bus No.	Load	
	Real power (MW)	Reactive power (MVAR)
1	0.0	0.0
2	21.7	12.7
3	94.2	19.0
4	47.8	-3.90
5	7.6	1.6
6	11.2	7.5
7	0.0	0.0
8	0.0	0.0
9	29.5	16.6
10	9.0	5.8
11	3.5	1.8
12	6.1	1.6
13	13.5	5.8
14	14.9	5.0

APPENDIX B

Test System 2: IEEE 30-bus System

Table B.1: Generator data

Bus No	P_g^{min} (MW)	P_g^{max} (MW)	Q_g^{min} (MVAR)	S_g^{max} (MVA)	cost coefficients		
					a	b	c
1	50	200	-20	250	0.0	2.00	0.00375
2	20	80	-20	100	0.0	1.75	0.01750
5	15	50	-15	80	0.0	1.00	0.06250
8	10	35	-15	60	0.0	3.25	0.00834
11	10	30	-10	50	0.0	3.00	0.02500
13	12	40	-15	60	0.0	3.00	0.02500

Table B.2: Branch data

Line No.	Line	R p.u.	X p.u.	B (total) p.u.	Rating MVA
1	1-2	0.0192	0.0575	0.0264	130
2	1-3	0.0452	0.1852	0.0204	130
3	2-4	0.0570	0.1737	0.0184	65
4	3-4	0.0132	0.0379	0.0042	130
5	2-5	0.0472	0.1983	0.0209	130
6	2-6	0.0581	0.1763	0.0187	65
7	4-6	0.0119	0.0414	0.0045	90
8	5-7	0.0460	0.1160	0.0102	70
9	6-7	0.0267	0.0820	0.0085	130
10	6-8	0.0120	0.0420	0.0045	32
11	6-9	0.0	0.2080	0.0	65
12	6-10	0.0	0.5560	0.0	32
13	9-11	0.0	0.2080	0.0	65
14	9-10	0.0	0.1100	0.0	65
15	4-12	0.0	0.2560	0.0	65
16	12-13	0.0	0.1400	0.0	65
17	12-14	0.1231	0.2559	0.0	32
18	12-15	0.0662	0.1304	0.0	32
19	12-16	0.0945	0.1987	0.0	32
20	14-15	0.2210	0.1997	0.0	16

Line No.	Line	R p.u.	X p.u.	B (total) p.u.	Rating MVA
21	16-17	0.0824	0.1932	0.0	16
22	15-18	0.1070	0.2185	0.0	16
23	18-19	0.0639	0.1292	0.0	16
24	19-20	0.0340	0.0680	0.0	32
25	10-20	0.0936	0.2090	0.0	32
26	10-17	0.0324	0.0845	0.0	32
27	10-21	0.0348	0.0749	0.0	32
28	10-22	0.0727	0.1499	0.0	32
29	21-22	0.0116	0.0236	0.0	32
30	15-23	0.1000	0.2020	0.0	16
31	22-24	0.1150	0.1790	0.0	16
32	23-24	0.1320	0.2700	0.0	16
33	24-25	0.1885	0.3292	0.0	16
34	25-26	0.2544	0.3800	0.0	16
35	25-27	0.1093	0.2087	0.0	16
36	28-27	0.0	0.3960	0.0	65
37	27-29	0.2198	0.4153	0.0	16
38	27-30	0.3202	0.6027	0.0	16
39	29-30	0.2399	0.4533	0.0	16
40	8-28	0.0636	0.2000	0.0214	32
41	6-28	0.0169	0.0599	0.0065	32

Table B.3: Load data

Bus No.	Load	
	Real power (MW)	Reactive power (MVAR)
1	0.0	0.0
2	21.7	12.7
3	2.4	1.2
4	7.6	1.6
5	94.2	19.0
6	0.0	0.0
7	22.8	10.9
8	30.0	30.0
9	0.0	0.0
10	5.8	2.0
11	0.0	0.0
12	11.2	7.5
13	0.0	0.0
14	6.2	1.6
15	8.2	2.5

Bus No.	Load	
	Real power (MW)	Reactive power (MVAR)
16	3.5	1.8
17	9.0	5.8
18	3.2	0.9
19	9.5	3.4
20	2.2	0.7
21	17.5	11.2
22	0.0	0.0
23	3.2	1.6
24	8.7	6.7
25	0.0	0.0
26	3.5	2.3
27	0.0	0.0
28	0.0	0.0
29	2.4	0.9
30	10.6	1.9

APPENDIX C

Feedforward Neural Networks with Backpropagation Learning

C.1 Architecture

Neural networks have a variety of architectures, but the feedforward network trained by backpropagation algorithm, commonly called backpropagation network has proven to be the most popularly used so far in real industrial problems. This is because they perform better over a larger set of different problems, which has been proved by a number of research papers. The architecture of this network consists of one input layer together with one output layer that represent the system inputs and outputs respectively, and one or more hidden layers providing the learning capability for the network. Fig C.1. shows the structure of the feedforward neural network with one hidden layer. The basic element of the network is an artificial neuron, which perform a differentiable nonlinear activation function on the weighted sum of the input . Fig C.2 and C.3. depicts two of the most

