Sudan University of Science & Technology
Faculty of Computer Science and Information Technology

Modeling Risk Assessment in Computational Grid Using Machine Learning Techniques

نمذجة تقييم المخاطر في منظومة الشبكة الحوسبية باستخدام طرق تعلم الآلة

In Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in Computer Science

Candidate
Sara Abdelwahab Abdelghani Ghorashi

Supervisor
Professor Dr. Ajith Abraham

July 2016

بسم الله الرحمن الرحيم

Sudan University of Science & Technology

College of Graduate Studies

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

SUST

كلية الدراسات العليا

Approval Page

Name of Candidate: Sara Abdelwahab Abdelgani Ghorashi

Thesis title: Modeling Risk Assessment in Computational Grid Using Machine Learning Techniques

نمذجة تقييم المخاطر في منظومة الحوسبة ~~الشبكية~~ الشبكية باستخدام طرق التعليم الآلي

Approved by:

1. External Examiner

Name: Dr. Abubakar A. Margani

Signature: .... Date: 28/5/2016

2. Internal Examiner

Name: Dr. Mohamed Elhafiz Mustafa

Signature: .... Date: 28/5/2016

3. Supervisor

Name: Prof. Ajith Abraham

Signature: .... Date: 28/5/2016

Sudan University of Science and Technology
College of Graduate Studies

Declaration

I, the signing here-under, declare that I'm the sole author of the Ph.D. thesis entitled Modeling Risk Assessment in Computational Grid Using Machine Learning Techniques which is an original intellectual work. Willingly, I assign the copy-right of this work to the College of Graduate Studies (CGS), Sudan University of Science & Technology (SUST). Accordingly, SUST has all the rights to publish this work for scientific purposes.

Candidate's name: Sara Abdelwahab Abdelgani Ghorashi

Candidate's signature: Sara Date: 14/7/2016

إقرار

أنا الموقع أدناه أقر بأننى المؤلف الوحيد لرسالة الدكتوراه المعنونة بنمذجة تقييم المخاطر في منظومة الشبكة الحوسبية باستخدام طرق تعلم الآلة

وهى منتج فكري أصيل . وباختياري أعطى حقوق طبع ونشر هذا العمل لكلية الدراسات العليا جامعه السودان للعلوم والتكنولوجيا ،عليه يحق للجامعه نشر هذا العمل للأغراض العلمية .

اسم الدارس : ساره عبدالوهاب عبدالغني قرشي

توقيع الدارس : ساره التاريخ : ١٤/ ٧/ ٢٠١٦م

DEDICATION

To my beloved mother, soul of my father, lovely husband who gave me great support, my Kids who suffer more with me, my supervisor for his continuous support brothers, sisters, friends, and to whole Muslim Umma.

ACKNOWLEDEMENT

“Praise be to Allah, the cherisher and the sustainer of the world”, “praise be to him he who taught by the pen, taught man, that which he did not know”

First and foremost, my sincere thanks to God, who endowed me to complete this PhD thesis. I would like to thank my supervisor, Professor Dr.Ajith Abraham, for all his guidance, support, brilliant ideas, numerous hours of discussions, patience, and the opportunities he have presented me. His managerial skills and uncompromising quest for excellence always motivated me to present the best of what I can. Many thanks to Prof.Izzaldin Mohammed Othman, who be as our Spiritual Father. All thanks to Dr.Talaat Wahby and Dr.Mohmmed Alhafiz and all the staff of computer science and information college in SUST for great support and help. And most importantly, I would like to thank my family for their love and support, especially my mother, my husband and my kids for their continuous support and supplication. Great thanks are extended to Dr. Nadir Omer from Sudan, Dr.Mohmmed Alfakki from Saudi Arabia, Dr. Anwar, and the other friends for great help and support. Finally, I have made many friends during my time at SUST and I would like to thank all my friends and colleagues for their support and help.

ABSTRACT

Assessing risk in a computational grid environment is an essential need for a user who runs applications from a remote machine on the grid, where resource sharing is the main concern. As grid computing is the ultimate solution believed to meet the ever-expanding computational needs of organizations, analysis of the various possible risks to evaluate and develop solutions to resolve these risks is needed. For correctly predicting the risk environment, we made a comparative analysis of various machine learning modeling methods on a dataset of risk factors. First we conducted a survey with International experts about the various risk factors associated with grid computing. Second we assigned numerical ranges to each risk factor based on a generic grid environment. We utilized data mining tools to pick the contributing attributes that improve the quality of the risk assessment prediction process. Finally,we modeled the prediction process of risk assessment in grid computing utilizing Meta learning approaches in order to improve the performance of the individual predictive models. Prediction of risk assessment is demanding because it is one of the most important contributory factors towards grid computing. Hence, researchers were motivated for developing and deploying grids on diverse computers, which is responsible for spreading resources across administrative domains so that resource sharing becomes effective. We present an adaptive neuro-fuzzy inference system that can provide an insight of predicting the risk environment. Also, we used a function approximation tool, namely, flexible neural tree for risk prediction and risk (factors) identification. Flexible neural tree is a feed forward neural network model, where network architecture was evolved like a tree. Our comprehensive experiment finds score for each risk factor in grid computing together with a general tree-based model for predicting risk.The empirical results illustrate that the proposed framework is able to provide risk assessment with a good accuracy. We concluded that data mining tools can provide further steps in building a risk assessment model in a Grid environment with good accuracy, according to the obtained empirical results.

المستخلص

تقييم المخاطر في بيئة الحوسبة الشبكية يتعبر من الإحتياجات الضرورية للمستخدمين الذين يقومون بتنفيذ تطبيقات من طرفيات أو أجهزة بعيدة على الشبكة التي يكون فيها مشاركة الموارد من الإعتبارات المهمة في بيئة الحوسبة الشبكية الحل الناجح والأمثل ينبغي أن يتضمن التوسع في متطلبات الحوسبة للمؤسسات ، وكذلك تحليل المخاطر المحتملة المختلفة لتقييم وتطوير الحلول المختلفة لمجابهة تلك المخاطر. في سبيل التوقع الصحيح لتلك المخاطر قمنا بتحليل ومقارنة منهجيات مختلفة من أساليب تعلم الآلة على بينات متعلقة بعوامل المخاطر. أولاً قمنا بعمل إستبيان مباشر مع مجموعة من الخبراء الدوليين فيما يتعلق بعوامل المخاطر المختلفة التي لها علاقة بالحوسبة الشبكية. ثانياً قمنا بتخصيص أو تعيين قيم رقمية لكل عامل من عوامل المخاطر بناء على بيئة الحوسبة العامة. لقد تمت الإستفادة من أدوات تنقيب البيانات لإختيار الخصائص الفعالة التي تؤدي لتحسين جودة عملية توقع تقييم المخاطر. أيضاً قمنا بنمذجة عملية التوقع لتقييم المخاطر في بيئة الحوسبة بالإستفادة من أساليب تعلم ال Meta بهدف تحسين كفاءة نماذج التوقع الآحادية. هناك حوجة ماثلة لتوقع تقييم المخاطر لأنها تعتبر من العوامل المهمة في بيئة الحوسبة الشبكية مما يشجع الباحثين على تطوير وتنفيذ بيئة الحوسبة الشبكية على الحواسيب المختلفة مما يقود لتوزيع الموارد المختلفة على نطاقات إدارية مختلفة مما ينتج عنه توزيع الموارد بصورة فعالة. قمنا بتقديم نظام إستدلالي ضبابي عصبي يقوم بالتوقع الداخلي لبيئة المخاطر. أيضا تم إستخدام التقريب الدالي التي تعرف الشجرة العصبية المرنة لتوقع وتعريف المخاطر. الشجرة العصبية المرنة هي أحد نماذج الشبكة العصبيية ذات التغذية الأمامية التي تعتبر بنية الشبكة فيها مثل الشجرة. التجارب أوجدت قيمة لكل عامل خطر في بيئة الحوسبة الشبكية مقروناً مع نموذج شبكة عام لتوقع المخاطر.

نتائج الدراسة أوضحت أن الإطار المقترح له القدرة على تقييم المخاطر بدقة جيدة ، كما خلصنا أيضاً إلى أن أدوات تنقيب البينات يمكنها تقديم خطوات متقدمة في بناء نموذج تقييم المخاطر في بيئة الحوسبة الشبكية بدقة جيدة إعتماداً على النتائج المستخلصة.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

VO	-	Virtual Organization
RA	-	Risk Assessment
QoS	-	Quality of Services
NASA	-	National Aeronautics and Space Administration
IPG	-	Information Power Grid
NSF	-	National Science Foundation
GT	-	Globus Toolkit
OGSA	-	Open Grid Services Architecture
GSI	-	Grid Security Infrastructure
SLAV	-	Service Level Agreement Violation
CDA	-	Cross Domain Attacks
JS	-	Job Starvation
RF	-	Resource Failure
RA	-	Resource Attacks
PA	-	Privilege Attack
CB	-	Confidentiality Breaches
IV	-	Integrity Violation
DDoS	-	DDoS Attacks
DA	-	Data Attack
DE	-	Data Exposure
CV	-	Credential Violation
MMA	-	Man in the Middle Attack
PV	-	Privacy Violation
SA	-	Sybil Attack
HIC	-	Hosting Illegal Content
SIO	-	Stealing the Input or Output

ShUTh	-	Shared Use Threats
SIO	-	Stealing or altering the software
PM	-	Policy Mapping
RE	-	Risk Exposure
UO	-	Unsatisfactory Outcome
ENISA	-	European Network and Information Security Agency
EME	-	Small and Medium Enterprises
ISO/IEC	-	International Organization for Standardization (ISO) and the International Electro technical Commission (IEC).
SLA@SOI	-	Services Level Agreement at Service-Oriented Infrastructure
MTTF	-	Mean Time between Failure
MTTR	-	Mean Time to Repair
MEP	-	Multi Expression Programming
GP	-	Genetic Programming
FIS	-	Fuzzy Inference System
HiNFRA	-	Hierarchical Neuro Fuzzy Risk Assessment
NRTSAPD	-	Near Real Time Statistical Asset Priority Driven Risk Assessment
RP	-	Resource Provider
AGP	-	Access Grid Project
FAIR	-	Factor Analysis for Information Risk
OCTAVE	-	Operationally Critical Threat, Asset and Vulnerability Evaluation
NHPP	-	Non Homogeneous Poisson Process
IPS	-	Intrusion Prevention System
IDS	-	Intrusion Detection System
DIPPS	-	Distributed Intrusion Prediction and Prevention Systems
DIDS	-	Distributed Intrusion Detection System
HMM	-	Hidden Markov Model
CFS	-	Correlation based Feature Selection

FNT	-	Flexible Neural Tree
ANFIS	-	Adaptive Neuro Fuzzy Inference System
ANN	-	Artificial Neural Network
RMSE	-	Root Mean Square Error
CC	-	Correlation Coefficient
CGS	-	Computational Grid System

CHAPTER 1

INTRODUCTION

1.1 Grid Computing

Grid is a term devised in the mid-1990s by Foster [1, 2], which represents an emerging computing paradigm. Currently, Grid has been applied in many applications to solve large-scale scientific and commercial problems [3-7]. A Grid is a collection of diverse computers and resources spread across several administrative domains with the purpose of resource sharing. While Grid computing was introduced in 1998 as a viable option for high-performance computing, with the idea that sharing of resources provides improved performance at a lower cost than if each organization was to own its own "closed-box" resources [8-11]. Its main concern is resource allocation, which includes processing power and storage capacity. This is carried out in a well-coordinated manner by virtual organizations (VO) [12]. Grid computing has no formal definition [13], but many researchers provide various perspective that try to define Grid. A definition by Czajkowski [14] states that "Grid technologies allow large-scale sharing of resources within formal or informal group of companies or individuals which is known as virtual organizations". According to Foster [15], a Grid is a system that conforms to three specific categories: it coordinates resources that are not subject to centralized control, it uses standard, open, general purpose protocols and interfaces, and it delivers nontrivial quality of service.

Grid computing is also, defined in the literature as "systems and applications that integrate and manage resources and services distributed across multiple control domains" [16].

1.2 Grid Computing Security

Data security in grid computing is an area full of challenges and of paramount importance and is still in its infancy now. Many research problems are not yet be identified and a recent research shows that data security in grid have become the primary concern for people to shift to grid computing because the data is stored and hosted in many different locations. Data security in the grid is not only focused on the process of data transmission, but also the system security and data protection for the data stored in the storages of the grid. The following Section describes an overview the current grid security situation as well as the security policy involved in grid Environment.

1.3 Grid Computing Security Issues

In grid environment, resources from many domains are connected together. The concern is to protect data and applications from both unauthorized users and the computer system that runs the applications. Strong authentication measures are required for genuine users and programs. In addition the users problems can be run on local system. Local execution should also be secured from remote systems. Interoperability between various security policies are needed since multiple administrative systems are involved in the grid [17]. As a ground-breaking technology, the Grid causes new security issues, in terms of the requirement for improved intensity and flexibility of security mechanisms. Also, Grid entities must have the capacity to negotiate their security policies. For security policy negotiation to be achieved, effective security policy reconciliation is needed. Even though the use of grid computing has become the best choice for scientific, engineering and commercial applications; there are many challenges computational grid is facing in terms of secure utilities [18].

Managing security in computational grid is a serious issue as a result of the various distributed resources and broad range of users, each with different requirements for the grid. Therefore, security in the grid is a crucial aspect. Without satisfying it, the grid becomes susceptible to unauthorized users which leads to data tampering, and malicious activities that may likely make grid futile [19]. In grid computing, Virtual Organizations (VOs) enable different groups of organizations and/or individuals, with different administrative domains, to share resources in a controlled manner, which brings the challenge of some security issues. Therefore it is the responsibility of the resource management system to ensure that various resources are handled properly while conforming to the various usage policies. Due to the nature of the grid computing environments, they are easily targeted by intruders who are looking for potential vulnerabilities to exploit. The intruders impersonate legitimate users, to gain access to resources and act maliciously [20].

Security in general is the degree of resistance to or protection from harm. It applies to any vulnerable and valuable asset. Also, security assurance is to guarantee the integrity and confidentiality of data and authentication to ensure the identity of the user before granting permission to access resources [21]. Traditionally the definition of security is to protect a system from its users or to protect data from compromise. While security in grid computing is to ensure the protection of applications and data from misuse. Therefore, strong and reliable means of authentication is essential for both users and codes. Furthermore, security policies must be put in place to protect local execution from remote systems. Computational Grid resources can be accessed in many ways, each way having its unique security requirements and implications for both the resource provider and the user. Their design objective is to provide easy and secure access to the diverse resources in the grid. Foster [22], mentioned that, managing transactions in the grid have a number of interesting requirements, such as the following:

Single sign-on: This is a situation in which a user should be able to authenticate once and initialize computations that get resources, use the resource and release it, as well as to communicate internally, without the need for re-authentication.

Delegation: Is a scenario in which another entity get the right to carry out some action on behalf of a user [23]. The proxy credential creation is a form of delegation; it is important operation in Grid environment. A computation that cut across many resources generates sub computations that may generate requests to other resources and services, and so on. The more these delegated credentials the greater the risk.

Authorization and policy: In a large grid environment, the policies that control resources access cannot be based on individual identity and resources cannot keep track of Virtual Organization (VO) membership and privileges. Instead, the resources and its users have to express policies in terms of other criteria, like group membership. Authentication, authorization, and policy are among the most challenging issues in grid.

1.4 Network Security Issues

Threat to the information security poses a security issue by hacking into a computer system or a network. During the design of computer operating systems and application software, there are often some flaws or vulnerabilities. An attacker mostly searches for these flaws to invade the system. All computer platforms are vulnerable and subject to network attacks. Once the attacker is able to locate the flaws, he/she try to gain control of the computer system, and cause damages. The attackers may steal passwords, intercept data, transmit viruses and sometimes destroy whole computer systems. Majority of the successful intrusions result from the internal network and currently most of the intrusion detection systems are difficult to detect attacks from the internal network [24].

1.5 Grid Computing Security Policies

The wide area of security within grid computing requires many parties' collaborative efforts to overcome risks, gaps and vulnerabilities, which are threatening the grid security. Risk is a function of threats exploiting vulnerabilities to cause damage or destroy assets. Thus, threats (actual, conceptual, or inherent) may exist, but if there are no vulnerabilities then there is little or no risk. Equally, you can have vulnerability, but if you have no threat, then you have little or no risk. Risk is considered as the possibility that a valuated entity will be negatively affected by vulnerability while "vulnerability" is any unsafe situation with potential for harm. In addition, risk is defined as a measure under uncertainty of the severity of vulnerability. Vulnerability is a weakness or gap in our protection efforts, and risk is the intersection of assets, threats, and vulnerabilities While, a threat is what we're trying to protect against vulnerabilities. In order to prevent security breaches, grid uses controls such as authentication, single sign on, access control, security policy, and so on to protect resources from various types of threats. Even though with the use of controls the grid is still not fully protected. In general, information security requirements for a system include three main security properties: confidentiality, integrity, and availability. In the grid computing the four classical security areas are: authentication and authorization, confidentiality or privacy, and integrity, and availability [16], [25-28], which is explained as follows:

1.5.1 Authentication and Authorization

Authentication is a core security in grid computing that requires mutual trust between parties. Common tools used for authentication in grid computing such as protocols and certificates are based on cryptographic algorithms. In general, authentication is achieved through the presentation of some token that cannot be forged. Biometrics can be used, especially as a mechanism by which a human can acquire a token that is later presented to a service for authentication purposes. For example a fingerprint scanner can be used to log in to a local machine [16]. While, authorization, in general, is based on authentication schemes. There are two general approaches for authorization, which are identity-based or token based. Identity based approach is typically associated with access control lists, while token-based approach is also referred to as capability-based authorization [16]. A drawback of identity-based approach is that it cannot easily support delegation. On the other hand, a drawback of a token-based approach is it may be very difficult to dynamically revoke access rights.

1.5.2 Confidentiality

Important data or programs that are transmitted or transported between parties should be secured and protected. Although, cryptographic algorithms and policies are necessary to gain a high level of confidentiality, port monitoring of remote machines are also very important.

1.5.3 Integrity

Integrity is defined as an" issue that concentrate on what prevent subversion of a system if someone did get it. Based on this definition, there is a need to apply a physical security schemes in grid environment such as policies and tool, which can save the data, applications, and any equipment’s from damage or loss. So far, the responsibility for integrity in grid computing relies on the individual organizations or user.

1.5.4 Availability

Availability of information refers to ensuring that authorized parties are able to access the information when needed. Information only has value if the right people can access it at the right times.

1.6 Risk Assessment Associated with Grid Computing

Risk assessment is a wide concept that can be applied in many context of Grid computing involving performance, resource failure and security [29]. Currently, Grid has been applied in many applications to solve large-scale scientific and e-commerce problems [30]. Therefore, risk reduction is needed to avoid security breaches. In order to offer reliable Grid computing services, a mechanism is needed to assess the risks and make precaution measures to avoid them. Risk assessment is a set of methods that is applied in information system to investigate the probability of event that causes harm to assets [31]. Risk assessment has been studied extensively using different approaches to model it such as quantitative, qualitative and hybrid approaches. Numerous risk assessment models have been provided using different techniques to make risk assessment more accurate and reliable. More details are provided in Chapter 2.

1.7 Feature Selection for Prediction Technique

Feature selection is a preprocessing step that reduces dimensionality from a dataset, in order to have better prediction performance [32]. Feature Selection can be viewed as a search problem, searching of a subset from the search space in which each state represents a subset of the possible features. To avoid high computational cost and enhance the prediction accuracy, irrelevant input features are reduced from the dataset before constructing the prediction model. There are three main components of feature selection algorithms. The first component is the algorithm that searches the space of feature subsets, while the second component is the search evaluation function, which takes a state from a search space as an input, and produces a numeric evaluation as an output. The search algorithm aims to maximize this function. The third component is the predicator, which is the target algorithm using the final subset of features found by the search algorithm, due to its highest evaluation function value. Figure 1.1 depicts the Filter Approach for Feature Selection [33].

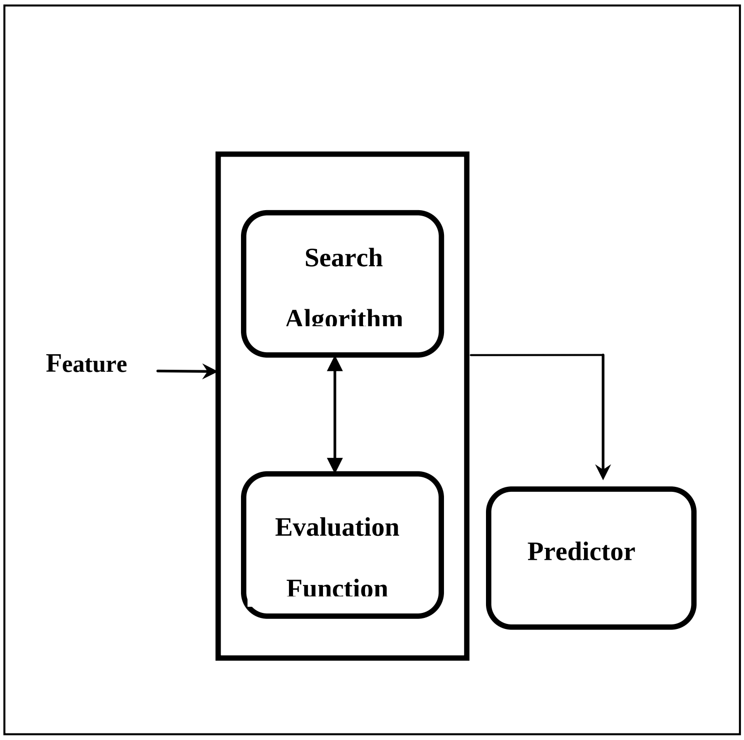


Figure 1.1. depicts the Filter Approach for Feature Selection

1.8 Research Motivation

As grid computing is the ultimate solution believed to meet the ever expanding computational needs of organizations, analysis of the various possible risks to evaluate and develop solutions to resolve these risks is needed [27]. Assessing risk in a computational grid environment is an essential need for a user who runs applications from a remote machine on the grid, where resource sharing is the main concern [34]. An accurate identification of risk associated with the grid, contributes significantly in supporting the decision maker, which results in more efficient use of computational grids.

1.9 Problem Statement

The problem of predicting risk is a complex issue because there are many factors that affect the grid computing directly and indirectly. Therefore, many risk factors were reported. These factors affect the grid computing in many ways such as availability, integrity and confidentiality. However, the problem of facilitating security of grid computing becomes even more challenging. Some current proposed models [35] used various reliability models to assess and evaluate on the basis of assuming Weibull distribution as the best-fit model. However, their work suffered a limitation of requirement of aggregation at both the levels of component and node level. Further, this concept was enhanced to improve reliability within the grids using stochastic model that extracted grid-trace-logs and thus enhanced the job resubmission strategy. Moreover, these works does not address the component-level risk assessment in grids, where the components could be either the disks, CPU, computer software, computer memory, etc. While, in [36]; authors address the problem of risk assessment in computational grid considering security aspect, also do not reflect any insight of the grid failure data.

Recently, the importance of probabilistic RA method has been highlighted by several research works [37-40]. These models while can be useful to predict the risk of resource failure in grid environment but most of them do not consider dynamic data operations. Hence, the reliability, performance and flexibility of grid computing under a number of node failures, grid applications and risk failures should be investigated and clarified.

Many various approaches in RA has been applied [41, 42]. The authors used fuzzy logic, to addressed the uncertainty problem in modeling risk, however the proposed model face the challenge of No distinct way to formulate human knowledge as knowledgebase [43]. Many researchers [44, 45] tackled the predicting problem in RA approach, however these models were performed statically and failed to reflect the changes in a dynamic grid environment [46]. Modeling risk is a complex problem and inaccurate as there are many risk factors associated with grid computational services, each factor with many characteristics. For every risky situation, it is probable that the change of depending factors can occur during run time; consequently we need to build a compact and dynamic model based on such emerging security conditions in order to reflect the continuous change in Grid computing environments.

1.10 Research Questions

In order to expand the stated research problem, the following sub-problems need to be answered:

1. How the risk factors associated with grid computing will be identified?
2. What is the appropriate scope for the assessment?
3. What is an appropriate approach, and what level of detail is needed?
4. Who is going to be involved?
5. How to affect the performance of several machine-learning methods for the risk assessment prediction process for the identified risk factors?
6. Can the proposed model reduce the threats to the data integrity that are redundantly stored in multiple physical locations?

1.11 Research Objectives

The main objectives of this research are:

1. To identify the security risk factors associated with grid computing.
2. To simulate the data based on risk factors associated to grid computing.

3. To formulate the proposed model for assessing risk in the grid computing Environment.

4. To enhance the performance in predicting risks associated with Grid computing.

1.12 Scope of the Study

The scope of this research is to identify current security concerns about grid computing environments and describes the methodology for ensuring application and data security. Our research is scoped in different levels as follows and illustrated in Figure 1.2:

[1]. **Availability:** Availability indicates the percentage of time, usually on a monthly basis, in which the Grid service supplied by the provider will be available [47].

[2]. **Integrity:** refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change**.**

[3]. **Confidentiality and Access Control:** refers to the protection of information from unauthorized disclosure [48].

The overall security of the grid is at stake if the authentication and authorization mechanisms are not strong enough to handle the access control properly. This will result in an unauthorized access to the resources.

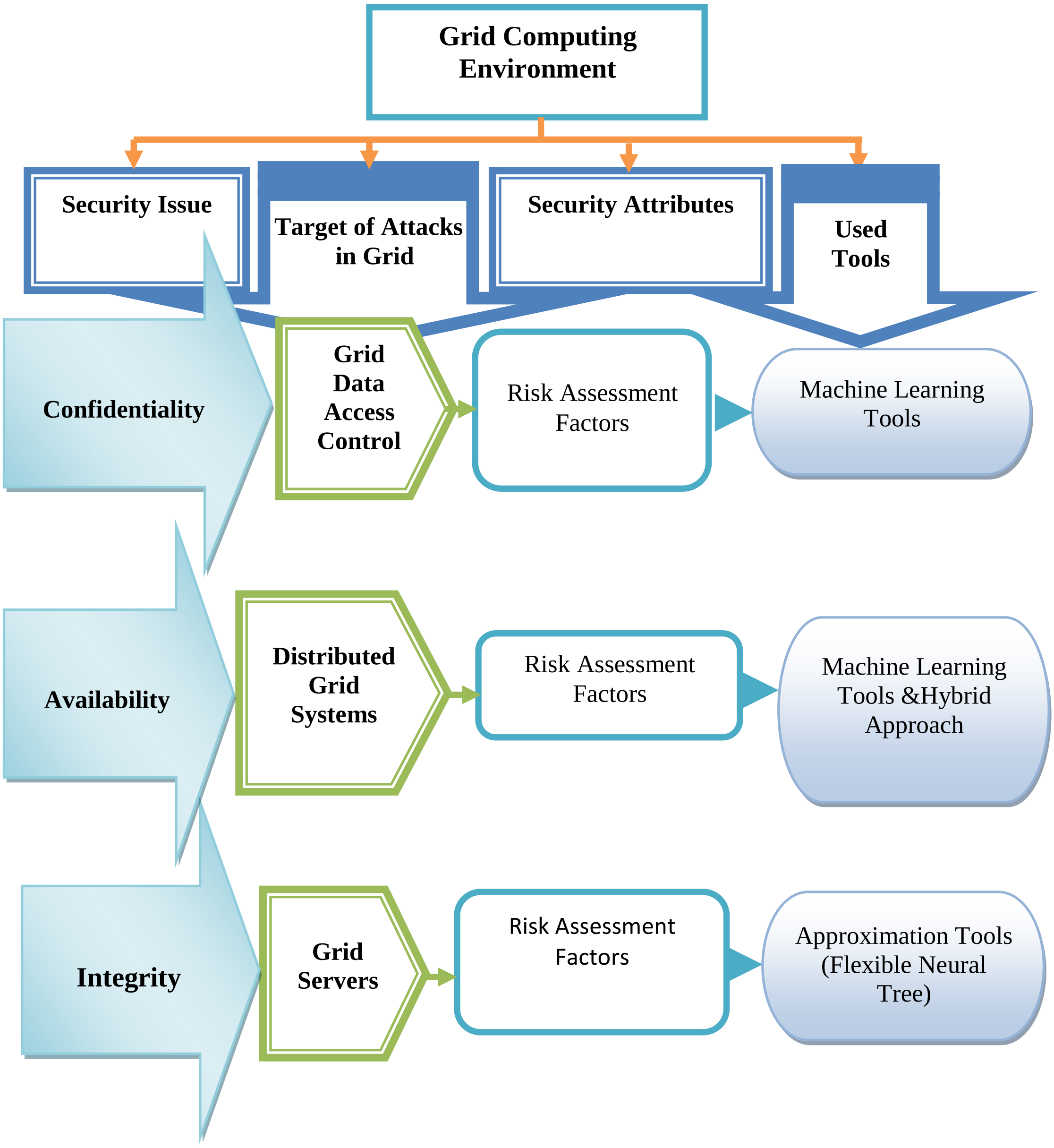


Figure 1.2. Research Scope Framework Concerning to Simulated Data

1.13 Research Contributions

The following are the list of contributions of this research:

[1]. Mechanism of identifying the security risk factors associated with the grid computing.

[2]. Risk assessment model based on feature selection of prediction techniques to facilitate the confidentiality, availability and integrity of grid computing data.

[3]. The model is simulated after detailed analysis of risk factors associated with the grid computing.

[4]. The performance evaluation of the proposed model.

1.14 Thesis Organization

The results obtained from this research are presented into seven chapters. This thesis is organized as follows:

Chapter 1 gives a general introduction of grid computing, security issues, and security policy in grid computing. Beside, presents the problem statement, objectives, contributions and the scope of the research.

Chapter 2 continues with a comprehensive literature review of grid computing service models, types of grid computing, grid computing characteristics, grid uses, and security goals in grid computing. This chapter also has considered risk assessment and discussed the types of methods for risk assessment. Finally, existing feature selection for prediction technique associated with grid computing and existing security policies in grid computing are also highlighted.

Chapter 3 presents the research methodology employed in conducting the research work. Specifically, research approaches for identifying risk factors in grid computing, determined the significant factors, developing a framework for predicting risk in grid computing, and finally constructing an ensemble model were discussed

Chapter 4 discusses risk and risk analysis, the types of methods used to analyze risk. Examples of risk items identified are provided. A list of utilized risk factors for experimental design adopted in order to identify and assess risk is presented.

Chapter 5 presents our proposed risk assessment model that aid to predict risk in grid computing environment. As well as the validation of the proposed model is presented.

Chapter 6 presents the details of the results driven from risk assessment model to predict risk in grid computing model, and a comparative evaluation of results is also discussed.

The research is ended up with a conclusion and future work in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

This Chapter presents a detailed literature review consisting of a number of salient themes, frameworks, models, architectures and approaches important for this research. This Chapter illustrate grid computing service models, types of grid computing, grid computing characteristics, grid uses, and security goals in grid computing. Risk assessment is considered in detail and discussed the types of methods for risk assessment. Finally, existing feature selection for prediction technique associated with grid computing and existing security issues in grid computing are highlighted.

2.1 Grid Computing

Grid computing is applying the resources of many computers in a network to a single computational problem that requires large number of computer processing cycles or access to huge amounts of data [49]. One of the basic requirements of a grid system is the ability to provide the high-level quality of service needed for a satisfactory user experience. Thus, Quality of service (QoS) validation must exist as a basic feature in any grid system, as measured by the available resource metrics [50]. These metrics include response time measurements, aggregated event performance monitoring and measurements, security fulfillment, resource scalability, availability, autonomic features, fail-over mechanisms and networking services [51].

2.2 Grid Computing Service Models

A Grid can be viewed as a seamless, integrated computational and collaborative environment and a high-level view of activities within the Grid [52]. The users interact with the Grid resource broker to solve problems, which in turn performs resource discovery, scheduling, and the processing of application jobs on the distributed Grid resources. From the end-user point of view, Grids can be used to provide the following types of services [53]:

2.2.1 Grid Computational Services Model

These are concerned with providing secure services for executing application jobs on distributed computational resources individually or collectively. Resources brokers provide the services for collective use of distributed resources. A Grid providing computational services is often called a computational Grid. Some examples of computational Grids are: NASA IPG [54] and the NSF TeraGrid [55].

2.2.2 Grid Data Services Model

These are concerned with proving secure access to distributed datasets and their management. To provide a scalable storage and access to the data sets, they may be replicated, catalogued, and even different datasets stored in different locations to create an illusion of mass storage. The processing of datasets is carried out using computational Grid services and such a combination is commonly called data Grids. Sample applications that need such services for management, sharing, and processing of large datasets are high-energy physics [56] and accessing distributed chemical databases for drug design [57].

2.2.3 Grid Application Services Model

These are concerned with application management and providing access to remote software and libraries transparently. The emerging technologies such as Web services are expected to play a leading role in defining application services. They build on computational and data services provided by the Grid. An example system that can be used to develop such services is NetSolve [58].

2.2.4 Grid Information Services Model

These are concerned with the extraction and presentation of data with meaning by using the services of computational, data, and/or application services. The low-level details handled by this model are the way that information is represented, stored, accessed, shared, and maintained. Given its key role in many scientific endeavors, the Web is the obvious point of departure for this level [57].

2.2.5 Grid Knowledge Services Model

These are concerned with the way that knowledge is acquired, used, retrieved, published, and maintained to assist users in achieving their particular goals and objectives [59]. Knowledge is understood as information applied to achieve a goal, solve a problem, or execute a decision. An example of this is data mining for automatically building a new knowledge [57].

2.3 Grid Architecture

Many researchers have categorized the grid component into different classes. Schwiegelshohn et al. [60] classified grid component into three groups. These are hardware resources, domain independent software component, and application software. The domain independent software component is used to control access to resources and virtual organizations (VO). The application software component is dedicated to the needs of different VO within a virtual research environment. It is evident that the implementation of this classification can vary on account of the first and third layers. This is as a result of the large number of diverse resources in the grid coupled with many disciplines that can exploit the computational grid. Grid is a protocol based architecture that determine the fundamental approach employed by VO to control the relationship that exist among partners [61]. Built on top of these protocols are a set of standard protocols, middleware, toolkits, and services that are provided and defined by grids, to help in the construction of VO [62]. Essentially grid system components and procedures can be determined by the system architecture, as well as how these components communicate with each other [61]. The grid architecture can be viewed as five layers, where the components, which share common attributes, form a layer. The description of each of the five layers is provided below [38]. The architecture aims to recognize the requirements for general classes of components, instead of counting all needed protocols which leads to flexible and open architectural structure [61, 62]. Components are arranged in layers, as illustrated in Figure 2.1. The higher layers were built based on capabilities and behaviors of lower layers.

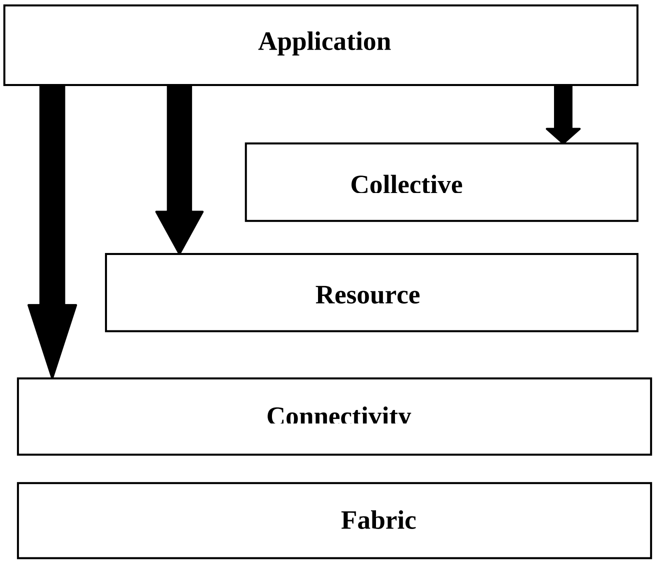


Figure 2.1. Grid Layered Architecture

Fabric Layer:

As shown in Figure 2.1, the Grid Fabric layer provides access to the resources that are shared, with the help of Grid protocols. Also, the resources are many and they include storage systems, computational resources, network resources, catalogs and so on. It can also be a logical entity, for instance a distributed file system, computer cluster or distributed computer pool [61, 62].