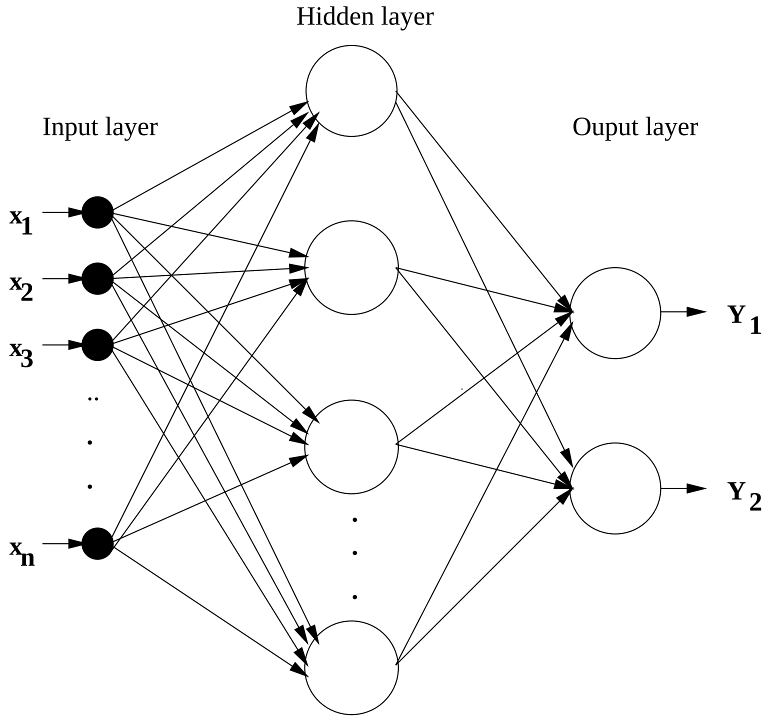

Fig. C.1: Three layer feedforward neural network

commonly used activation functions in feedforward neural networks:

$$\text{Logistic function:} f(x) = \frac{1}{1+e^{-x}} \tag{C.1}$$

$$\text{Hyperbolic tangent function:} f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{C.2}$$

Both the hyperbolic tangent and logistic functions approximate the signum and step function, respectively, and yet provide smooth, nonzero derivatives with respect to input signals. Sometimes these two activation functions are referred to as **squashing functions** since the inputs to these functions are squashed to the range [0,1] or [-1,1]. They are also called **sigmoidal functions** because their S-shaped curves exhibit smoothness and asymptotic properties. Both of these

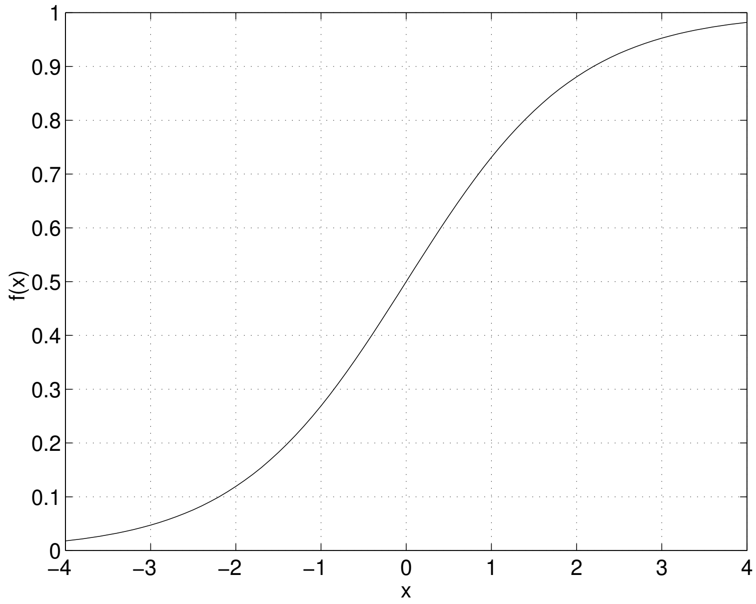

Fig. C.2: Logistic sigmoidal function.

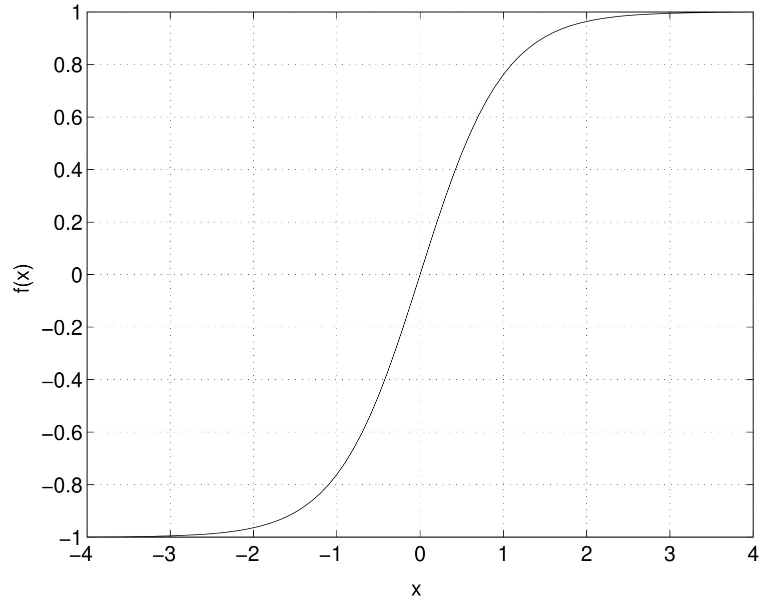

Fig. C.3: Hyperbolic sigmoidal function.

activations are used often on regression and classification problems. For a neural network to approximate a continuous-valued function not limited to the interval [0,1] or [-1,1], usually the node function for the output layer will be a weighted sum with no squashing functions. This is equivalent to a situation in which the activation function is an identity function, and output nodes of this type are often called **linear nodes**.