The connectivity layer:

This layer identifies a core communication and authentication protocols, for easy and secure network transactions [62].

The resource layer:

This layer is concerned mainly on individual resource management by defining protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations. The Resource Management System (RMS) is used to manage all the resource processes such as allocation, monitoring, and utilization [63].

The collective layer:

The job of this layer is the organization of multiple resources. It is not directly associated with any particular resource. It contains protocols as well as services that are global. It also captures interactions that exist across resource collections.

The application layer:

This is the layer that the user interacts with. It is the final layer at the top of the Grid architecture. It includes the user applications used within the Virtual Organization (VO) environment. All the layers have well defined protocols that allow access to relevant services. Applications are designed with respect to services that are defined at any layer.

Grid Middleware

This is software that gives an integral part of the grid infrastructure. It is responsible for the formation of layers between programs or tasks that need to be executed on the grid and on the physical machines [60, 64]. Grid Middleware also gives several functions like job scheduling, task parallelizing and even security. Essentially, good middleware is needed to run tasks, which helps to avoid miserable failure of the grid infrastructure.

Globus Toolkit

The Globus Toolkit (GT) was developed in the late 1990s to aid the development of service-oriented distributed computing applications and infrastructures. Core GT components address, basic issues. The issues are concerning security, resource management, resource access, resource discovery, and data movement. These GT components facilitate a wider "Globus ecosystem" of tools and components that interoperate with, core GT functions to give a varying degree of good application-level functionality. These tools have been applied in the development of wide range of both "Grid" infrastructures and distributed applications [65]. Globus Toolkit is a free middleware, and this makes it very popular [63]. Also, many of the defined standards for computational grids have been implemented in the middleware, such as Open Grid Services Architecture (OGSA) and Grid Security Infrastructure (GSI).

2.4 Types of Grid Computing

Currently there are three types of grid computing namely computational grid, scavenging grids, and data grids. Grid computing types summarized as follows:

2.4.1 Computational Grid

In this type of grid the resources are high performance servers, and the main concern of this type is a resource allocation particularly for computing power [9]. Computational grids are being used to solve large-scale scientific, engineering, and commerce problems. The advantages of grid computing are many [57] and does the following:

- Enable resource sharing
- Provide transparent access to remote resources
- Allow on-demand aggregation of resources at multiple sites
- Reduce execution time for large-scale, data processing applications
- Provide access to remote databases and software
- Take advantage of time zone and random diversity (in peak hours, users can access resources in off-peak zones)
- Provide the flexibility to meet unforeseen emergency demands by renting external resources for a required period instead of owning them

The enabling factors in the creation of computational grids have been the proliferation of the Internet and the Web and the availability of low-cost, high-performance computers [66].

2.4.2 Scavenging Grids

A scavenging grid (also known as desktop grid) is most commonly used with large numbers of desktop machines. Machines are scavenged for available CPU cycles and other resources [67]. Owners of the desktop machines are usually given control over when their resources are available to participate in the grid. It is use to find and harvest machine cycles from idle servers and desktop computers for use in resource-intensive tasks [68].

2.4.3 Data Grids

A third type is the data grids that provide a unified interface for all data repositories in an organization, and through which data can be queried, managed and secured [69]. A data grid is responsible for housing and providing access to data across multiple organizations. Users are not concerned with where this data is located as long as they have access to the data [70].

2.5 Grid Computing Characteristics

Ten definitions extracted from main grid literature sources have been examined to find out the essential characteristics that a grid is supposed to have in order to be considered as such [71]. As a result, a total number of ten characteristics have been identified. Both the definitions and the characteristics found in them, either explicitly or implicitly. These characteristics are described as follows:

- Large Grid Scale: a grid must be able to deal with a number of resources ranging from just a few to millions. This raises caused very serious problem of avoiding potential performance degradation as the grid size increases [72].

- Geographical Grid Distribution: grid’s resources may be located at distant places [71, 73].

- Heterogeneity Grid: a grid hosts both software and hardware resources that can be very varied ranging from data, files, software components or programs to sensors, scientific instruments, display devices, personal digital organizers, computers, super-computers and networks [74].

- Resource Grid Sharing: resources in a grid belong to many different organizations that allow other organizations (i.e. users) to access them. Nonlocal resources can thus be used by applications, promoting efficiency and reducing costs.

- Multiple Grid Administrations: each organization may establish different security and administrative policies under which their owned resources can be accessed and used. As a result, the already challenging network security problem is complicated even more with the need of taking into account all different policies.

- Resource Grid Coordination: resources in a grid must be coordinated in order to provide aggregated computing capabilities.

- Transparent Grid Access: a grid should be seen as a single virtual computer.

- Dependable Grid Access: a grid must assure the delivery of services under established Quality of Service (QoS) requirements. The need for dependable service is fundamental since users require assurances that they will receive predictable, sustained and often high levels of performance.

- Consistent Grid Access: a grid must be built with standard services, protocols and interfaces thus hiding the heterogeneity of the resources while allowing its scalability. Without such standards, application development and pervasive use would not be possible [75].

- Pervasive Grid Access: the grid must grant access to available resources by adapting to a dynamic environment in which resource failure is commonplace. This does not imply that resources are everywhere or universally available but that the grid must tailor its behavior as to extract the maximum performance from the available resources [76].

2.6 Grid Uses

Opposite to what is often believed, the grid is not only a computing paradigm for providing computational resources for grand-challenge applications. Instead, it is an infrastructure that bonds and unifies globally remote and diverse resources in order to provide computing support for a wide range of applications [71]. It is important to notice that grid uses are thus not defined in terms of applications (as usually found in the literature) but rather of the support the grid provides. The different types of computing support offered by grids can be categorized according to the main challenges that they present from the grid architecture point of view. This categorization is the following:

- Distributed Grid Supercomputing Support: allows applications to use grids to couple computational resources in order to reduce the completion time of a job [19] or to tackle problems that cannot be solved on a single system [66]. The main problems raised by applications requiring this support are the need to co-schedule the use of scarce and highly expensive resources, the scalability of protocols and algorithms to a large number of nodes, latency-tolerant algorithms as well as achieving high levels of performance [66]. Typical applications that require distributed supercomputing are weather forecasting and military scenario simulations.

- High-throughput Grid Computing Support: allows applications to use grids to put unused processor cycles to work in generally loosely coupled or independent tasks [66]. 'Parameter sweep' applications such as Monte Carlo simulations are well suited for high-throughput computing.

- On-demand Grid Computing Support: allows applications to use grids to retrieve resources that cannot be cost-effectively or conveniently located locally [66]. Challenging issues in order to provide on-demand computing support are resource location, scheduling, code management, configuration, fault tolerance, security, and payment mechanisms [66]. A financial application allowing users to perform accurate stock market analysis and price prediction employing their home desktop computer is a representative example of application requiring on-demand computing.

- Data-intensive Grid Computing Support: allows applications to use grids to synthesize new information from distributed data repositories, digital libraries and databases [66]. The creation of a new database using data mined from a number of online databases would be an example of data-intensive computing application.

- Collaborative Grid Computing Support: allows applications to use the grid to enable and enhance human-to-human interactions [66] in a synchronous or asynchronous way [77] via a virtual space. The real-time requirements imposed by human perceptual capabilities as well as the wide range of many different interactions that can take place are one of the most challenging issues of collaborative computing support [66]. Typical examples of applications that may use a collaborative computing infrastructure provided by grids are groupware applications and multi conferencing applications.

- Multimedia Grid Computing Support: allows applications to use grids to deliver contents assuring end-to-end QoS [19]. Main challenges for the multimedia computing support derive from the need to provide QoS across multiple different machines. Video conference applications are a typical example of application requiring multimedia computing support.

2.7 Importance of Grid Computing

- Grid computing is emerging as a viable technology that businesses can use to wring more profits and productivity out of IT resources and it's going to be up to you developers and administrators to understand Grid computing and put it to work [78].

- It's really more about bringing a problem to the computer (or Grid) and getting a solution to that problem. Grid computing is flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources [79]. Grid computing enables the virtualization of distributed computing resources such as processing, network bandwidth, and storage capacity to create a single system image, granting users and applications seamless access to vast IT capabilities. Just as an Internet user views a unified instance of content via the World Wide Web, a Grid user essentially sees a single, large, virtual computer.

- Grid computing will give worldwide access to a network of distributed resources - CPU cycles, storage capacity, devices for input and output, services, whole applications, and more abstract elements like licenses and certificates [80].

- For example, to solve a compute-intensive problem, the problem is split into multiple tasks that are distributed over local and remote systems, and the individual results are consolidated at the end. Viewed from another perspective, these systems are connected to one big computing Grid. The individual nodes can have different architectures, operating systems, and software versions. Some of the target systems can be clusters of nodes themselves or high performance servers [81].

2.8 Grid Computing Security

Some argue that customer data is more secure when managed internally, while others argue that grid have a strong incentive to maintain trust and as such employ a higher level of security [82, 83]. However, in the grid, the data will be distributed over these individual computers regardless of where the base repository of data is ultimately stored [84].

2.8.1 Security Goals in Grid Computing

There are seven security policies focused on grid computing security, these policies are: authentication, authorization, confidentiality, integrity, non-repudiation, management and access control in order to achieve an adequate grid computing security policies. The summary of grid computing security policies, requirements and their descriptions are shown in Table 2.1.

Table 2.1: Summaries of Grid Security Policies, Requirements and their descriptions

Grid Security Policies	**Descriptions**
Authentication	Ensuring mutual trust between parties
Authorization	Ensuring the process of giving someone permission to do or have something
Confidentiality	Ensuring that data is not disclosed to unauthorized persons
Integrity	Ensuring that data held in a system is a prior representation of the data and that it has not been modified by an unauthorized person
Non-Repudiation	Ensuring that a party in a dispute cannot repudiate or refute the validity of the statement
Management	Ensuring wide spread and variety of resources and the decentralization
Access Control	Ensuring the process by which users are granted access and certain privileges to systems, resources or information

This research is focus only on some security policy such as availability, Integrity, Confidentiality and access control and determined some of the security risk factors that affect these polices. Many security risk factors were reported in the literature [85-87]. These factors affect the grid computing in many ways, such as:

Availability: Threats to availability indicates the percentage of time, usually on a monthly basis, in which the Grid service supplied by the provider will be available [88]. The risk factors that threaten the availability are: Service Level Agreement (SLA) Violation: SLAs are contracts between service providers and users, specifying acceptable QoS levels. Cross-Domain Attack: Cross-Domain Attack in which the attacker compromises one site and can then spread his attack easily to the other federated sites. Job Starvation: Job starvation happens when the resources used by local job are taken away by stranger job scheduled on the host. Resource Failure: It is a failure if and only if one of the following two conditions is satisfied. a. Resource stops due to resource crash; and b. Availability of resources does not meet the minimum levels of QoS. Distributed Denial of Service (DDoS) attack, is an attack in which the computing power of thousands of compromised machines known as "zombies" are used for a target a victim. Zombies are gathered to send useless service requests, packets at the same time. Resource attacks: Illegal use of software or physical resources.

Integrity: refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change [48], and gives the assurance that the data received are not altered [89]. The Integrity related risk factors are: Integrity Violation, in computational grid, the grid program may be malicious and threatens the integrity of the resource. Privilege Attack: Many grids are designed to give remote users interactive access to a command shell so they can run their own application. However, giving the user access to a command shell within a predefined script or application is extremely dangerous, because when the users access the command shell they can obtain an extra privilege. Hosting Illegal content: By exploiting the leased nodes to send junk mail and host illegal content for others. The last factor is stealing the input or output or modifies the result of computation.

Confidentiality: refers to the protection of information from unauthorized disclosure. The risk factors related to Confidentiality are: Confidentiality Breaches: Indicates that all data sent by users should be accessible to only legitimate users. Data Exposure: The data protection issue is concerned about protecting the pre-existing data on the host that is associated with the grid system. Data Attacks: Illegal access to or modification of data. Man in the Middle Attack: When a message between peers is intercepted and modified. Sybil Attack: When a large number of malicious peers in the system are launched by an enemy, the peers in the system exchange the role of a resource provider and at each time one of them is scheduled, and then provide malicious service before it is replaced by another peer and be disconnected. And Privacy Violation: Compromising the passwords and security system, by exploiting the large computation power that the grid provides.

Access Control: The overall security of the grid is at stake if the authentication and authorization mechanisms are not strong enough to handle the access control properly. This will result in an unauthorized access to the resources. The risk factors related to Access Control are: Credential Violation: Credentials are tickets or tokens used to identify, authorize, or authenticate a user. Policy Mapping: Due to the spread of VO across multiple administrative domains with multiple policies, users might be concerned with how to map different policies across the grid. As a result of the grid's heterogeneous nature and its promise of virtualization at the user level, such mapping policies are a very important issue. Shared Use Threats: These issues are caused due to incompatibility between the attributes of grid users and conventional users of the computing resources that form the basis of the grid. The last factor is stealing software or the information contained in the database.

2.9 Related Works

2.9.1 Risk Assessment Associated with Grid Computing

A fundamental concept in risk assessment is the concept of Risk Exposure (RE), sometimes referred to as risk impact [90, 91]. RE is defined as:

$$RE = Prob\ (UO) * Loss\ (UO) \qquad (2.1)$$

Where Prob (UO) is the probability of an unsatisfactory outcome and Loss (UO) is the loss to the parties affected if the unsatisfactory outcome occurs. RE is then used to produce a ranked ordering of the risk items identified [38].

In consideration of risk assessment, the probability and the loss of an unsatisfactory outcome are assessed via application of the qualitative risk analysis technique. Boehm [92] proposes the use of a scale 0–10 in order to assess the probabilities and losses of unsatisfactory outcomes; such assessments are often the result of surveying several domain experts and are frequently subjective. A major source of risk. Keil*et al.* [93] adopt a three-phase Delphi survey in order to immediately identify the most important risk items, rather than simply identifying probability or loss associated with an unsatisfactory outcome. The survey identified that 11 risk factors as the most important.

The aim of this survey is to serve as a checklist of the most important risks for project managers to focus on. Wallace and Keil [94] map the 53 risk items identified in (Schmidt, 2001) into the four risk categories proposed in namely Customer Mandate, Scope & Requirements, Execution and Environment. A survey of 507 project managers, representing multiple industries, indicated the extent to which each risk item was present during their most recently completed projects. A scale from 1–7 is utilized so as to represent the presence of a risk item; higher numbers represent a higher presence and lower numbers a lower presence. The result identifies the risk associated with the Scope & Requirements and Execution categories to be the most critical, and that the Environment category is not of great importance.

The qualitative assessment of the 35 security risk items identified by ENISA in [95-97] is based on three scenarios: Small and Medium Enterprises (SME) migration to cloud computing services, the impact of cloud computing on service resilience and cloud computing in e-Government. The risk assessment is based on the ISO/IEC 27005:2008 information security risk management (*ISO/IEC, 2008)*; the risk is estimated on the basis of the likelihood of an incident scenario and the negative impact of that scenario; and the likelihood and the negative impact of a scenario are estimated using the following scale:

- 0, or Very Low,
- 1, or Low.
- 2, or Medium,
- 3, or High,
- 4, or Very High

The likelihood and the negative impact are determined by several domain experts. The risk is measured as the sum of the likelihood and the impact.

$$\textit{Risk} = \textit{likelihood} + \textit{impact} \qquad (2.2)$$

The risk is mapped to a simple risk rating: *Low Risk* 0-2, *Medium Risk* 3-5 and *High Risk* 6-8. This qualitative risk assessment is based on surveying several domain experts and might be subjective. Furthermore, there is some degree of uncertainty in terms of estimating the likelihood or the negative impact, which is, itself, a major source of risk.

The objective of the Consequence project [98] is to provide an information protection framework and to thereby identify the security risk in sharing data in a distributed environment. The risk items are used as a checklist of items to be addressed in the Consequence architecture, without any assessment of the probability and the negative impact of a risk item.

The SLA@SOI project [99] does not explicitly address risk assessment, although it does propose the utilization of a prediction service for estimating the probability of software failure, hardware availability and network failure in an attempt to evaluate the QoS. Notably, a number of limitations can be identified. The hardware availability is defined as:

$$\textit{Hardware Availability} = \textit{MTTF} / (\textit{MTTF} + \textit{MTTR}) \qquad (2.3)$$

This availability is for the entire lifecycle of the hardware, and it is not the probability that a hardware resource is available just at the point in time when it is required by service execution as assumed in the prediction service [100]. Another shortcoming is that the hardware might be unavailable owing to software failure or network failure; this means a single failure is considered twice in the analysis. Finally, the prediction service is not able to aggregate the probability of software and network failure to predict the probability of system failure as other components affecting the system failure are not addressed, i.e. hardware failure, electricity outage, air conditioning failure, etc.

Also, Risk Assessment (RA) has been extensively studied by many researchers[101-107]. A significant number of researchers [38, 41, 108] have proposed RA methods by producing framework for modeling risk, with focusing on enhancing accuracy of risk assessment by using different approaches implemented in RA. Assessing Risk has been done using artificial intelligence techniques in [43, 101, 104]. The authors in [101] proposed a genetic programming model to assess the risk by using multi expression programming (MEP), in which chromosome encrypts multi expressions. However this approach has difficulties in tuning the Genetic programming (GP) parameters such as Fitness function, population size, etc. Feng et al. [104] used the improved evidence theory as base of the presented model. The main aim of this approach is to minimize uncertainty that caused by experts due to their conflicting evidence. Although the model provides a good solution to deal with uncertain environment, however it requires domain experts' opinion reference at the evidence level individually [104]. The Hierarchical Neuro- Fuzzy learning for online Risk Assessment model, HiNFRA [102], provides integrated model in which neural network learning algorithms are used to set Fuzzy Inference System (FIS) variables. However the proposed approach needs significant care to represent the knowledge precisely [102].