The parameters of this network are the weight matrix W_1, containing weights $w_{i,j}$ that connect the neuron i to the input j, and the weight matrix W_2, and the bias vector θ. The estimation of these parameters is called the "training" of the network, and is done by the minimization of a loss function.

C.2 Learning in Feedforward Neural Networks

Backpropagation learning algorithm is one of the most frequently used methods in training multilayer feedforward networks. The error, which is the difference between the actual output (t_k) and the network response (O_k) is calculated and propagated backwards from the output to the hidden layer and then to the input. This is performed by minimizing the error function:

$$E = \sum_p E_p = \frac{1}{2} \sum_p \sum_k (t_k^{(p)} - O_k^{(p)})^2 \tag{C.3}$$

Each weight is adjusted according to the rule:

$$\Delta w_{ij} = -k \frac{dE}{dw_{ij}} \tag{C.4}$$

where k is a constant of proportionality, and w_{ij} represents the weights of the connection between neuron j and neuron i. The process is repeated until the difference between all output nodes and all patterns are within some acceptable tolerance.

Before training begins, the network architecture (i.e. number of hidden neurons) must be established. No definitive rule exists for specifying the network architecture. The number of hidden neurons determine a network's learning capabilities. A network with too few hidden neurons will not be capable of accurately modeling the input-output relationship. On the other hand, too many hidden neurons may force the network to memorize the training data. The network may perform poorly when applied to different data. Also, the hidden-layer size affects the training time. A trial-and-error procedure is followed to select the optimum number of hidden neurons.

Backpropagation networks have been applied to many pattern recognition problems including the classification of sonar targets, speech recognition, and fault diagnosis. These applications have shown that backpropagation network have the ability to learn a set of training examples and can often generalize successfully to new cases also.

APPENDIX D

Working Principle of Genetic Algorithm

In the genetic algorithm a solution, i.e., a point in the search space, is represented by a finite sequence of zeros's and one's, called a *chromosome*. Fig. Fig D.1 presents the pseudo-code of the working of a simple GA. First, a population of strings representing the decision variables is randomly generated. The size of the population depends on the string length and the problem being optimized. Each string is then evaluated using the objective function value. In GA terminology, the objective function value of a string is known as the "fitness" of the string. This evaluation function acts as a pseudo objective function, since it is a raw measure of the solution value. For constrained optimization problems, the evaluation function typically comprises a weighted sum of the objective and penalty functions to consider the constraints. This approach allows constraints to be violated, but a penalty depending on the magnitude of the violation is incurred. A highly infeasible individual has a high penalty value and will rarely be selected for next generation, allowing the GA to concentrate on feasible or near-feasible solutions. Once the strings are evaluated, the three genetic operators - reproduction, crossover and mutation are applied to create a new population.

Fig. D.1: Pseudo-code of working of GAs

```
Initialize a population of strings at random
Evaluate each string in the population
repeat
        Reproduction
        Crossover
        Mutation
        Evaluation of the population
until (termination criterion)
```

D.1 Reproduction

The selection of individuals to produce successive generations plays an important role GA. Reproduction comprises forming a new population, usually with the same total number of chromosome, by selecting from members of the current population, following a particular scheme. The higher the fitness, the more likely it is that the chromosome will be selected for the next generation. There are several strategies for selecting the individuals (e.g) roulette wheel selection, ranking methods and tournament selection. Here we use tournament selection.

In tournament selection, n individuals are selected in random from the population, and the best of the n is inserted into the new population for further genetic processing. This procedure is repeated until the mating pool is filled. Tournaments are often held between pairs of individuals ($n = 2$), although larger tournaments can be used.

Though this scheme selects good individuals for the next generation, it cannot guarantee that the best solution obtained survives throughout the optimization process. In other words, the best solution obtained from the GA may not be included in the final solutions, which are clustered towards a solution. To avoid this, the GA is modified such that once an individual with highest fitness among the current generation is found, it will be kept unchanged into the next generation. This process is called "elitism".

D.2 Crossover operation

After the reproduction phase is over, the population is enriched with good strings. Reproduction makes copies of good strings, but does not create any new string. A crossover operator is used to recombine the strings with the hope of creating a better string. An overall crossover probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur. This probability is often in the range of 0.6 to 0.9. For the binary-coded GA, there exist a number of crossover operators in the literature, but in almost all crossover operators, two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings.

The single point crossover operator randomly selects cut point. The two parents are randomly split at this point, and children are created by concatenating segments from both parents. For example, the cut point generated below is 3, creating a split within the parents (P1 and P2) after the third bit.

P1=1 0 1 | 0 1 0

P2=0 0 1 | 1 0 0

The following two children, C1 and C2 are created from these two parents

C1=1 0 1 | 1 0 0

C2=0 0 1 | 0 1 0

D.3 Mutation

Mutation acts as a background operator and is used to search the unexplored search space by randomly changing the values at one or more positions of the selected chromosome. This is often carried out with a constant probability between 0.001 to 0.01 for each element in the population. An example mutation operator for binary genetic codes is one that randomly changes the genetic codes from 1's to 0's and vice versa.

After new strings are created they are evaluated by calculating their fitness function values. This completes a cycle of genetic algorithm. Such iterations are continued until a termination criterion is satisfied.

REFERENCES

[1] Aggoune M., El-Sharkawi M.A., Park D.C., Damborg M.J. and Marks II R.J., "Preliminary results on using artificial neural network for security assessment", IEEE Transactions on Power Systems, Vol. 6, No. 2, pp. 890-896, May 1991.

[2] Albuyeh F., Bose A. and Heath B., "Reactive power considerations in automatic contingency selection", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 1, pp. 107-112, Jan. 1982.

[3] Alsac O. and Scott B., "Optimal load flow with steady state security", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-93, pp. 745-751, 1974.

[4] Alsac O., Bright J., Prais M. and Scott B., "Further developments on LP-based optimal power flow", IEEE Transactions on Power Systems, Vol. 5, No. 3, pp. 697-711, Aug. 1990.

[5] Arora C.M. and Surana S.L., "Transient security evaluation and preventive control of power systems using PR techniques", Institute of Engineers (India) Journal, Vol, 76, pp. 199-203, Feb. 1996.

[6] Babuska R., "Fuzzy modeling for control", Kluwer Academic Publishers, Boston, 1998.

[7] Bacher R., in "Optimization in planning and operation of Electric Power Systems", Physica-Verlag, Heidelberg, 1993.

[8] Back T., Hammel U. and Schwefel H.P., "Evolutionary computation: Comments on the history and current state", IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp. 3-17, April 1997.

[9] Balu N.J. *et al.*, "On-line power system security analysis", Proceedings of the IEEE, Vol. 80, No. 2, pp. 262-280, Feb. 1992.

[10] Battiti R., "Using mutual information for selecting features in supervised neural net learning", IEEE Transactions on Neural Networks, Vol. 5, No. 4, pp. 537-550, July 1994.

[11] Bertram T.J., Demaree K.D. and Dangelmaier L.C., "An integrated package for real-time security enhancement", IEEE Transactions on Power Systems, Vol. 5, No. 2, pp. 592-600, May 1990.

[12] Bishop C.M., "Neural networks for pattern recognition", Oxford University Press, London, 1995.

[13] Brandwajn V., "Efficient bounding method for linear contingency analysis", IEEE Transactions on Power Systems, Vol. 3, No. 1, pp. 38-43, Feb. 1988

[14] Brandwajn V. and Lauby M.G., "Complete bounding method for AC contingency screening", IEEE Transactions on Power Systems, Vol. 4, No. 2, pp. 724-729, May 1989.

[15] Burchett R.C., Happ H.H. and Vierath D.R., "Quadratic converged optimal power flow", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 11, pp. 3267-3275, Nov. 1985.

[16] Chang C.Y., "Dynamic programming as applied to feature subset selection in a pattern recognition system", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 3, pp. 166-171, March 1973.

[17] Chang C.S., Chung T.S. and Lo K.L., "Application of pattern recognition technique to power system security analysis and optimization", IEEE Transactions on Power Systems, Vol. 5, No. 3, pp. 835-841, Aug. 1990.

[18] Chauhan S. and Dave M.P., "Input feature selection for ANN based voltage stability assessment", Proceedings of the Eleventh national power systems conference, Bangalore, India, Vol. 2, pp. 433-437, Dec. 2000.

[19] Chen S., Cowan C.F.N. and Grant P.M., "Orthogonal least squares learning algorithm for radial basis function networks", IEEE Transactions on Neural Networks, Vol. 2, No. 2, pp. 302-309, March 1991.

[20] Dabbaghchi I., Christie R.D., Rosenwald G.W. and Liu C.C., "AI application areas in power systems", IEEE Expert, pp. 58-66, Jan./Feb. 1997.

[21] Davis L., "Handbook of genetic algorithms", Van Nostrand Reinhold, New York, 1991.

[22] Deb K. and Agrawal R.B., "Simulated binary crossover for continuous search space", Complex Systems, Vol. 9, pp. 115-148, 1995.

[23] Deb K, "Optimization for engineering design", Prentice-Hall of India, New Delhi, 2000.

[24] Demuth H. and Beale H.M., *Neural Network Toolbox for Use with Matlab*, Version 3, Mathworks, 1998.

[25] Devaraj D. and Yegnanarayana B., "A combined genetic algorithm approach for optimal power flow", Proceedings of the Eleventh national power systems conference, Bangalore, India, Vol. 2, pp. 524-528, Dec. 2000.

[26] Devaraj D., Yegnanarayana B. and Ramar K., "Radial basis function networks for fast contingency ranking", Accepted for publication in Electric power and energy systems Journal.

[27] Dhar R.N., "Computer aided power system operation and analysis", Tata McGraw-Hill Publishing Company Limited, New Delhi, 1982.

[28] Djukanovic M., Calovic M., Milosevic B. and Sobajic D.J., "Neural-net based real-time economic dispatch for thermal power plants", IEEE Transactions on Energy Conversion, Vol. 11, No. 4, pp. 755-761, 1996.