Many researchers [46, 109] try to model risk dynamically. In constantly changing environments, Dynamic RA methodology is employed. Dynamic Risk Assessment approaches, implemented on the environment that change continuously and it depends on updating of the RA variables clarifying the IS and its environment regularly. However the presented models were need respectable effort due to its complexity and lake of acceptance accuracy [46]. Author name Charles Pak demonstrated Near Real Time Statistical Asset Priority Driven Risk Assessment (NRTSAPD) Methodology that arrange the organizational mission critical assets according to its priority [109]. However the updating of inputs in dynamic environment need computational effort to assess the asset and it importance again.

The Risk Assessment in Grid computing is a concept, which was presented at the two layers - Resource Provider (RP) and the broker by project of Assess Grid (AGP) [110]. AssessGrid project supported risk assessment and management for all three Grid actors; end user, broker, and resource provider. However AccessGrid did not provide any mechanism to determine the reason of the component failure and the influence of failure types on each other [111]. In the beginning the Risk modeling of the project was conceded at the Resource Provider that had taken into consideration the probabilistic as well as the possibilistic approaches. The Risk Assessment at the RP level in AGP was accomplished by the Bayesian model and provided the values of risk assessment at node level. his approach followed the same context as that of node as the work proposed in [111]. Assessing Risk in Grid computing has been done using stochastic processes; the risk assessment problem is tackled at the node level as well as the component level, all the suggested risk assessment models were built on historical failure data. Zadeh [112] proposed the possibilistic modeling which formed the basis for AGP at RP. On the other hand the work in AGP at broker level was intended to present a Broker. The Broker was introduced to facilitate the End User to communicate as well as negotiate with the RP. Also the level was designed so that it can make a selection of the relevant RP among many more existing works[40, 113]. The risk modeling is accountable in AGP at node level rather than the component level.

Authors in [37] provide details and direct discussion on the Risk Assessment within the Grids. Whereas the works found in [114, 115] cover indirect approaches available and it presents the Grid availability model and demonstrates the impact of Grid Failures on the performance of Grids. In these models the impact of risk assessment at any layer is not at all considered. An idea of risk assessment based on trust relationships along with the performance implications due to failures in the grids is done in [116, 117], also a general framework of grid failure has been provided. In [118] the authors use various reliability models to assess and evaluate on the basis of assuming Weibull distribution as the best fit model. This work suffered a limitation of requirement of aggregation at both the levels of component and node level. Further this concept was enhanced to improve reliability within the grids using stochastic model that extracted grid-trace-logs and thus enhanced the job resubmission strategy. The work in [119] is based on management of risks in grids with Markov Chain approach. This approach could deal with imperfect mechanisms associated with the risks but however failed to address the risk modeling due to grid failure. In [120] a study related to characteristics of disk failure and its patterns can be found but there is no discussion of estimation of risk analysis due to disk failure or any other components. Also this research does not address the Component Level Risk Assessment in Grids where the components could be either the Disks, CPU, computer software, computer memory. The types of grids also are not classified based on risk assessment that is whether the grids are replaceable or repairable.

Risk assessment has been studied extensively in the literature and there are many methodologies used in assessing risk such as FAIR (Factor Analysis for Information Risk) [121], and OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) [122]. The main drawback of the presented methodologies is that, they do not include the human factor as a risk factor [107]. Risk Assessment (RA) in Grid computing has been addressed by many researchers[29, 111, 123-125]. Although a significant number of researchers have proposed RA methods, the risk information in Grid computing is limited, due to the dependability of risk assessment efforts on the node or machine level [123]. Assessing Risk in Grid computing has been done using stochastic processes; the risk assessment problem is tackled at the node level as well as the component level, all the suggested risk assessment models were built on historical failure data.

Sangrasi et al. [123] provided a risk assessment model at the component level on the basis of Non-Homogeneous Poisson Process (NHPP). In [9], they used Grid failure data for the experimentation at the component level. Sangrasi[31] proposed a probabilistic risk model at the component level; the suggested model involves series and parallel models.

On the other hand, Alsoghayer et al. [12] used a probabilistic risk assessment method, where a sufficient failure data is available. They analyzed the failure data by using a frequentist approach. And they estimate the parameters of the distribution by utilizing the Maximum Likelihood Method. They take into consideration the failures that affect the whole system. Sangrasi et al. [124] extended the model proposed by Alsoghayer et al [12] and they introduced Risk assessment aggregation model build at the node level based on R-out-of -N model. The proposed model provides the risk estimates for any number of chosen nodes and estimates the risk for those failures. The provided model is built on assumption that when all the nodes fail the SLA fails. However the main drawback of the proposed model is that it is not applicable to all values of time in the given scenario.

The probability of resource failure plays a significant role in Risk Assessment process. However the main drawback of the provided probability models that highlighted in literature is that, all provided models are built on unrealistic assumption that the resource failure represent poisson process [126]. Alsoghayer et al. [126] proposed a mathematical model, by using historical and discrete time analytical model (Markov model), to predict the risk of resource failure in Grid environment. However most of proposed methods [123, 126] do not address the key issue of security risk that threaten the Grid environment. A significant amount of the literature on Grid computing addresses the problem of risk assessment by providing hybrid model [39].

Carlsson [125] developed a framework for resource management in Grid computing by utilizing the predictive probabilistic approach. They introduced the upper limit of failures number and approximated the likelihood of successful of a specific computing task. They used a fuzzy nonparametric regression technique to estimate the possibility distribution of the future number of node failures. The proposed model is utilized by resource provider to get alternative risk assessments.

Christer et al. [127] provided a model for assessing the risk of an SLA for a computing task in a Grid environment based on node failures that have spare resources available. The provided hybrid model is constructed based on a probabilistic and possibilistic technique. The constructed hybrid model takes into account the possibility distribution for the maximal number of failures derived from a resource provider's observation. However the proposed model focuses on node failure and ignores other factors that may cause a violation of the SLA. However the proposed methods addressed risk assessment in Grid with the aspect of resource failure. In our work we addressed the risk assessment in Grid computing in context of the security aspect.

2.9.2 Existing Feature Selection for Prediction Technique Associated with Grid Computing

Feature selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept [128]. The terms features, variables, measurements, and attributes are used interchangeably in the literature. Selecting the appropriate set of features is extremely important since the feature set selected is the only source of information for any learning algorithm using the data of interest.

A goal of feature selection is to avoid selecting too many or too few features than is necessary. If too few features are selected, there is a good chance that the information content in this set of features is low. On the other hand, if too many (irrelevant) features are selected, the effects due to noise present in (most real-world) data may overshadow the information present. Hence, this is a tradeoff, which must be addressed by any feature selection method.

Prediction on risk assessment is addressed via many researchers such as [44, 102, 129]. Distributed monitoring is helpful in early detection of planned and coordinated attacks.

The proposed model in [129] make use of predicting attacker tools versus Infrastructure for network computing, the model provides applicable solution by using a sampling algorithm that integrated with attacks simulator which is planned automatically. Then a security matrix, that characterize the weakest machine in the network, is computed based on attacker tools [129] .

Kjetil et al. [44] demonstrated risk based on distributed monitoring of intrusion attempts, one step forwards foretelling of such trial, and online risk assessment using fuzzy inference systems. The IPS block a suspect traffic flow by throwing away dubious information packets. However the proposed model depends on information gain from IDS, which add communication burden between IDS and it needs respectable care for deploying in the network to be monitoring efficiently. The Fuzzy online RA for Distributed Intrusion Prediction and Prevention Systems (DIPPS) was proposed to forecast possible intrusion beside revealing and blocking obtrusion [45].

In [45], a distributed Intrusion detection system (DIDS) is used to perform DIPPS with extensive real time traffic monitoring and online RA. Using a Hidden Markov Model (HMM) that reflects how the attacker and network interact, led to modeling and forecasting the next step of the attacker. Table 2.2 summarizes different approach in RA models with Strengths and Weaknesses of each technique.

Table2.2: Comparison of Risk Assessment Models

RA Methodology	Dynamic and online Techniques [132]	Soft Computing Techniques [47]	Predictive Approaches [48]
Strengths	-It provides the management with quick and easy approach to sense the existing risk situation Give precise result by regular updates. -It works near real time that leads to minimize the time that the system be in vulnerable state.	- It provides acceptance simplicity, - It can deal with uncertainty - Make good model for RA in rapid and efficient manner	- Capable to deal with imprecise , incomplete and uncertain information. Effective for real time applications
Weaknesses	-It is too difficult to Combine and integrate all information system to get the asset value.	- Difficulty in tuning the GP (Genetic programming) parameters such as Fitness function, population size, etc - No distinct way to formulate human knowledge as knowledgebase	-using simulation do not give a chance to gather information from realistic scenarios - lack the intuitive visualization - Cause a DoS.

2.9.3 Existing Security Issues in Grid Computing

Over the past years, there are many researchers focusing on the security issues in grid computing [23, 130, 131]. Chakrabarti et al. [23] addressed the security issues by classifying them into three levels. The first level is host level issue, which includes data protection issue and job starvation. In data protection issue, the main concern is protecting the pre-existing data of the host that is associated with the grid system. While, job starvation happens when the resources used by local job are taken away by stranger job scheduled on the host. In the second level (architecture level), the following issues are addressed: policy mapping, denial of services (DOS), resource hacking, information security, authentication, Integrity and confidentiality. The third level is the credential level that is categorized into: credential repositories which are responsible for user's credential storage, and the second is a credential federation system that supports managing credentials among multiple systems and regality. Butt et al. [132] mentioned that sharing of resource in grid environment involve execution of unreliable code from arbitrary users, which cause security risks such as securing access to shared resources.

In addition, the authors addressed two possible situations that can occur in grid environment and has the power to affect both the program executing in shared resource and the integrity of the resource. In the first scenario, the resource may be malicious that can affect the programs using these resource, or the grid program may be malicious which affect the integrity of resource. Chakrabarti [131], investigated a taxonomy of grid security issues that is composed of three categories. The first class addressed the architectural issues, which include data confidentiality, integrity and authentication. While the second category is security issues coupled with infrastructure such as data protection, job starvation, and host availability. The most critical security threats found in the grid infrastructure is malicious service disruption. The third class addressed the security issues that are related to management which include credentials management, and trust management. Some researchers [133, 134] analyzed the security threats in on-demand grid computing. Authors analysis is built on trust relationship between grid actors. They provided three different levels that address the security issues in on-demand computing and each level has possible forms of misuse. Level 1 addresses the threats that arise between users and solution producer. Whereas, level 2 provides the threats made by solution producer to use the resource provider in a bad manner, resulting in possible type of misuse. At the third level resource provider posed threats to users and solution producer caused different type of misuse. Smith et al. [134] addressed the security issues related to on demand grid computing by categorizing it to three types:

internal versus external attacks, software attacks, and privilege threats and shared use threats, these threats threaten traditional grid as well. Cody et al. [135] reported the most common vulnerabilities in the three different types of grid system. Each of the three identified grid systems has vulnerabilities common to them. The first type is computational grid, where the grid architecture is responsible for resource allocation to gain computing power to solve complex problems on high performance servers. The most popular vulnerability is node downfall that diminishes the functionality of the system. This could happen when the program contains infinite loops. In the second type of grid system called data grid, the main focus of grid architecture here is on storage and offering access to large amount of data across multiple organizations.

The possible risk that is associated with this type is overwrite or data corruption that occurs when user override their obtainable space. Denial of Service attack (DoS) is the most widespread attack that threatens the service grid that is considered as the third type of grid system. Kar et al. [86] provided vulnerabilities of grid computing in context of Distributed Denial of Service (DDoS) attack. Using spoofed IP address by attackers make DoS attacks so hard to detect, especially in large distributed system like grid, where it becomes more complicated. He presented four types of grid intrusions: unauthorized access, misuse, grid exploit, and host or network specific attacks. Kussul [136] addressed the most significant security threats for a utility based reputation model in grid. Based on the resource behavior observed in the past, the reputation can be seen as quality and reliability expected from that resource. The authors mentioned nine types of attack according the reputation model: Individual malicious peers, Malicious collectives, Malicious collectives with camouflage, Malicious spies, Sybil attack, Man in the middle attack, Driving down the reputation of a reliable peer, Partially malicious collectives, and Malicious pre-trusted peers. Hassan et al. [137] proposed the problem of Cross Domain Attack (CDA). Authors viewed that when a grid node is compromised it is so difficult to determine it, due to the existence of different administrative domains collaborating with each other, each with multiple nodes. In this case, the attack is likely propagated to another organization’s network that is part of the grid network, resulting in cross-domain attack. Carlsson [39] addressed the problem of Services Level Agreements

(SLAs) in Grid. The authors mentioned that a resource provider (RP) in grid computing offers resources and services to other Grid users based on agreed service level agreements (SLAs). The research problem that they have addressed is formulated as follows: the RP is running a risk of violating SLA if one or more of the resources offered to prospective customers will fail when carrying out the tasks.

Three types of failures were addressed by Lee [138], the process failure, which is expanded into two types that are Process stop failure and a starvation of process failure. Processor failure is the second type of failure which is further categories into a processor crash (Processor stop failure) and a decrease of processor throughput due to burst job (Processor QoS failure) while the third type of failure is a network failure that is classified into a network disconnection and partition (Network disconnection failure) and a decrease of network bandwidth due to communication traffic (Network QoS failure).

2.10 Summary

The literature reviewed consists of a number of salient themes, frameworks, models, architectures and approaches important for this study. This chapter discussed grid computing service models, types of grid computing, grid computing characteristics, grid uses, and security goals in grid computing. This chapter also has considered risk assessment and discussed the types of methods for risk assessment. Finally, existing feature selection for prediction technique associated with grid computing and existing security Issues in grid computing are also highlighted.

CHAPTER 3

RESEARCH METHODOLGY

3.1 Introduction

This Chapter presents the research methodology for our proposed risk assessment based on prediction technique of grid computing data modeling. The data set and the evaluation strategies used are also discussed. This research shall be carried out in five steps as illustrated in the following research methodology phases (Figure. 3.1). A methodology is a formal approach to solve the security problem based on a structured sequence of procedures. Using a methodology ensures a rigorous process, and increases the likelihood of achieving the desired final objective. This chapter discusses the research approach that is employed to achieve the objectives listed in Section 1.11. The chapter provides the research approach in Section 3.2.While its subsections providedthe methodology phases. The conclusion and summary of the chapter is presented in Section 3.3.

3.2 Research Methodology Phases

This research is conducted by employing the phases described in the following subsections. Figure 3.1 shows the research methodology phases carried out in this work.

Currently, there is a lack of a formal risk assessment based prediction technique model for collaborative grid computing environment, and there are no fast and hard rules on how to construct the proposed model. The investigation of the problems and then analyzed the construction of the proposed model takes into account the problems identified from the result of the model simulation. This is very important to make sure the proposed model is based on the objectives and the limitations.

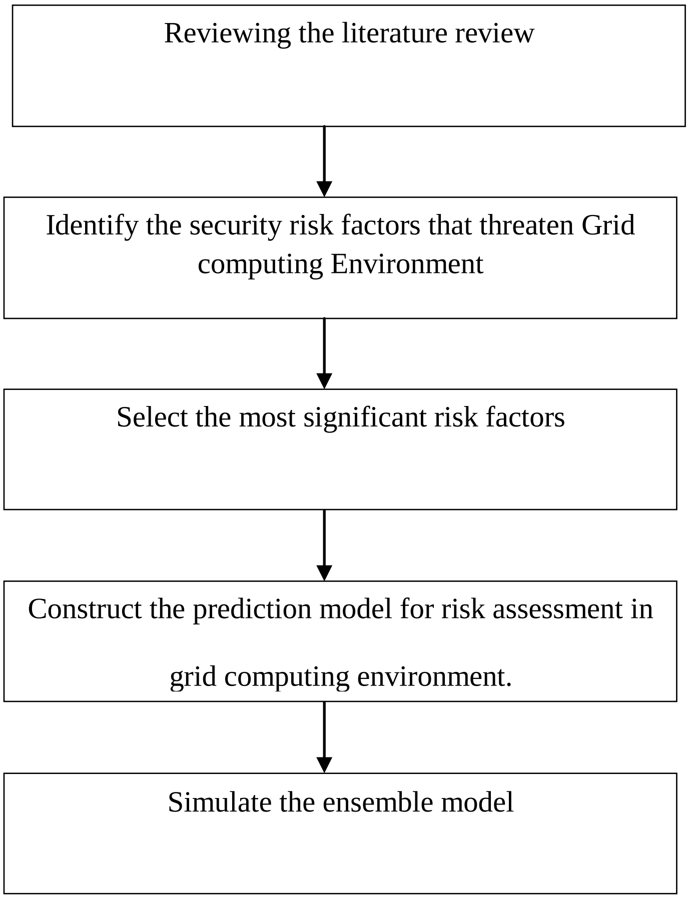

Reviewing the literature review
Identify the security risk factors that threaten Grid computing Environment
Select the most significant risk factors
Construct the prediction model for risk assessment in grid computing environment.
Simulate the ensemble model

Figure 3.1. Research Methodology Phases

3.2.1 Phase 1: Reviewing the Literature Review

This phase has been carried out as detailed in Chapter 2. In addition to Identification of risk factors associated with grid environment were explored and analyzed in order to find out the influence of these factors to security measures in grid computing environment. Furthermore, risk assessment models provided in grid computing were discussed. Challenges associated with security issues in grid environment are given special attention in this work since it is important to consider the security risk factors in addressing the role of risk assessment in making grid more reliable.

3.2.2 Phase 2: Identify the security risk factors that threaten Grid computing Environment

This phase is detailed in Chapter 4 to satisfy the first objective. In the process of identifying the security risk factors, reviewing the security risk factors was carried out. In addition, the corresponding numeric range to each factor was assigned. First we explore the security problems in grid computing and review the risk factors that are highlighted in literature. Total of 31 factors is found in the covered literature. Due to the appearance of some factors under different names, we identified only 20 risk factors that have influence to security measures in grid computing environment.

3.2.3 Phase 3: Select the most significant risk factors

Feature selection is a pre-processing technique aimed to reduce the dimensionality and remove the irrelevant features to increase the accuracy of prediction algorithms. It focus on choosing a subset of features that represents the concept ideally and sufficiently [139].