[29] Drezga I. and Rahman S., "Input variable selection for ANN-based short-term load forecasting", IEEE Transactions on Power Systems, Vol. 13, No. 4, pp. 1238-1244, Nov. 1998.

[30] Dommel H.W. and Tinney W.F., "Optimal power flow solutions", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87, No. 10, pp. 1866-1876, Oct. 1968.

[31] Dong Z.Y., Wang Y., Hill D.J. and Makarov Y.V., "A new approach to power system VAR planning aimed at voltage stability enhancement with feedback control", Proceedings of the Power Tech'99 Conference, Budapest, Hungary, 1999.

[32] Duda R.O. and Hart P.E., "Pattern classification and scene analysis", John Wiley & Sons, New York, 1973.

[33] Dy Liacco T.E., "Real-time computer control of power systems", Proceedings of the IEEE, Vol. 62, No. 7, pp. 884-891, July 1974.

[34] Dy Liacco T.E., "System security:the computer's role", IEEE Spectrum, pp. 43-50, June 1978.

[35] Ejebe G.C. and Wollenberg B.F., "Automatic contingency selection for AC contingency screening", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-98, No. 1, pp. 97-109, Jan./Feb. 1979.

[36] Ejebe G.C., Van Meeteran H.P. and Wollenberg B.F., "Fast contingency screening and evaluation for voltage security analysis", IEEE Transactions on Power Systems, Vol. 4, No. 4, pp. 1582-1590, Nov. 1988.

[37] El-Sharkawi M.A. and Marks R.J., "What role neural networks play in power system engineering", IEEE Power Engineering Review, pp. 14-16, Feb. 1994.

[38] Eshelman L.J. and Schaffer J.D., "Real-coded genetic algorithms and interval schemata", In D. Whitley, Edition. pp. 187-202, 1993.

[39] Fink L.H. and Carlsen K., "Operating under stress and strain", IEEE Spectrum, pp. 48-53, March 1978.

[40] Fogel D.B., "An introduction to simulated evolutionary optimization", IEEE Transactions on Neural Networks, Vol. 5, No. 1, pp. 3-14, Jan. 1994.

[41] Fogel D.B., Fukuda T. and Guan L. Edition, "Computational Intelligence", Proceedings of the IEEE, Vol. 87, No. 9, Sep. 1999.

[42] Fu Y. and Chung T.S., "A hybrid artificial neural network and ward equivalent approach for on-line power system voltage security assessment", Electric Power Systems Research, Vol. 53, pp. 165-171, 2000.

[43] Galina F.D., "Bound estimates of the severity of line outages in power system contingency analysis and ranking", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 9, pp. 2612-2624, Sep. 1984.

[44] Ghosh S. and Chowdhury B.H., "Design of artificial neural network for fast line flow contingency ranking", Electrical Power & Energy, Vol. 18, No. 5, pp. 271-277, 1996.

[45] Goldberg D., "Genetic algorithms in search, optimization and machine learning", Addison-Wesley, 1989.

[46] Grainer J.J. and Stevenson W.D., "Power System Analysis", McGraw-Hill, Inc. 1994.

[47] Haykin S.,"Neural networks - A comprehensive foundation", Prentice Hall International. Inc, New Jersey, 1999.

[48] Hippert H.S., Pedreira C.E. and Souza R.C., "Neural networks for short-term load forecasting: A review and evaluation", IEEE Transactions on Power Systems, Vol. 16, No. 1, pp. 44-55, Feb. 2001.

[49] Holland J.H., "Genetic algorithms", Scientific American, July 1992.

[50] Hornik K., Stinchcombe M. and White H., "Multilayer feedforward networks are universal approximator", Neural Network, Vol. 2, pp. 359-366, 1989.

[51] Hsu Y.Y. and Kuo H.C., "Fuzzy-set based contingency ranking", IEEE Transactions on Power Systems, Vol. 7, No. 3, pp. 1189-1196, Aug. 1992.

[52] Irisarri G., Levner D. and Sasson A.M., "Automatic contingency selection for on-line security analysis - real-time tests", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-98, No. 5, pp. 1552-1559, Sep./Oct. 1979.

[53] Jain L.C. and Jain R.K., "Hybrid intelligent engineering systems", World Scientific, Singapore, 1997.

[54] Jain A.K., Robert Duin P.W. and Mao L., "Statistical pattern recognition: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4-37, Jan. 2000.

[55] Jang J.S.R., Sun C.T. and Mijutani E., "Neuro-Fuzzy and Soft computing", Prentice Hall, Englewood Cliffs, New Jersey, 1997.

[56] Kukolj D., Popovic D., Kulic F. and Gorecan Z., "Fast dynamic stability analysis of a power system using artificial neural networks", European Transactions on Electric Power, Vol. 8, No. 3, pp. 207-212, May/June 1998.

[57] Kundur P. and Wang L., "Power system dynamic security: on-line assessment and control", Proceedings of the Eleventh national power systems conference, Bangalore, India, Vol. 2, pp. 399-404, Dec. 2000.

[58] Kusic G.L., "Computer-aided power systems analysis", Prentice Hall of India Private Limited, New Delhi, 1989.

[59] Lachs W.R., "Transmission-line overloads:real-time control", IEE Proceedings, Vol. 134, Pt. C, No. 5, pp. 342-347, Sep. 1987.