3.2.3.1 Feature Selection using machine learning algorithms

Starting with 20 risk factors, that was identified from the previous phase, we used WEKA [140] platform to find out the feature subset that contributes in assessing risk in grid computing. In this work, we adopt filter techniques to implement feature selection because they do not use the prediction algorithm. They are usually fast and therefore suitable for use with large datasets. Additionally, they are easily applicable to various prediction algorithms [139].

We used Correlation based Feature Selection (CFS Sub Eval), as evaluation function that evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Also, we used ReliefF Attribute Evaluator (ReliefF Attribute Eval), that evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Different search methods are applied such as Ranker, Evolutionary Search, Best First Search and Exhaustive Search for feature selection. The Filter methods apply feature-ranking function to select the best feature. A relevance score on the basis of a sequence of examples is assigned to the input feature by the ranking function. Then features with the highest rank are selected [140].

Filter-based feature ranking techniques (rankers) arrange the features, without relying on any prediction algorithm and the best features are chosen from the rank list [141]. Evolutionary Search algorithms rely on aggregated learning process within a population of individuals; each individual denotes a search point in the space of possible solutions to a given problem [142] . The main characteristic of evolutionary search is the fitness function, which has a value that expresses the performance of an individual so the individual can be evaluated and compared with other individuals. Best first search is a method that saves all attribute subsets that were evaluated before, and terminates when the performance starts to drop. The attribute subsets are arranged according to the performance measure; therefore an earlier configuration can be reviewed. Exhaustive search is a complete search that leads it to the optimal solution based on the identified evaluation criteria. Exhaustive search grantees that all reachable nodes are visited in the same level and then proceeding to the next level of the tree, so the possible moves in the search space are examined regularly [143].

After implementing the different attribute selection filter methods such as Relief Attribute Evaluation and Correlation based Feature Selection Subset Evaluator (CFS Subset Eval), we obtained 8 different sub datasets with different search methods.

3.2.3.2 Feature Selection using approximation Tool

Flexible Neural Tree (FNT) has been used as a approximation tool for feature selection, which gave more accurate and clear features selection because it assigned each feature a score according to (3.2) that determined the significance level of the feature (input variable) to the prediction of risk. A clear understanding of risk variables significance level served the objective of giving priority to managing risk variables by the administrator who manages grid computing environment. FNT feature selection was based on evolutionary process and the selection is automatic.

Our objective is to find significant input features. In other words, we determined significant risk factors. We calculated the score of risk factors A_j, that is score of j-th as follows:

$$Score(A_j) = \frac{\sum_i^M \left(fm_i \times \mathbb{I}(A_j)\right)}{M}, \tag{3.1}$$

wherefm_i is the fitness of model i, M is total number of models, and function $\mathbb{I}(A_j)$ is an identity function that returns 1 if attribute (risk factor) $A_j$is selected by model m_i, otherwise it returns 0.Once we calculate the score of all attributes, i.e., all

N attributes (here $N$is 20), we calculated the final score by normalizing their values as follows:

$$Score(A_j) = \frac{Score(A_j)}{max_{j=1\ toN}(A_j)} \quad (3.2)$$

where max is a function that returns maximum value.

3.2.4 Phase 4: Construct the prediction model for risk assessment in grid computing

In Supervised learning, the relationship between the input features and the target or output feature is represented by a structure known as a model. In a prediction problem, where we have dataset with n many independent variables X and a dependent variable Y, an approximation model finds relationship between independent variables X and a dependent variable. In the previous phase, the data is filtered to remove irrelevant and redundant features and to improve the quality.

3.2.4.1 Machine learning Algorithms for building a prediction risk model

To accomplish this phase, we divided the datasets into training and testing data with different percentages to investigate the effectiveness of data splitting.

We used prediction algorithms avaiable in WEKA to predict risk in grid computing, eight algorithms give good performance. Fllowing are the algorithms used to build the prediction model for risk assessment in grid computing environment:

Isotonic Regression algorithm (Isoreg)

Isotonic regression is a regression method that uses the weighted least squares to evaluates linear regression models [144].

Instance Based Knowledge (IBk) Algorithm

Instance Based Knowledge uses the instances themselves from the training set to represent what are learned, and be kept. When an unseen instance is provided the memory is searched for the training instance.

Randomizable Filter Classifier (RFC) Algorithm

This method used an arbitrary classifier on data that has been passed through an arbitrary filter. Like the classifier, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure [140].

Extra Tree Algorithm (Etree)

This method is an extremely randomized decision tree that uses another randomization process. At each node of an extra tree, partitioned rules are depicted randomly, then on the basis of a computational score the rule that proceed well is selected to be linked with that node [145].

Bagging Algorithm

In bagging, a classifier model is learned with every training set, which has been classified as tuples. Bagging is the most common method that synchronously processes samples. It merges the various outputs of learned predictors into a single computational model, that results in improved accuracy [146].

Ensemble Selection (EnsmS) Algorithm

The principle of ensemble methodology is to combine a set of models that solve the same original mission, with the aim of achieving more accurate and reliable estimates than that achieved from a single model [146].

Random Subspace (RsubS) Algorithm

Random subspace method utilizes random subsets of the available features to train the individual classifiers in an ensemble. Random Subspace is a random combination of models [147].

Random Forest (Rforest) Algorithm

Random forests create a lot of classification and regression trees, by recursively using partitioning, and then combining the results. Utilizing a bootstrap sample of the training data each tree can be created [147].

3.2.4.2 Adaptive Neuro fuzzy Inference System (ANFIS)for building a prediction risk model

We used an adaptive neuro fuzzy inference system (ANFIS) for modeling risk prediction in computational grid environment. ANFIS is a fuzzy inference system learned using neural network type learning methods. By using a hybrid learning procedure, ANFIS can construct an input-output mapping based on both human-knowledge as fuzzy '*if-then'* rules and approximate membership functions from the stipulated input-output data pairs. ANFIS learning employs a hybrid method consisting of back-propagation for tuning the parameters associated with input membership and least-squares-estimation for tuning the parameters associated with the output parameters [148].

Researchers excessively used ANFIS in many significant research problems, such as industry, financial, weather-prediction, health, etc.[149, 150]. Beghdad el al. [151] used combination of ANFIS and clustering process applied on the CPU Load time series to predict values of CPU load. Their proposed model achieves significant improvement and outperforms the existing CPU load prediction models reported in literatures. In [152], ANFIS was used to predict average air temperature while authors in [148] used ANFIS to predict roughness surface in ball end milling aluminum.

3.2.4.3 Flexible Neural Tree (FNT)

Flexible Neural Tree (FNT), has been used for predicting and identifying risks in a grid-computing environment. We developed an effective prediction model that can predict risk in grid computing environment. A wide range of applications accept artificial neural network (ANN) as the most convenient tool for the approximation [153]. Thus, it becomes a universal approximator. ANN performance heavily relies on its structure, parameters, and activation-functions (squashing function) [153] optimization. Researchers have investigated various ways in the past to optimize the individual components of ANN using evolutionary procedure [157]. Chen et al.[158] proposed a model called flexible neural tree (FNT) that addressed ANN optimization in all of its components, including structure and parameters and does automatic feature selection. FNT was conceptualized around a multi-layered feed-forward neural network to build a tree-based model, where network structure and parameters were optimized by using meta-heuristic optimization algorithms (the nature inspired stochastic algorithms for function optimization).

3.2.5 Phase 5: Simulate the ensemble model

Ensemble learning is a process that integates or combines a set of models obtained as a result of applying learning process to a given problm [156]. In regression problem the aim of ensemble is to improve the prediction performance that can be obtained from any algorithm individually . Two main phases to conduct ensemble, the first phase is generation phase, which is accomplished in the fourth phase. The gneration approcach can be homogeneous, if all generated models were generated with the same induction algorithm. Otherwise it can be heterogenous.The Second phase is integration phase, which can be done using combination or selection method.

3.2.5.1 Meta Learning Ensemble

Meta learning has been developed in the field of data mining to aid experts in selecting the best algorithms to be used with certain datasets. Meta level learning accumulates knowledge about the learning process itself, and finds a relation between problem domains and learning strategies [157]. Utilizing Meta schemes available in WEKA we found out possible combinations of ensemble, using vote and multischeme.

Vote: In vote combining schema each algorithm has the same weight. A prediction of an unseen instance is performed according to the class that obtains the highest number of votes . Based on vote the final predictor is conducted using a combination rule. In this work we adopted the average of probability as a combination rule.

Multischeme: Using the performance on the training data, which is measured based on mean-squared error (regression); the classifier among several classifiers is selected.

3.2.5.2 FNT Ensemble

A collective decision with consensus of many members is better than a decision of an Individual. Hence, ensemble of many models (predictors) may offer the most general solution to a problem [158]. There are two components in ensemble system [156] construction: (1) Construction of as diverse and as accurate models as possible. (2) Combining the models using a combination rules. To construct diverse and accurate models, we use the following techniques: (a) Training models with different sets of data, the algorithm Bagging is an example [159]; (b) Training models with a different set of input features, the algorithm Random Sub-space is an example; and (c) Training models with different set of parameters. Once many models are constructed with high diversity and accuracy, then we need to combine them for a collective decision.

3.3 Summary

This Chapter discussed the research methodology that is carried out in five phases. In Phase 1 the research plans and methods are investigated and our proposed model is formulated. The proposed model is based on synthesized component extracted from earlier literature review, and verified via a pre-survey (more details in Chapter 5). Identify the security risk factors that threaten grid computing environment are managed in Phase 2. In Phase 3the most significant risk factors using machine learning algorithms and approximation tool are selected. The prediction model for risk assessment in grid computing using machine learning algorithms, FNT and ANFIS is constructed in Phase 4. In Phase 5 our proposed ensemble model highlighted the ways of simulation.

CHAPTER 4

RISKFACTORSASSOCIATED TO GRID

This Chapter examines the definition of 'risk‘, introduces risk analysis and approaches used. As well as the security risk factors associated to grid computing are reported. A list of utilized risk factors in order to identify and assess risk in grid computing environment is presented.

4.1 Definitions of Risk

Risk has been defined using different expressions. For example, "Risk is the net negative impact of the exercise of vulnerability, considering both the probability and the impact of occurrence". Also it is defined as " The probability and magnitude of a loss, disaster, or other undesirable event" [160] or " a measure of the potential loss occurring due to natural or human activities" [161]. Regardless of the wording used to define the term, risk is nevertheless related to future events and their consequences. Principally, there is uncertainty associated with events and their consequences. The events uncertainty can be expressed by means of probability or likelihood, based on background knowledge [162].

Another important term linked to risk is ‘hazard‘. Hazard typically refers to the source of the risk, i.e. risk is created by a hazard. For example, a toxic gas that is a hazard to human health does not represent a risk unless humans are exposed to it.

4.2 Risk Analysis

Risk analysis is the process of characterizing and managing the potential events, which may lead to negative consequences or losses. As with the definition of risk, different disciplines often categories risk differently. Such categorization can be carried out based on the events causing the risk or the consequences of such events. However, Modarres [161] categorizes the risk into five broad categories: health, security, safety, financial and environmental.

Generally, there are three predefined approaches for analyzing risk: the quantitative approaches, the qualitative approaches and the mixed or hybrid approaches that combine the quantitative and the qualitative.

4.2.1 Quantitative Risk Analysis

The quantitative risk analysis attempts to estimate the risk in the form of the frequency of events and the magnitude of the losses or consequences. In this context, the 'uncertainty' associated with the estimation of the frequency of the occurrence of events and their consequences are characterized by using the probability concept.

Quantitative risk analysis is the preferred method when sufficient filed data, test data or other evidences exist so as to estimate the probability of events and magnitude of losses; however, quantitative risk analysis is complicated, time-consuming and expensive to conduct [161, 163, 164].

In the quantitative approach, risk disclosure has been handled as a probability function of a threat and the expected loss caused by weakness of the controls being applied [104], the drawbacks of quantitative approaches are: It require more time , costly to formulate and it is too complex, and the subjectivity of a weighted factors.

Quantitative risk analysis techniques includes: discriminate function analysis, Bayesian analysis, decision tree analysis, factor analysis, neural networks, risk matrix, risk register, and Mont Carlo analysis [165-167].

4.2.2 Qualitative Risk Analysis

Qualitative risk analysis is the most widely applied method, because it is simple and quick to perform. In this regard, the risk is estimated using a linguistic scale, such as low, medium and high. The frequency of events is measured by the likelihood of occurrence. In this type of analysis, a matrix is formed, which characterizes the risk in the form of the likelihood of events versus the potential magnitude of losses in qualitative scale. This type of analysis does not rely on actual data and probability treatment of such data; accordingly, it is far simpler to use and understand than the quantitative risk analysis, although it is extremely subjective [161, 163, 164]. The qualitative approaches does not employ the probability data, it only use the estimated potential loss by support of a linguistic scale producing a matrix. Qualitative risk analysis techniques include brainstorming, assumption analysis, interviews, hazard and operability studies, and risk mapping [168].

4.2.3 Mixed Risk Analysis

Due to the complexity in Information System, the use of one approach in isolation of other approach failed to formulate the RA model in efficient manner, so the combination of quantitative and qualitative approaches must be needed to give acceptance result in modeling Risk. Mixed risk analysis adopts a combination of qualitative and quantitative analyses.

This mix can occur in two ways: either the frequency of an event is measured qualitatively, but the consequences are measured quantitatively or vice versa; or both the frequency of an event and the consequences are measured using quantitative methods, but the policy setting and decision-making are reliant on qualitative methods [161].

4.3 Security Risk Factors in Grid Computing

This Section summarizes the risk factors in grid computing and highlighted the risk factors used. The total numbers of risk factors are summarized from the literature review are thirty one factors and only twenty factors are used in this research as summarized in Table 4.1 below.

4.3.1 Risk Assessment Factors

Many risk factors were reported in the literature by [8, 9, 21, 23, 35, 64, 66, 131, 169-173]. The descriptions of the factors are as below:

1. ***Service Level Agreement (SLAs) Violation:*** SLAs are contracts between service providers and users, specifying acceptable QoS levels. An important challenge in grid environment is how to monitor and enforce SLAs when many users share the same resources, especially because a key part of a grid environment's definition is that it provide nontrivial QoS [174]. A SLA can go through a number of stages once it has been specified. Assuming that the SLA is initiated by a client application, these stages include: discovering providers; defining the SLA; agreeing on the terms of the SLA (in addition to the penalties if the SLOs are not met); monitoring SLA violations; terminating an SLA; enforcement of penalties for SLA violation. Monitoring plays an important role in determining whether an SLA has been violated,

and determining the particular penalty clause that should be invoked as a consequence. Monitoring SLA violations begins once an SLA has been defined. Both the client and the provider must maintain a copy of the SLA. It is necessary to distinguish between an ‘agreement date’ (forming of an SLA) and an ‘effective date’ (subsequently providing a service based on the SLOs that have been agreed). For instance, a request to invoke a service based on the SLOs may be undertaken at a time much later than when the SLOs were agreed. During provision it is necessary to determine whether the terms agreed in the SLA have been complied with during provision. In this context, a monitoring infrastructure is used to identify the difference between the agreed upon SLO and the value that was actually delivered during service provisioning – which is ‘trusted’ by both the client and the provider.

2. ***Node downfall:*** Running applications on the Grid environment poses significant challenges due to the diverse failures encountered during execution. This could happen when the program executing in grid environment contains infinite loops, which result in diminishing the functionality of the grid.

3. ***Data overwrite or corruption:*** This occurs when the user of a data grid system overrides their obtainable space. Data corruption is a when data becomes unusable, unreadable or in some other way inaccessible to a user or application. Data corruption occurs when a data element or instance loses its base integrity and transforms into a form that is not meaningful for the user or the application accessing it. Although there are many factors that trigger data corruption, it is often enabled through an external virus stored or installed within the target computer or device. The virus overwrites the original data, modifies the code or permanently deletes it. Besides viruses, data corruption may also occur as a result of hardware or software malfunctions, errors and environmental calamities such as power outages, storms or other disasters. Data can be restored through a backup copy or it can be rebuilt using various data integrity checking algorithms.
4. ***Denial of Service attack (DoS):*** This involves sending large number of packets to a destination or a victim, which is flooded with traffic that is

difficult to handle or manage, to prevent legitimate users from accessing information or services. In a Distributed Denial of Service (DDoS) attack the computing power of thousands of compromised machines known as "zombies" are used to a target a victim. Zombies are gathered to send useless service requests, packets at the same time.

5. ***Quality of Services (QoS) Violation:*** Where access to certain services is denied, this can be as a result of congestion, delaying or dropping packets or resource hacking. Quality may mean different things for different users under different environments. In general, quality is a nonfunctional character such as performance, cost, security, reliability, etc., or a combination of them. In a shared network environment like Grid, there are several new issues related to QoS support that do not arise in a single computer system. The first issue is the variation of resource availability. This variation may be due to resource contention, dynamic system configuration, software or hardware failures, and other factors beyond the control of a user. The uncertainty of resource availability has a big impact on application quality. The second issue is parallel processing. The total workload of a large-scale application is often partitioned into smaller pieces, called subtasks. These subtasks are then allocated to resources in a distributed system to be processed concurrently. The challenge of parallel processing in a shared network environment lies on that the computing resources may be heterogeneous and have individual availability patterns. The third issue is non-centralized control. In a general Grid environment, the computing resources are autonomous. Local schedulers schedule local jobs and the Grid scheduler does not have the control of the local jobs. These new QoS issues make supporting QoS of Grid computing extremely challenge.

6. ***Cross-Domain Attack (CDA):*** In a single administrative domain networks there is only one security policy, which can be evaluated by the IT security manager. Grid networks are often composed of different administrative domains owned by different organizations dispersed globally. Such networks are referred to as multi-administrative domain networks. Each domain might have its own security policy and may not want to share its security data with

less-protected networks, making it more complex to ensure the security of such networks and protecting them from cross-domain attacks.

7. ***Data protection:*** The data protection issue is concerned about protecting the pre-existing data of the host that is associated with grid system. Data protection is how organizations, businesses or the government uses your personal information.

   Everyone responsible for using data has to follow strict rules called 'data protection principles'. They must make sure the information is:

   - used fairly and lawfully
   - used for limited, specifically stated purposes
   - used in a way that is adequate, relevant and not excessive
   - accurate
   - kept for no longer than is absolutely necessary
   - handled according to people's data protection rights
   - kept safe and secure

8. ***Job starvation:*** Job starvation happens when the resources used by local job are taken away by stranger job scheduled on the host. Starvation can occur in any system where the potential exists for a job to be overlooked by the scheduler for an indefinite period. In the case of backfill, small jobs may continue to be run on available resources as they become available while a large job sits in the queue never able to find enough nodes available simultaneously to run on. To avoid such situations, priority reservations are created for high priority jobs, which cannot run immediately. When making these reservations, the scheduler determines the earliest time the job could start, and then reserves these resources for use by this job at that future time.

9. ***Policy mapping:*** Due to the spread of VO across multiple administrative domains with multiple policies, users might be concerned with how to map different policies across the grid. As a result of the grid's heterogeneous nature and its promise of virtualization at the user level, such mapping policies are a very important issue.

A policy is defined as an administrator-specified directive which manages certain aspects of the desired outcome of interactions among users, applications and services in a distributed system. The policy provides guidelines for how the different system elements should handle the work resulting from different users and applications. As an example, a resource allocation policy may place limits on how much traffic within a network can be used by a class of applications, e.g., multicast traffic may not take more than 10% of total network capacity. Policies are applicable to different aspects of a distributed environment, including (but not limited to): access to network and system resources by different users/applications, restrictions on the set of applications accessible by a user, or support for different service levels and performance targets within the network or server.

10. ***Resource Failure or Allocation failure:*** It is a failure if and only if one of the following two conditions is satisfied. A. Resource stops due to resource crash B. Availability of resource does not meet the minimum levels of QoS [175].Service Level Agreements (SLAs) are introduced in order to overcome the limitations associated with the best-effort approach in Grid computing, and to accordingly make Grid computing more attractive for commercial uses. However, commercial Grid providers are not keen to adopt SLAs since there is a risk of SLA violation as a result of resource failure, which will result in a penalty fee; therefore, the need to model the resources risk of failure is critical to Grid resource providers. Essentially, moving from the best-effort approach for accepting SLAs to a risk aware approach assists the Grid resource provider to provide a high-level QoS. Moreover, risk is an important factor in establishing the resource price and penalty fee in the case of resource failure. Analyzing the Grid resources failures and understanding the performance of those resources with time is a key requirement for their modeling.

11. ***The malicious resource:*** The resource may be malicious that affect the programs using these resource. Grid technologies allows resource sharing among several entities, but selecting the most appropriate and secure resource to run a specific job remains one of its main problems. Most of the Grid

applications involve very large databases with highly secured data. Security requires the three fundamental services: authentication, authorization, and encryption. A grid resource must be authenticated before any checks can be done as to whether or not any requested access or operation is allowed within the grid. Once the grid resources have been authenticated within the grid, the grid user can be granted certain rights to access a grid resource. But within the grid application the one who uses the resource also needs reliable and secure services. So there is a need of reliable system, which ensures a level of robustness against malicious nodes. Users are able to submit jobs to remote resources and typically have no explicit control over the resources themselves. Therefore, mutually users and resources can be viewed as independent agents, having control of their own behavior. Since an individual cannot forecast the response of another to changing situations, this autonomy provides rise to inherent in security. So a better security mechanism is essential and crucial for secure and reliable communication in grid.

12. ***The integrity of resource:*** When a program executed in grid environment is malicious the integrity of resource is affected. In such a large distributed system, it is of particular importance to ensure data integrity. Since a Grid is usually a huge system, a lot of different users are using its resources. Some of these users may be malicious entities. Therefore, the risks of unauthorized alterations of data and information that are stored or processed on Grid resources, or even that are traveling on the Grid's network cannot be disregarded. Large amount of data are stored on Grid's resources. These data are used as input for distributed executions and/or are the results of these executions. It is crucial that these data are not illegitimately altered. Therefore, we have to ensure the integrity of these data. On another hand, the users need to have the guarantee that the asked executions are correctly processed. The jobs submitted on a Grid have to be executed in the right way with the proper input data. And in consequence, the resulting output data have to be reliable.

13. ***Securing access to shared resource:*** These issues are caused due to incompatibility between the attributes of grid users and conventional users of

the computing resources that form the basis of the grid. Grid computing technologies enable controlled resource sharing in distributed communities and the coordinated use of those shared resources as community members tackle common goals. These technologies include new protocols, services, and APIs for secure resource access, resource management, fault detection, communication, and so forth, that in term enable new application concepts such as virtual data, smart instruments, collaborative design spaces, and meta computations. Computational grids provide computing power by sharing resources across administrative domains. This sharing, coupled with the need to execute un-trusted code from arbiter users, introduces security hazards.

14. ***Exploit the leased nodes:*** To send junk mail and host illegal content for others.

15. ***Data attacks:*** Illegal access to or modification of data. With the growing use of Internet, attackers have become more and more active in identifying the flaw of the application or Operating system connected to the network protocols. Attackers are able to make the attacks on the network resources to make the damage on the network system or Application running in the system. Grid Computing is collection or heterogeneous resources or nodes from different organization globally. The need to support the integration and management of resources within VOs introduces challenging security issues. Grid system must detect the all type of attacks either it may be known or unknown or future attacks.

16. ***Meta data attacks:*** A malicious program can use operating system commands to acquire information about competitor's work. In some modern distributed file systems, data is stored on devices that can be accessed through the metadata, which is managed separately by one or more specialized metadata servers. Metadata is a data about data and it is structured information that describes, explains, locates, and makes easier to retrieve, use, or manage an information resource. The metadata file holds the information about a file stored in data servers.

17. ***Compromising the passwords and security system***, by exploiting the large computation power that grid provides.

18. ***Malicious acts*** such as faking of accounting, billing and malicious service disruption.

19. ***Store illegal software and data***; by utilizing the big store that grid offer [170, 176].

20. ***Download or steal account information from the resource provider***.

21. ***Hijack other nodes in the system***.

22. ***Stealing the software or the information contained in the database*.**

23. ***Altering the software or the information in the database allowing access to unauthorized parties.***

24. ***Stealing the input and output data*.**

25. ***Modifying the results.***
26. ***Resource attacks or resource hacking:*** Illegal use of software or physical resources such as CPU cycles and network bandwidth.

27. ***Credential level issues:*** Credentials are tickets or tokens used to identify, authorize, or authenticate a user. Secure operation in a Grid environment requires that applications and services be capable of supporting a variety of security functionality, such as credential conversion. Grid applications need to interact with other applications and services that have a range of security mechanisms and requirements. These mechanisms and requirements are likely to evolve over time as new mechanisms are developed or policies change. Grid applications must avoid embedding security mechanisms statically in order to adapt to changing requirements.

28. ***Man in the middle attack:*** When a message between other peers is intercepted and modified either by rewriting or changing reputation values, by a malicious peer. A man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. Man-in-the-middle attacks can be thought about through a chess analogy. Mallory, who barely knows how to play chess, claims that she can play two grandmasters simultaneously and either win one game or draw both. She waits for the first grandmaster to make a move and then makes this same move against the second grandmaster. When the second grandmaster responds, Mallory makes the same play against the first. She plays the entire game this way and cannot lose using this strategy unless she runs into difficulty with time because of the slight delay between relaying moves. A man-in-the-middle attack is a similar strategy and can be used against many cryptographic protocols. One example of man-in-the-middle attacks is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker must be able to intercept all relevant messages passing between the two victims and inject new ones.

29. ***Sybil attack:*** When a large number of malicious peers in the system is launched by an enemy, the peers in the system exchange the role of a resource provider and at each time one of them is scheduled, and then provides malicious service before it is replaced by another peer and be disconnected. In grids, such an attack is scarcely carried out in complete manner because the certificate authority should provide appropriate certificate to merge a resource into the grid system.

30. ***Privilege threats:*** Solution producer need more privilege to administer their system and to perform a security audit on all code submitted into the system.

31. ***Confidentiality:*** Indicates that all data sent by users should be accessible to only legitimate users. Confidentiality is roughly equivalent to privacy. Measures undertaken to ensure confidentiality are designed to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it: Access must be restricted to those authorized to view the data in question. It is common, as well, for data to be categorized according to the amount and type of damage that could be done should it fall into unintended hands. More or less stringent measures can then be implemented according to those categories.

4.3.2 Utilized Risk Factors

In the previous Section we presented a total of thirty one security risk factors that threaten the grid environment and affect the security measures, as appeared in the covered literature. In this Section we illustrate how the thirty one factors are reduced to twenty factors. Table 4.1 illustrates the complete set of factors that are used in our research without any modification or merging.

Table 4.1: The 9 Risk Factors as per literature without merge or modification

Utilized Factor Name	**Abbreviation**	**Technical Factors as reported in the literature**	**Number of the factor in the original list**
Distributed Denial of Services	DDoS	Distributed Denial of Services	(4)
Cross Domain Attack	CDA	Cross Domain Attack	(6)
Job Starvation	JS	Job Starvation	(8)
Policy Mapping	PM	Policy Mapping	(9)
Stealing Input Output	SIO	Stealing Input Output	(24)
Man in the middle attack	MMA	Man in the middle attack	(28)
Sybil attack	SA	Sybil attack	(29)
Shared Used Threat	ShUTh	Securing access to shared resource	(13)
Stealing Software	SS	Stealing Software	(22)

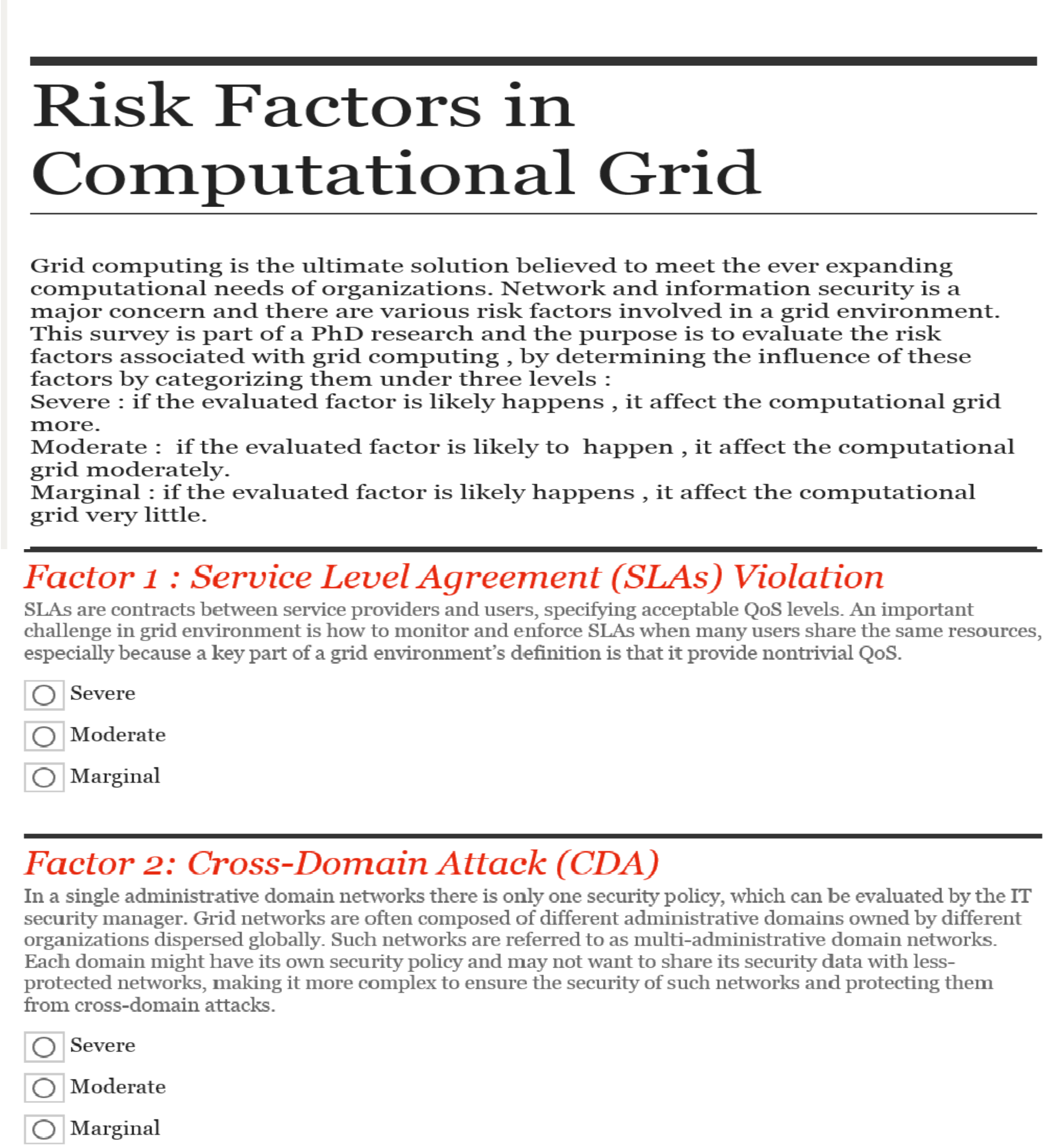
On the other hand, some factors are merged with other factors, because they are technically the same factor; according to the functionality and definition of each factor. For example, some of the merged factors are: Quality of services Violation can be considered as violation of Services Level Agreement, Node downfall and Resource Failure indicated the same factor, we used the factor Data Attack instead of Data Overwrite or Corruption and it is implicitly covered in the data protection issue. Table 4.2 shows the merged factors that impeded with included factors. The remaining Eleven Risk Factors are obtained by merging other factors that are technically same. Table 4.2 contains the merged factors.

Table 4.2: The11 merge Risk Factors

Revised Factors	**Abbreviation**	**Original Factors before merging with its number in the list**
Services Level Agreement Violation	SLAV	Services Level Agreement Violation (1), Quality of Services Violation (5)
Resource Attack	RA	Resource Attack (26), Hijack other node in the system (21)
Resource Failure	RF	Resource Failure(10), Node Downfall(2)
Data Attacks	DA	Data Attacks (15),Data overwrite or corruption (3), Meta Data Attacks(16)
Privilege Attack	PA	Privilege threats (30)
Confidentiality Breaches	CB	Malicious acts (18),Down load or steal account information (20), Altering the software in the database (23),Confidentiality (31)
Integrity Volition	IV	Malicious resource (11), The integrity of resource (12).
Data Exposure	DE	Data protection issue (7)
Credential Violation		Credential level issues (27)
Privacy Violation	PV	Compromising the password (17), Modify the results(25)
Hosting Illegal content	HIC	Store illegal software and data (19), Exploit the leased nodes to send junk mail (14)

4.4 The Survey about Risk Factors

A pilot study is conducted using a questionnaire survey (See Appendix A). Figure 4.1 shows the sample of our survey. We conducted an online survey with international experts to evaluate the risk factors associated with grid computing. We asked the experts to determine the influence of these factors by categorizing those under three levels: severe, moderate, and marginal. We received responses from 27 experts from nine different countries: France, Czech Republic, Romania, Canada, China, Malaysia, Brazil, Sudan and Ethiopia. All respondents agreed that all the predefined risk factors affect the grid in a major way. As a result total of twenty Factors were used in our research as illustrated in table 4.3.



Risk Factors in Computational Grid

Grid computing is the ultimate solution believed to meet the ever expanding computational needs of organizations. Network and information security is a major concern and there are various risk factors involved in a grid environment. This survey is part of a PhD research and the purpose is to evaluate the risk factors associated with grid computing , by determining the influence of these factors by categorizing them under three levels :
Severe : if the evaluated factor is likely happens , it affect the computational grid more.
Moderate : if the evaluated factor is likely to happen , it affect the computational grid moderately.
Marginal : if the evaluated factor is likely happens , it affect the computational grid very little.

Factor 1 : Service Level Agreement (SLAs) Violation

SLAs are contracts between service providers and users, specifying acceptable QoS levels. An important challenge in grid environment is how to monitor and enforce SLAs when many users share the same resources, especially because a key part of a grid environment's definition is that it provide nontrivial QoS.

○ Severe

○ Moderate

○ Marginal

Factor 2: Cross-Domain Attack (CDA)

In a single administrative domain networks there is only one security policy, which can be evaluated by the IT security manager. Grid networks are often composed of different administrative domains owned by different organizations dispersed globally. Such networks are referred to as multi-administrative domain networks. Each domain might have its own security policy and may not want to share its security data with less-protected networks, making it more complex to ensure the security of such networks and protecting them from cross-domain attacks.