[60] Lai L.L., Ma J.T., Yokoyama R. and Zhao M., "Improved genetic algorithms for optimal power flow uuunder normal and contingent operation states", Electric Power and Energy Systems, Vol. 19, No. 5, pp. 287-292, 1997.

[61] Lai L.L., "Intelligent system applications in power engineering", John Wiley & Sons, England, 1998.

[62] Lauby M.G., Mikolinnas T.A. and Reppen N.D., "Contingency selection of branch outages causing voltage problems", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-102, No. 12, pp. 3899-3904, Dec. 1983.

[63] Lauby M.G., "Evaluation of a local DC load flow screening method for branch contingency selection of overloads", IEEE Transactions on Power Systems, Vol. 3, No. 3, pp. 923, 1988.

[64] Lee K.Y., Park Y.M. and Ortiz J.L., "A united approach to optimal real and reactive power dispatch", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 5, pp. 1147-1153, May 1985.

[65] Leonard J.A. and Kramer M.A., "Radial basis function networks for classifying process faults", IEEE Control Systems, pp. 31-38, April 1991.

[66] Leonard J.A. and Kramer M.A., "Diagnosing dynamic faults using modular neural nets", IEEE Expert, pp. 44-53, April 1993.

[67] Lin C.T. and Lee C.S.G., "Neural fuzzy systems", Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[68] Lo K.L., Bismil M.A., McColl R.D. and Moffatt A.M., "A comparison of MW ranking methods", Electric Power Systems Research, Vol. 15, pp. 157-171, 1988.

[69] Lo K.L., Bismil M.A., McColl R.D. and Moffatt A.M., "A comparison of voltage ranking methods", Electric Power Systems Research, Vol. 16, pp. 127-140, 1989.

[70] Lo K.L., Peng L.J., Macqueen J.F., Ekwue. and Cheng D.T.Y., "Fast real power contingency ranking using a counterpropagation network' , IEEE Transactions on Power Systems, Vol. 13, No. 4, pp. 1259-1263, Nov. 1998.

[71] Matos M.A., Hatziargyriou N.D. and Pecas Lopes J.A., "Multicontingency steady state security evaluation using fuzzy clustering techniques", IEEE Transactions on Power Systems, Vol. 15, No. 1, pp. 177-182, Feb. 2000.

[72] Medicherla T.K.P., Billington R. and Sachdev M.S., "Generation rescheduling and load shedding to alleviate line overloads"-analysis, IEEE Transaction on Power Apparatus and Systems, Vol. PAS-98; No. 12, pp. 1876-1884, 1979.

[73] Medicherla T.K.P., Billington R. and Sachdev M.S., "Generation rescheduling and load shedding to alleviate line overloads"-system studies, IEEE Transaction on Power Apparatus and Systems, Vol. PAS-100; No. 1, pp. 36-42, 1981.

[74] Michalewicz Z. and Schoenauer M., "Evolutionary algorithms for constrained parameter optimization problems", Evolutionary Computation, Vol. 4, No. 1, pp. 1-32.

[75] Moechtar M., Farag A.S., Hu L. and Cheng T.C., "Combined genetic algorithms and neural network approach for power system transient stability evaluation", European Transaction on Electric Power, Vol. 9, No. 2, pp. 115-122, March/April 1999.

[76] Moller M.F., "A scaled conjugate gradient algorithm for fast supervised learning", Neural Networks, Vol. 6, pp.525-533, 1993.

[77] Monticelli A., Pereira M.V.F. and Granville S., "Security-constrained optimal power flow with post-contingency corrective rescheduling", IEEE Transactions on Power Systems, Vol. 2, No. 1, pp. 175-182, Feb. 1987.

[78] Moody J. and Darken C., "Fast learning in networks of locally-tuned processing units", Neural Computation, Vol. 1, pp. 281-294, 1989.

[79] Nahman J. and Skokljev I., "Fuzzy logic and probability based real time contingency ranking", Electrical Power and Energy Systems, Vol. 22, pp. 223-229, 2000.

[80] Narendra P.M. and Fukunaga K., "A branch and bound algorithm for feature subset selection", IEEE Transactions on Computers, Vol. 26, pp. 917-922, 1977.

[81] Neeharika A., "Generalization capability of feedforward neural networks for pattern recognition tasks", M.S. Thesis, M.S. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, 1996.

[82] Niebur D. and Germond A.J., "Power system static security assessment using the Kohonen neural network classifier", IEEE Transactions on Power Systems, Vol. 7, No. 2, pp. 865-872, May 1992.

[83] Obaidat M.S., "Artificial neural networks to systems, man, and cybernetics: Characteristics, structures, and applications", Editorial, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 28, No. 4, pp. 489-495, Aug. 1998.

[84] Pai M.A., "Computer techniques in power system analysis", Tata McGraw-Hill Publishing Company Limited, New Delhi, 1979.

[85] Pang C.K., Koivo A.J. and El-Abiad A.H., "Application of pattern recognition to steady-state security evaluation in power system", IEEE Transactions on System Man and Cybernaticts, pp. 87-99, Nov. 1973.

[86] Pang C.K., Prabhakara F.S., El-Abiad A.H. and Koivo A.J., "Security evaluation in power systems using pattern recognition", IEEE Transactions on Power Apparatus and Systems, pp. 969-976, 1974.