○ Severe

○ Moderate

○ Marginal

Figure 4.1. Sample of Survey in Risk Factors in Computational Grid

So the final Twenty Factors that utilized to conduct the experiment and construct the model, are illustrated in Table 4.3

Table 4.3: Utilized Risk Factors

Risk Factor	Definition	Ref.
Services Level Agreement Violation (SLAV)	SLA represents an agreement between a service user and a provider in the context of a particular service provision.	[177]
Cross Domain Attack (CDA)	CDA in which the attacker compromises one site and can then spread his attack easily to the other federated sites.	[35]
Job Starvation (JS)	In JS,stranger job scheduled on the host use local (host) resources.	[131]
Resource Failure (RF)	It is a failure if: (i) resource stops because of resource crash; (ii) available resources does not meet the minimum levels of QoS.	[178]
Resource Attacks (RA)	It is illegal use of host resources by attacker.	[21]
Privilege Attack (PA)	User may gain excess privilege to accessing command shell,if grid computing allows access to command shell using a predefined scripts.	[21]
Confidentiality Breaches (CB)	Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, or legal action against the organization.	[23]
Integrity Violation (IV)	Integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change	[23]
Distributed Denial of Services (DDoS)	DoS attacks involve sending large number of packets to a destination to prevent legitimate users from accessing information or services.	[179]
Data Attack (DA)	In grid security, DA is a scheme in which malicious code is embedded in innocuous-looking data which (when executed by a program) plays out the intended destructive results.	[21]
Data Exposure (DE)	DE is other side of widespread connectivity in which (while improving productivity) makes it easier to obtain unauthorized to sensitive data	[21]
Credential Violation (CV)	Credentials are tickets or tokens used to identify, authorize, or authenticate a user. Comprise CV causes theft of user credentials.	[23]
Man in the Middle Attack (MMA)	MMA is an attack, where the attacker secretly relays and possibly alters the communication between two parties.	[131]
Privacy Violation (PV)	PV is the interference of a person's right to privacy by various means such as showing photos in public.	[28]
Sybil Attack (SA)	In Sybil attacks, few entities fakes multiple identities. So it is concern for the systems that rely upon implicit certification.	[131]
Hosting Illegal Content (HIC)	This can be done by exploiting the leased nodes.	[21]
Stealing Input or Output (SIO)	It is a way to steal the data received by the system or to steal data sent from it.	[21]
Shared Use Threats (ShUTh)	Incompatibility between the attributes of grid usersand conventional users causes ShUTh. Hence, no strict separation between participants.	[21]
Stealing or altering the Software (SS)	SS caused by unauthorized means entering altered data, false data, unauthorized data, or unauthorized instruction to a system.	[21]
Policy Mapping (PM)	Multiple administrative domains with multiple policies causes difficulty to users to map different policies across the grid	[23]

4.5 Summary

This chapter has considered risk assessment and discussed the types of methods for risk assessment. Examples of risk items identified are provided a list of utilized risk factors in order to identify and assess risk in grid computing environment.

CHAPTER 5

RISK ASSESSMENT MODEL IN GRID ENVIRONMENT

This Chapter introduces a theoretical model for risk prediction in computational Grid environment. With the use of a machine learning tools different computational models were built. Moreover, the model with high performance was selected and the reasons for selection are presented. In addition, the ensemble model for risk assessment is developed.

5.1 The model and data simulation

In the previous chapter, the details about risk factors was presented. After collecting total of 31 risk factors. We found that many factors has the same definition, but appears in literature with different name. Also Some Factors were aggregating under one Factor. As a result the number of factors was decreased to 20 risk factors, more details about this point were presented in the previous Chapter. At the next step we conducted an online survey with international experts to evaluate the risk factors associated with Grid computing.

Then we assigned a numeric range to each included factor depending on its concept and chance of occurrence. Based on expert knowledge and some statistical approaches, we then simulated 1951 instances based on a generic Grid environment. The original data set has a numeric data type and consists of 20 input attributes (risk factors), and one output (risk value).

Table 5.1: Risk factors (attributes)

Risk Factor	Abbreviation	Range
Service Level Agreement Violation	SLAV	[0-1]
Cross Domain Attacks	CDA	[1-3]
Job Starvation	JS	[0-1]
Resource Failure	RF	[0-1]
Resource Attacks	RA	[0-1]
Privilege Attack	PA	[0-1]
Confidentiality Breaches	CB	[0-2]
Integrity Violation	IV	[0-2]
DDoS Attacks	DDoS	[1-3]
Data Attack	DA	[0-2]
Data Exposure	DE	[1-3]
Credential Violation	CV	[0-1]
Man in the Middle Attack	MMA	[0-1]
Privacy Violation	PV	[0-2]
Sybil Attack	SA	[1-3]
Hosting Illegal Content	HIC	[0-1]
Stealing the Input or Output	SIO	[0-1]
Shared Use Threats	ShUTh	[1-3]
Stealing or altering the software	SS	[0-1]
Policy Mapping	PM	[1-3]

As shown in table 5.1, we have 20 risk factors (variables or attributes) as inputs and each factor has a assigned numeric range,. The output is the expected risk value for the given inputs. Each variable was granulated Then We tried to granulate as low, medium, high and very high. Depending on the input variable, each granule has a numerical range.Then we formulated 40 expert rules linking all the 20 input variables and output. As follows:

If (SLAV is low) and (CDA is low) and (JS is low) and (RF is low) and (RA is low) and (PA is low) and (CB is low) and (IV is low) and (DDos is low) and (DA is low) and (DE is low) and (CV is low) and (MMA is low) and (PV is low) and (SA is low) and (HIC is low) and (SIO is low) and (ShUTh is low) and (SS is low) and (PM is low) then (Risk is low).

If (SLAV is medium) and (CDA is medium) and (JS is medium) and (RF is medium) and (RA is medium) and (PA is medium) and (CB is medium) and (IV is medium) and (DDos is medium) and (DA is medium) and (DE is medium) and (CV is medium) and (MMA is medium) and (PV is medium) and (SA is medium) and (HIC is medium) and (SIO is medium) and (ShUTh is medium) and (SS is medium and PM is medium) then (Risk is medium) then (Risk is medium).

If (SLAV is high) and (CDA is high) and (JS is high) and (RF is high) and (RA is high) and (PA is high) and (CB is high) and (IV is high) and (DDos is high) and (DA is high) and (DE is high) and (CV is high) and (MMA is high) and (PV is high) and (SA is high) and (HIC is high) and (SIO is high) and (ShUTh is high) and (SS is high) and (PM is high) then (Risk is high).

As each variable had a different numerical range, then numerical values were assigned, based on the 40 expert rules. As follows:

Rule 1 : If (SLAV = 0) and (CDA = 1) and (JS = 0)and (RF = 0) and (RA = 0) and (PA = 0) and (CB = 0) and (IV = 0) and (DDoS= 1) and (DA = 0) and (DE =

1) and (CV = 0) and (MMA = 0) and (PV =0) and (SA =1) and (HIC =0) and (SIO =0) and (ShUTh =1) and (SS =0) and (PM =1) then (Risk = 0).

Rule 2: If (SLAV = 0.002) and (CDA = 1.0003) and (JS = 0.0002) and (RF = 0.0004) and (RA = 0.0005) and (PA = 0.001) and (CB = 0.003) and (IV = 0.003) and (DDoS = 1.0003) and (DA = 0.003) and (DE = 1.0003) and (CV = 0.0005) and (MMA = 0.001) and (PV =0.003) and (SA =1.0003) and (HIC =0.0005) and (SIO =0.0005) and (ShUTh =1.0003) and (SS =0.001) and (PM =1.0003) then (Risk =0.0105).

Rule 3: If (SLAV = 0.005) and (CDA = 1.0009) and (JS = 0.0009) and (RF = 00.0009) and (RA is 0.0009) and (PA = 0.005) and (CB = 0.0006) and (IV = 0.006) and (DDoS = 1.0009) and (DA = 0.006) and (DE = 1.0009) and (CV = 0.0009) and (MMA = 0.005) and (PV =0.006) and (SA =1.0009) and (HIC =0.0009) and (SIO =0.0009) and (ShUTh =1.0009) and (SS =0.005) and (PM =1.0009) then (Risk = 0.2).

Rule 4: If (SLAV = 0.009) and (CDA = 1.001) and (JS = 0.001) and (RF = 0.001) and (RA = 0.001) and (PA = 0.009) and (CB = 0.009) and (IV = 0.009) and (DDoS = 1.001) and (DA = 0.009) and (DE = 1.001) and (CV = 0.001) and (MMA = 0.009) and (PV =0.009) and (SA =1.001) and (HIC =0.001) and (SIO =0.001) and (ShUTh =1.001) and (SS =0.009) and (PM =1.001) then (Risk = 0.205).

Rule 5: If (SLAV = 0.01) and (CDA = 1.005) and (JS = 0.005) and (RF = 0.006) and (RA = 0.003) and (PA = 0.01) and (CB = 0.01) and (IV = 0.01) and (DDoS = 1.005) and (DA = 0.01) and (DE = 1.005) and (CV = 0.003) and (MMA = 0.01) and (PV =0.01) and (SA =1.005) and (HIC =0.003) and (SIO =0.003) and (ShUTh =1.005) and (SS =0.01) and (PM =1.005) then (Risk = 0.22).

Rule 6: If (SLAV = 0.05) and (CDA = 1.009) and (JS = 0.009) and (RF = 0.009) and (RA = 0.006) and (PA = 0.015) and (CB = 0.05) and (IV = 0.05) and

(DDoS = 1.009) and (DA = 0.05) and (DE = 1.009) and (CV = 0.006) and (MMA = 0.015) and (PV =0.05) and (SA =1.009) and (HIC =0.006) and (SIO =0.006) and (ShUTh =1.009) and (SS =0.015) and (PM =1.009) then (Risk = 0.29).

Rule 7: If (SLAV = 0.09) and (CDA = 1.01) and (JS = 0.01) and (RF = 0.01) and (RA = 0.009) and (PA = 0.019) and (CB = 0.09) and (IV = 0.09) and (DDoS = 1.01) and (DA = 0.09) and (DE = 1.01) and (CV = 0.009) and (MMA = 0.019) and (PV =0.09) and (SA =1.01) and (HIC =0.009) and (SIO =0.009) and (ShUTh =1.01) and (SS =0.019) and (PM =1.01) then (Risk = 0.3).

Rule 8: If (SLAV = 0.1) and (CDA = 1.06) and (JS = 0.019) and (RF = 0.07) and (RA = 0.01) and (PA = 0.12) and (CB = 0.1) and (IV = 0.1) and (DDoS = 1.06) and (DA = 0.1) and (DE = 1.06) and (CV = 0.01) and (MMA = 0.12) and (PV =0.1) and (SA =1.06) and (HIC =0.01) and (SIO =0.01) and (ShUTh =1.06) and (SS =0.12) and (PM =1.06) then (Risk = 0.36).

Rule 9: If (SLAV = 0.19) and (CDA = 1.09) and (JS = 0.02) and (RF = 0.09) and (RA = 0.05) and (PA = 0.17) and (CB = 0.14) and (IV = 0.14) and (DDoS = 1.09) and (DA = 0.14) and (DE = 1.09) and (CV = 0.05) and (MMA = 0.17) and (PV =0.14) and (SA =1.09) and (HIC =0.05) and (SIO =0.05) and (ShUTh =1.09) and (SS =0.17) and (PM =1.09) then (Risk = 0.39).

Rule 10: If (SLAV = 0.2) and (CDA = 1.1) and (JS = 0.06) and (RF = 0.1) and (RA = 0.09) and (PA = 0.19) and (CB = 0.19) and (IV = 0.19) and (DDoS = 1.1) and (DA = 0.19) and (DE = 1.1) and (CV = 0.09) and (MMA = 0.19) and (PV =0.19) and (SA =1.1) and (HIC =0.09) and (SIO =0.09) and (ShUTh =1.1) and (SS =0.19) and (PM =1.1) then (Risk = 0.4).

Rule11: If (SLAV = 0.23) and (CDA = 1.2) and (JS = 0.09) and (RF = 0.19) and (RA = 0.1) and (PA = 0.2) and (CB = 0.2) and (IV = 0.2) and (DDoS = 1.2) and (DA = 0.2) and (DE = 1.2) and (CV = 0.1) and (MMA = 0.2) and (PV =0.2) and (SA

=1.2) and (HIC =0.1) and (SIO =0.1) and (ShUTh =1.2) and (SS =0.2) and (PM =1.2) then (Risk = 0.5).

Rule 12: If (SLAV = 0.26) and (CDA = 1.3) and (JS = 0.1) and (RF = 0.2) and (RA = 0.15) and (PA = 0.25) and (CB = 0.25) and (IV = 0.25) and (DDoS = 1.3) and (DA = 0.25) and (DE = 1.3) and (CV = 0.15) and (MMA = 0.25) and (PV =0.25) and (SA =1.3) and (HIC =0.15) and (SIO =0.15) and (ShUTh =1.3) and (SS =0.25) and (PM =1.3) then (Risk = 0.6).

Rule 13: If (SLAV = 0.29) and (CDA = 1.4) and (JS = 0.19) and (RF = 0.25) and (RA = 0.19) and (PA = 0.29) and (CB = 0.3) and (IV = 0.3) and (DDoS = 1.4) and (DA = 0.3) and (DE = 1.4) and (CV = 0.19) and (MMA = 0.29) and (PV =0.3) and (SA =1.4) and (HIC =0.19) and (SIO =0.19) and (ShUTh =1.4) and (SS =0.29) and (PM =1.4) then (Risk = 0.7).

Rule 14: If (SLAV = 0.3) and (CDA = 1.6) and (JS = 0.2) and (RF = 0.29) and (RA = 0.2) and (PA = 0.3) and (CB =0.305) and (IV = 0.305) and (DDoS = 1.6) and (DA = 0.305) and (DE = 1.6) and (CV = 0.2) and (MMA = 0.3) and (PV =0.305) and (SA =1.6) and (HIC =0.2) and (SIO =0.2) and (ShUTh =1.6) and (SS =0.3) and (PM =1.6) then (Risk = 0.8).

Rule 15: If (SLAV = 0.33) and (CDA = 1.9) and (JS = 0.25) and (RF = 0.3) and (RA = 0.25) and (PA = 0.35) and (CB = 0.309) and (IV = 0.309) and (DDoS = 1.9) and (DA = 0.309) and (DE = 1.9) and (CV = 0.25) and (MMA = 0.35) and (PV =0.309) and (SA =1.9) and (HIC =0.25) and (SIO =0.25) and (ShUTh =1.9) and (SS =0.35) and (PM =1.9) then (Risk = 0.9).

Rule 16: If (SLAV = 0.35) and (CDA = 1.908) and (JS = 0.3) and (RF =0.36) and (RA = 0.29) and (PA = 0.39) and (CB is 0.31) and (IV = 0.31) and (DDoS = 1.908) and (DA = 0.31) and (DE = 1.908) and (CV = 0.29) and (MMA = 0.39) and

(PV =0.31) and (SA =1.908) and (HIC =0.29) and (SIO =0.29) and (ShUTh =1.908) and (SS =0.39) and (PM =1.908) then (Risk = 1).

Rule 17: If (SLAV = 0.39) and (CDA = 1.91) and (JS = 0.35) and (RF = 0.39) and (RA = 0.3) and (PA = 0.4) and (CB = 0.35) and (IV = 0.35) and (DDoS = 1.91) and (DA = 0.35) and (DE = 1.91) and (CV = 0.3) and (MMA = 0.4) and (PV =0.35) and (SA =1.91) and (HIC =0.3) and (SIO =0.3) and (ShUTh =1.91) and (SS =0.4) and (PM =1.91) then (Risk = 1.05).

Rule 18: If (SLAV = 0.4) and (CDA = 1.99) and (JS = 0.39) and (RF = 0.4) and (RA = 0.35) and (PA = 0.405) and (CB = 0.39) and (IV = 0.39) and (DDoS = 1.99) and (DA = 0.39) and (DE = 1.99) and (CV = 0.35) and (MMA = 0.405) and (PV =0.39) and (SA =1.99) and (HIC =0.35) and (SIO =0.35) and (ShUTh =1.99) and (SS =0.405) and (PM =1.99) then (Risk = 1.3).

Rule 19: If (SLAV = 0.45) and (CDA = 2) and (JS = 0.4) and (RF = 0.46) and (RA = 0.39) and (PA = 0.409) and (CB = 0.4) and (IV = 0.4) and (DDoS = 2) and (DA = 0.4) and (DE = 2) and (CV = 0.39) and (MMA = 0.409) and (PV =0.4) and (SA =2) and (HIC =0.39) and (SIO =0.39) and (ShUTh =2) and (SS =0.409) and (PM =2) then (Risk = 1.5).

Rule 20: If (SLAV = 0.49) and (CDA = 2.001) and (JS = 0.45) and (RF = 0.49) and (RA = 0.4) and (PA = 0.41) and (CB = 0.45) and (IV = 0.45) and (DDoS = 2.001) and (DA = 0.45) and (DE = 2.001) and (CV = 0.4) and (MMA = 0.41) and (PV =0.45) and (SA =2.001) and (HIC =0.4) and (SIO =0.4) and (ShUTh =2.001) and (SS =0.41) and (PM =2.001) then (Risk = 1.6).

Rule 21: If (SLAV = 0.5) and (CDA = 2.005) and (JS = 0.49) and (RF = 0.5) and (RA = 0.45) and (PA = 0.45) and (CB = 0.49) and (IV = 0.49) and (DDoS = 2.005) and (DA = 0.49) and (DE = 2.005) and (CV = 0.45) and (MMA = 0.45) and

(PV =0.49) and (SA =2.005) and (HIC =0.45) and (SIO =0.45) and (ShUTh =2.005) and (SS =0.45) and (PM =2.005) then (Risk = 1.7).

Rule 22: If (SLAV =0.53) and (CDA = 2.009) and (JS = 0.5) and (RF = 0.54) and (RA = 0.49) and (PA = 0.49) and (CB = 0.5) and (IV = 0.5) and (DDoS = 2.009) and (DA = 0.5) and (DE = 2.009) and (CV = 0.49) and (MMA = 0.49) and (PV =0.5) and (SA =2.009) and (HIC =0.49) and (SIO =0.49) and (ShUTh =2.009) and (SS =0.49) and (PM =2.009) then (Risk = 1.8).

Rule 23: If (SLAV = 0.55) and (CDA = 2.01) and (JS = 0.55) and (RF = 0.59) and (RA = 0.5) and (PA = 0.5) and (CB = 0.503) and (IV = 0.503) and (DDoS = 2.01) and (DA = 0.503) and (DE = 2.01) and (CV = 0.5) and (MMA = 0.5) and (PV =0.503) and (SA =2.01) and (HIC =0.5) and (SIO =0.5) and (ShUTh =2.01) and (SS =0.5) and (PM =2.01) then (Risk = 1.9).

Rule 24: If (SLAV = 0.59) and (CDA = 2.07) and (JS = 0.59) and (RF = 0.6) and (RA = 0.55) and (PA = 0.53) and (CB = 0.509) and (IV = 0.509) and (DDoS = 2.07) and (DA = 0.509) and (DE = 2.07) and (CV = 0.55) and (MMA = 0.53) and (PV =0.509) and (SA =2.07) and (HIC =0.55) and (SIO =0.55) and (ShUTh =2.07) and (SS =0.53) and (PM =2.07) then (Risk = 2).