[87] Popovic D., Kukolj D. and Kulic F., "Monitoring and assessment of voltage stability margins using artificial neural networks with reduced input set", IEE Proceedings-Generation, Transmission and Distribution, Vol. 145, No. 4, pp. 355-362, July 1998.

[88] Raghu P.P., "Artificial neural network models for texture analysis", Ph.D. thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, 1995.

[89] Ray D., Chakravorti S. and Mukherjee P.K., "An application of artificial neural network for power system security evaluation", Institute of Engineers Journal (India), Vol. 75, pp. 131-135, 1994.

[90] Raymer M.L., Punch W.F., Goodman E.D., Kuhn L.A. and Jain A.K., "Dimensionality reduction using genetic algorithms", IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, pp. 164-171, July 2000.

[91] Refaee J.A., Mohandes M. and Maghrabi H., "Radial basis function networks for contingency analysis of bulk power systems", IEEE Transactions on Power Systems, Vol. 14, No. 2, pp. 772-778, May 1999.

[92] Reza F.M., "An introduction to information theory", McGraw-Hill, Inc, 1961.

[93] Sarkar M., "Uncertainty-based pattern classification by modular neural networks", Ph.D. thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, 1998.

[94] Sarkar M., Yegnanarayana B. and Khemani D., "A clustering algorithm using an evolutionary programming-based approach", Pattern Recognition Letters, Vol. 18, No. 10, pp. 975-986, 1997.

[95] Schmidt H.P., "Application of artificial neural networks to the dynamic analysis of the voltage stability problem", IEE Proceedings-Generation, Transmission and Distribution, Vol. 144, No. 4, pp. 371-376, July 1997.

[96] Schafer K.F. and Verstege J.F., "Adaptive procedure for masking effect compensation in contingency selection algorithm", IEEE Transactions on Power Systems, Vol. 5, No. 2, pp. 539-546, May 1990.

[97] Schafer K.F., Schwartze C. and Verstege J.F., "CONTEX: an expert system for contingency selection", Electric Power Systems Research, Vol. 22, pp. 189-194, 1991.

[98] Schnyder G. and Glavitsch H., "Security enhancement using an optimal switching power flow", IEEE Transactions on Power Systems, Vol. 5, No. 2, pp. 674-681, May 1990.

[99] Se-Young Oh, "A pattern recognition and associative memory approach to power system security assessment", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-16, No. 1, pp. 62-72, Jan./Feb. 1986.

[100] Sidhu T.S. and Cui L., "Contingency screening for steady-state security analysis by using FFT and artificial neural networks", IEEE Transactions on Power Systems, Vol. 15, No. 1, pp. 421-426, Feb. 2000.

[101] Siedlecki W. and Sklansky J., "A note on genetic algorithms for large-scale feature selection", Pattern Recognition Letters Vol. 10, pp. 335-347, Nov. 1989.

[102] Singh S.N., Srivastava L. and Sharma J., "Fast voltage contingency screening and ranking using cascade neural network", Electric Power Systems Research, Vol. 53, pp. 197-205, 2000.

[103] Sinha A.K. and Nagrath I.J., "Pattern recognition method for power system steady-state security assessment", Institute of Engineers Journal (India), Vol. 64, pp. 269-273, April 1984.

[104] Sobajic D.J. and Pao Y.H., "An artificial intelligence system for power system contingency screening", IEEE Transactions on Power Systems, Vol. 3, No. 2, pp. 647-653, May 1988.

[105] Sobajic D.J. and Pao Y.H., “Artificial neural net based dynamic security assessment for electric power systems”, IEEE Transactions on Power Systems, Vol. 4, No. 1, pp. 220-228, Feb. 1989.

[106] Song Y.H. Edition, “Modern optimization techniques in power systems”, Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.

[107] Srinivasan N., Prakasa K.S. and Venkata S.S., “On-line computation of phase shifter distribution factors and line overload alleviation”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 7, pp. 1656-1662, 1985.

[108] Srivastava L., Singh S.N. and Sharma J., “Parallel self-organizing hierarchical neural-network-based fast voltage estimation”, IEE Proceedings-Generation, Transmission and Distribution, Vol. 145, No. 1, pp. 98-104, Jan. 1998.

[109] Srivastava L., Singh S.N. and Sharma J., “Knowledge-based neural network for voltage contingency selection and ranking”, IEE Proceedings-Generation, Transmission and Distribution, Vol. 146, No. 6, pp. 649-655, Nov. 1999.

[110] Srivastava L., Singh S.N. and Sharma J., “Modular neural network-based fast voltage contingency selection”, Institute of Engineers Journal (India), Vol. 80, pp. 118-125, Nov. 1999.

[111] Srivastava L., Singh S.N. and Sharma J., “A hybrid neural network model for fast voltage contingency screening and ranking”, Electrical Power and Energy Systems, Vol. 22, pp. 35-41, 2000.

[112] Srivastava L., Singh S.N. and Sharma J., “Comparison of feature selection techniques for ANN-based voltage estimation”, Electric Power Systems Research, Vol. 53, pp. 187-195, 2000.

[113] Srivastava L., Singh S.N. and Sharma J., “Fast line flows estimation using parallel self-organizing hierarchical neural network”, Proceedings of the Eleventh national power systems conference, Bangalore, India, Vol. 2, pp. 449-454, Dec. 2000.

[114] Stagg G.W. and El-Abiad A.H., “Computer methods in power systems analysis”, McGraw-Hill Book Co-Singapore, 1968.

[115] Stott B. and Alsac O., “Fast decoupled load flow”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-93, pp. 859-869, May/June 1974.

[116] Stott B. and Hobson E., "Power system security control calculations using linear programming", Part-I and II, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-97, No. 5, pp. 1713-1731, Sep./Oct. 1978.

[117] Stott B., Alsac O. and Monticelli A.J., "Security analysis and optimization", Proceedings of the IEEE, Vol. 75, No. 12, pp. 1623-1644, Dec. 1987.

[118] Sun D.I., Ashley B., Brewer B., Hughes A. and Tinney W.F., "Optimal power flow by Newton approach", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 10, pp. 2864-2880, Oct. 1984.

[119] Testa A., Menniti D., Picardi C. and Sorrentino N., "Steady-state security prediction in presence of load uncertainty", European Transaction on Electric Power, Vol. 8, No. 2, pp. 97-104, MarchApril 1998.

[120] Walters D.C. and Sheble C., "Genetic algorithm solution of economic dispatch with valve point loading", IEEE Transactions on Power Systems, Vol. 8, No. 3, pp. 1325-1332, Aug. 1993.

[121] Wan H.B. and Ekwue A.O., "Artificial neural network based contingency ranking method for voltage collapse", Electrical Power and Energy Systems, Vol. 22, pp. 349-354, 2000.

[122] Warwick K., Ekwue A. and Aggarwal R. Edition, "Artificial intelligence techniques in power systems", The Institution of Electrical Engineers, London, 1997.

[123] Weerasooriya S., El-Sharkawi M.A., Damborg M. and Marks R.J., "Towards static-security assessment of a large-scale power system using neural networks", IEE Proceedings-Generation, Transmission and Distribution, Vol. 139, No. 1, pp. 64-70, 1992.

[124] Wehenkel L., "Machine-learning approaches to power system security assessment", IEEE Expert, pp. 60-72, Sep./Oct. 1997.

[125] Whitehead B.A. and Choate T.D., "Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction", IEEE Transactions on Neural Networks, Vol. 7, No. 4, pp. 869-880, July 1996.

[126] Whitehead B.A., "Genetic evolution of radial basis function coverage using orthogonal niches", IEEE Transactions on Neural Networks, Vol. 7, No. 6, pp. 1525-1528, Nov. 1996.

[127] Wong K.P. and Wong Y.W., "Genetic and genetic/simulated-annealing approaches to economic dispatch", IEE Proceedings-Generation, Transmission and Distribution, Vol. 141, No. 5, pp. 507-513, 1994.

[128] Wood A.J. and Wollenberg B.F., "Power generation, operation and control", pp. 421-427, John Wiley & Sons, Inc., New York, 1996.

[129] Yalcinoz T. and Short M.J., "Neural networks approach for solving economic dispatch problem with transmission capacity constraints", IEEE Transactions on Power Systems, Vol. 13, No. 2, pp. 307-313, May 1998.

[130] Yao X., "Evolving artificial neural networks", Proceedings of the IEEE, Vol. 87, No. 9, pp. 1423-1447, Sep. 1999.

[131] Yegnanarayana B., "Artificial neural networks for pattern recognition", Sadhana, Vol. 19, pp. 189-238, Apr. 1994.

[132] Yegnanarayana B., "Artificial neural networks", Prentice Hall of India, 1999.

[133] Yuryevich J. and Wong K.P., "Evolutionary programming based optimal power flow algorithm", IEEE transactions on Power Systems, Vol. 14, No. 4, pp. 1245-1250, 1999.

[134] Zaborszky J., Whang K. and Prasad K., "Fast contingency evaluation using concentric relaxation", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-99, No. 1, pp. 28-36, Jan./Feb. 1980.

[135] Zimmerman R.D. and Gan D., "Matpower-Power system simulation package", Cornell University, 1997.

LIST OF PUBLICATIONS

Papers Accepted in Journals

1. D.Devaraj, B.Yegnanarayana and K.Ramar, "Radial basis function networks for fast contingency ranking", Accepted for publication in Electric power and energy systems Journal.

Papers Communicated to Journals

1. D.Devaraj and B.Yegnanarayana,"Fast contingency selection for static-security assessment: GA-RBF network approach", Submitted to Electric Power Systems Research Journal.

Papers in International Conferences

1. D.Devaraj, T.V.S.L.N.Murthy and B.Yegnanarayana, "A Fuzzy system model for Plant condition Monitoring", International AMSE conference on Computer Modeling, Simulation and Communication, Jaipur, India, pp. 210-214, December-1999.
2. D.Devaraj and B.Yegnanarayana, "An adaptive fuzzy system model for power system security assessment", International conference on Communications, Control and Signal Processing, Bangalore, India, pp. 284-288, July-2000.
3. D.Devaraj and B.Yegnanarayana, "Performance of neural network-based contingency selection with reduced input features", Accepted for presentation in Intellligent System Applications to Power Systems (ISAP-2001) conference, Hungary, June 2001.

Papers in National Conferences

1. "A combined Genetic Algorithm approach for optimal power flow", D.Devaraj and B.Yegnanarayana. Eleventh national power systems conference, Bangalore, India, pp. 524-528, December-2000.