Rule 25: If (SLAV = 0.6) and (CDA = 2.09) and (JS = 0.6) and (RF = 0.65) and (RA = 0.59) and (PA is 0.57) and (CB = 0.51) and (IV = 0.51) and (DDoS = 2.09) and (DA = 0.51) and (DE = 2.09) and (CV = 0.59) and (MMA = 0.57) and (PV =0.51) and (SA =2.09) and (HIC =0.59) and (SIO =0.59) and (ShUTh =2.09) and (SS =0.57) and (PM =2.09) then (Risk = 2.105).

Rule 26: If (SLAV = 0.63) and (CDA = 2.1) and (JS = 0.64) and (RF = 0.69) and (RA = 0.6) and (PA = 0.59) and (CB = 0.55) and (IV = 0.55) and (DDoS = 2.1) and (DA = 0.55) and (DE = 2.1) and (CV = 0.6) and (MMA = 0.59) and (PV =0.55) and (SA =2.1) and (HIC =0.6) and (SIO =0.6) and (ShUTh =2.1) and (SS =0.59) and (PM =2.1) then (Risk = 2.2).

Rule 27: If (SLAV = 0.65) and (CDA = 2.19) and (JS = 0.69) and (RF = 0.7) and (RA = 0.65) and (PA = 0.6) and (CB = 0.59) and (IV = 0.59) and (DDoS = 2.19) and (DA = 0.59) and (DE = 2.19) and (CV = 0.65) and (MMA = 0.6) and (PV =0.59) and (SA =2.19) and (HIC =0.65) and (SIO =0.65) and (ShUTh =2.19) and (SS =0.6) and (PM =2.19) then (Risk = 2.29).

Rule 28: If (SLAV = 0.69) and (CDA = 2.2) and (JS = 0.7) and (RF = 0.75) and (RA = 0.69) and (PA =0.65) and (CB = 0.6) and (IV = 0.6) and (DDoS = 2.2) and (DA = 0.6) and (DE = 2.2) and (CV = 0.69) and (MMA = 0.65) and (PV =0.6) and (SA =2.2) and (HIC =0.69) and (SIO =0.69) and (ShUTh =2.2) and (SS =0.65) and (PM =2.2) then (Risk = 2.3).

Rule 29: If (SLAV = 0.7) and (CDA = 2.4) and (JS = 0.705) and (RF = 0.79) and (RA = 0.7) and (PA = 0.69) and (CB = 0.7) and (IV = 0.7) and (DDoS = 2.4) and (DA = 0.7) and (DE = 2.4) and (CV = 0.7) and (MMA = 0.69) and (PV =0.7) and (SA =2.4) and (HIC =0.7) and (SIO =0.7) and (ShUTh =2.4) and (SS =0.69) and (PM =2.4) then (Risk = 2.4).

Rule 30: If (SLAV =0.705) and (CDA = 2.48) and (JS = 0.709) and (RF = 0.8) and (RA = 0.705) and (PA = 0.7) and (CB = 0.8) and (IV = 0.8) and (DDoS = 2.48) and (DA = 0.8) and (DE = 2.48) and (CV = 0.705) and (MMA = 0.7) and (PV =0.8) and (SA =2.48) and (HIC =0.705) and (SIO =0.705) and (ShUTh =2.48) and (SS =0.7) and (PM =2.48) then (Risk = 2.5).

Rule 31: If (SLAV = 0.709) and (CDA = 2.5) and (JS = 0.71) and (RF = 0.83) and (RA = 0.71) and (PA = 0.72) and (CB is 0.9) and (IV = 0.9) and (DDoS = 2.5) and (DA = 0.9) and (DE = 2.5) and (CV = 0.71) and (MMA = 0.72) and (PV

=0.9) and (SA =2.5) and (HIC =0.71) and (SIO =0.71) and (ShUTh =2.5) and (SS =0.72) and (PM =2.5) then (Risk = 2.6).

Rule 32: If (SLAV = 0.71) and (CDA = 2.59) and (JS =0.79) and (RF = 0.86) and (RA = 0.76) and (PA =0.76) and (CB = 1) and (IV = 1) and (DDoS = 2.59) and (DA = 1) and (DE = 2.59) and (CV = 0.76) and (MMA = 0.76) and (PV =1) and (SA =2.59) and (HIC =0.76) and (SIO =0.76) and (ShUTh =2.59) and (SS =0.76) and (PM =2.59) then (Risk = 2.609).

Rule 33: If (SLAV = 0.76) and (CDA = 2.6) and (JS = 0.8) and (RF = 0.89) and (RA = 0.79) and (PA = 0.79) and (CB = 1.2) and (IV = 1.2) and (DDoS = 2.6) and (DA = 1.2) and (DE = 2.6) and (CV = 0.79) and (MMA = 0.79) and (PV =1.2) and (SA =2.6) and (HIC =0.79) and (SIO =0.79) and (ShUTh =2.6) and (SS =0.79) and (PM =2.6) then (Risk = 2.65).

Rule 34 : If (SLAV = 0.79) and (CDA = 2.65) and (JS = 0.85) and (RF = 0.9) and (RA = 0.8) and (PA = 0.8) and (CB = 1.3) and (IV = 1.3) and (DDoS = 2.65) and (DA = 1.3) and (DE = 2.65) and (CV = 0.8) and (MMA = 0.8) and (PV =1.3) and (SA =2.65) and (HIC =0.8) and (SIO =0.8) and (ShUTh =2.65) and (SS =0.8) and (PM =2.65) then (Risk = 2.7).

Rule 35 : If (SLAV = 0.8) and (CDA = 2.69) and (JS = 0.89) and (RF = 0.905) and (RA = 0.85) and (PA = 0.86) and (CB = 1.4) and (IV = 1.4) and (DDoS = 2.69) and (DA = 1.4) and (DE = 2.69) and (CV = 0.85) and (MMA = 0.86) and (PV =1.4) and (SA =2.69) and (HIC =0.85) and (SIO =0.85) and (ShUTh =2.69) and (SS =0.86) and (PM =2.69) then (Risk =2.75).

Rule 36: If (SLAV = 0.85) and (CDA = 2.7) and (JS = 0.9) and (RF = 0.909) and (RA = 0.89) and (PA = 0.89) and (CB = 1.5) and (IV = 1.5) and (DDoS = 2.7) and (DA = 1.5) and (DE = 2.7) and (CV = 0.89) and (MMA = 0.89) and (PV =1.5) and (SA =2.7) and (HIC =0.89) and (SIO =0.89) and (ShUTh =2.7) and (SS =0.89) and (PM =2.7) then (Risk = 2.8).

Rule 37: If (SLAV = 0.89) and (CDA = 2.8) and (JS = 0.905) and (RF = 0.91) and (RA = 0.9) and (PA = 0.9) and (CB is= 1.7) and (IV = 1.7) and (DDoS = 2.8) and (DA = 1.7) and (DE = 2.8) and (CV = 0.9) and (MMA = 0. 9) and (PV =1.7) and (SA =2.8) and (HIC =0.9) and (SIO =0.9) and (ShUTh =2.8) and (SS =0.9) and (PM =2.8) then (Risk = 2.9).

Rule 38: If (SLAV = 0.9) and (CDA = 2.9) and (JS = 0.95) and (RF = 0.95) and (RA = 0.95) and (PA = 0.95) and (CB = 1.9) and (IV = 1.9) and (DDoS = 2.9) and (DA = 1.9) and (DE = 2.9) and (CV = 0.95) and (MMA = 0. 95) and (PV =1.9) and (SA =2.9) and (HIC =0.95) and (SIO =0.95) and (ShUTh =2.9) and (SS =0.95) and (PM =2.9) then (Risk = 2.95).

Rule 39: If (SLAV = 0.99) and (CDA = 2.99) and (JS = 0.99) and (RF = 0.99) and (RA = 0.99) and (PA = 0.99) and (CB = 1.99) and (IV = 1.99) and (DDoS = 2.99) and (DA = 1.99) and (DE = 2.99) and (CV = 0.99) and (MMA = 0. 99) and (PV =1.99) and (SA =2.99) and (HIC =0.99) and (SIO =0.99) and (ShUTh =2.99) and (SS =0.99) and (PM =2.99) then (Risk = 2.99).

Rule 40: If (SLAV = 1) and (CDA = 3) and (JS = 1) and (RF = 1) and (RA = 1) and (PA = 1) and (CB = 2) and (IV = 2) and (DDoS = 3) and (DA = 2) and (DE = 3) and (CV = 1) and (MMA = 1) and (PV =2) and (SA =3) and (HIC =1) and (SIO =1) and (ShUTh =3) and (SS =1) and (PM =3) then (Risk = 3).

Then we use simple linear interpolation to generate data between the 40 rules. We generated 50 data samples (between each rule) using appropriate step sizes (as the assigned values for different variables were different).

5.2 The model Components

We formulate an accurate model to predict risk in grid computing, different techniques were used to achieve this goal. Risk assessment is a set of techniques applied in order to investigate the probability of an event, and to thereby assess the effects/consequences. Risk assessment is the most important phase in risk management: if the risk assessment method is not conducted appropriately, the risk management will then fail to achieve its objectives. Selecting an assessment technique is not a straightforward task. The selection of a technique viewed as most suitable for application on a process should be determined after considering the following:

- availability of resources for analysis,
- size and complexity of the process which will be analyzed,
- phase in which the risk assessment will be considered in the process lifecycle, and
- Availability of information.

The authors also emphasize the importance of the data considered in the risk assessment. The data considered should be accurate, adequate, relevant, coherent, unbiased and valid. Regardless of the analytical techniques applied in the risk assessment, in order for the risk assessment process to be effective, various characteristics must be taken into account. The risk analysis must be:

- Timely: The process produces the best available data in an accepted time range.
- Cost-Effective: The cost of accomplishing a risk assessment is lower than the benefit gained from the results.
- Complete: The risk assessment must address all aspects of the process without taking anything for granted.
- Consistent: The methods used for evaluating risk and reporting threats must be consistent throughout the process.
- Understandable: The results must be communicated to the appropriate authority with clear terms.

The model was constructed using three experimental phases after the preprocessing phase.

5.3 Preprocessing and feature selection phase

In this phase, as a preprosseing phase, different attribute selection methods such as Relief Attribute Evaluation and Correlation based Feature Selection Subset Evaluator (CFS Subset Eval), were adoptedwith different search methods which are Evolutionary Search, Best first search, and Exhaustive search for the risk data set. As a result eight different sub datasets were obtained. Also we divided the datasets into training and testing data with different percentages to investigate the effectiveness of data splitting.

- A: Split 60 % training, 40% testing
- B: Split 70 % training, 30% testing
- C: Split 80 % training, 20% testing
- D: Split 90 % training, 10% testing

5.3.1 Feature Selection using approximation Tool

We determined the score of the risk factors involved in grid computing. We assigned score to each risk factor, score one (highest) to the risk factor that contributes most in predicting risk, and score zero (lowest) to the one that has no influence on risk prediction. In other words, we determined the risk factors that a grid-computing administrator needs to consider for improving resource distribution. In this stage, we proposed to use cross-validation, where we used flexible neural tree (FNT) model for prediction.

To determined significant risk factors, we assign a score to each risk factor A_j, using the formula:

$$Score(A_j) = \frac{\sum_i^M \left(fm_i \times \mathbb{I}(A_j)\right)}{M}, \qquad (5.1)$$

wherefm_i is the fitness of model i, M is total number of models, and function $\mathbb{I}(A_j)$ is an identity function that returns 1 if attribute (risk factor) $A_j$is selected by model m_i, otherwise it returns 0.Once we calculate the score of all attributes, i.e., all N attributes (here $N$is 20), we calculated the final score by normalizing their values as follows:

$$Score(A_j) = \frac{Score(A_j)}{max_{j=1\ toN}(A_j)} \qquad (5.2)$$

Wheremax is a function that returns maximum value?

5.4 Experimental First Phase

Regression algorithms are applied to the preprocessed data, using one of machine learning tool named WEKA (Waikato Environmentfor Knowledge Analysis). WEKA is a workbench designed to aid in the application of machine learning technology to real world data sets [180]. In this phase,the dataset were trained to build the prediction model to access risk in grid computing environment. More than 32 algorithm avialable in WEKA were applied to the data, Eight algorithms give a good performance according to performance measure which is Root Mean Square Error (RMSE).The algorithms used are detailed below:

5.4.1 Isotonic Regression algorithm (Isoreg)

Isotonic regression is a regression method that uses the weighted least squares to evaluates linear regression models [144].

5.4.2 Instance Based Knowledge (IBk) Algorithm

Instance Based Knowledge uses the instances themselves from the training set to represent what are learned, and be kept. When an unseen instance is provided the memory is searched for the training instance.

5.4.3 Randomizable Filter Classifier (RFC) Algorithm

This method used an arbitrary classifier on data that has been passed through an arbitrary filter. Like the classifier, the structure of the filter is based exclusively on the training data and test instances will be processed by the filter without changing their structure [140].

5.4.4 Extra Tree Algorithm (Etree)

This method is an extremely randomized decision tree that uses another randomization process. At each node of an extra tree, partitioned rules are depicted randomly, then on the basis of a computational score the rule that proceed well is selected to be linked with that node [145].

5.4.5 Bagging Algorithm

In bagging, a classifier model is learned with every training set, which has been classified as tuples. Bagging is the most common method that synchronously processes samples. It merges the various outputs of learned predictors into a single computational model, that results in improved accuracy [146].

5.4.6 Ensemble Selection (EnsmS) Algorithm

The principle of ensemble methodology is to combine a set of models that solve the same original mission, with the aim of achieving more accurate and reliable estimates than that achieved from a single model [146].

5.4.7 Random Subspace (RsubS) Algorithm

Random subspace method utilizes random subsets of the available features to train the individual classifiers in an ensemble. Random Subspace is a random combination of models [147].

5.4.8 Random Forest (Rforest) Algorithm

Random forests create a lot of classification and regression trees, by recursively using partitioning, and then combining the results. Utilizing a bootstrap sample of the training data each tree can be created [147].

5.5 Experimental Second Phase (ANFIS model)

In this phase, a hybrid approach, Adaptive Neuro Fuzzy Inference System (ANFIS) is adopted to build a risk prediction model. ANFIS model was constructed using grid partitioning and the membership function and consequent parameters were tuned using a hybrid learning process for 100 epochs. We used different membership functions to represent each input variable [181]. In this work, we used:

Trapezoidal membership function (Trapmf): Trapezoidal curve is a function of a vector x and depends on four parameters *a, b, c* and *d,* as given by:

$$f(x,a,b,c,d) = \max\left(\min\left(\frac{\mathrm{x-a}}{\mathrm{b-a}}, 1, \frac{\mathrm{d-x}}{\mathrm{d-c}}\right), \mathrm{o}\right) \qquad (5.3)$$

The parameter *a* and parameter *d* locate the "feet" of the trapezoid and the parameters *b* and *c* locate the shoulder.

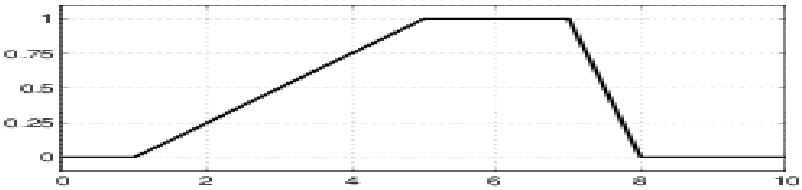


Figure 5.1. Trapezoidal membership function (Trapmf)

Triangular membership function (Trimf): The triangular curve is a function of a vector x and depends on three scalar parameters *a,b,* and *c,* given by:

$$\mathrm{f(x,a,b,c)} = \max\left(\min\left(\frac{\mathrm{x-a}}{\mathrm{b-a}}, \frac{\mathrm{c-x}}{\mathrm{c-b}}\right), \mathrm{o}\right) \qquad (5.4)$$

The parameter *a* and parameter *c* locates the "feet" of the triangle and the parameter b locates the peak.

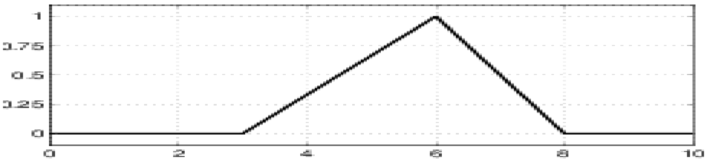


Figure 5.2. Triangular membership function (Trimf)

Generalized bell function (Gbell): Depends on three scalar parameters a, b and c, given by:

$$f(x, a, b, c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \qquad (5.5)$$

Where the parameter b is usually, positive and parameter c locates the center of the curve.

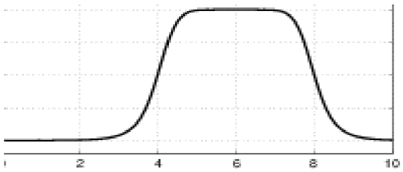


Figure 5.3. Generalized bell function (Gbell)

Gaussian membership function (Gaussmf): The symmetric Gaussian function depends on two parameters σ and c as given by:

$$f(x, \sigma, c) = e^{-\frac{(x-c)}{2\sigma^2}} \qquad (5.6)$$

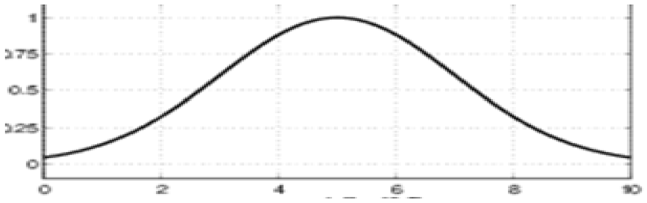


Figure 5.4. Gaussian membership function (Gaussmf)

5.6 Experimental Third Phase (FNT Model)

Flexible Neural Tree (FNT) was conceptualized around a multi-layered feed-forward neural network to build a tree-based model, where network structure and parameters were optimized by using meta-heuristic optimization algorithms (the nature inspired stochastic algorithms for function optimization).

We define FNT as a set of function-nodes and terminals, where the function-node indicates a computational node and terminals indicate a set of all input features. The function instruction set F and a terminal instruction set T for generating FNT model are described as:

$$S = F \cup T = \{+_2, +_3, +_4, \cdots, +_N\} \cup \{x_1, x_2, \cdots, x_n\} \qquad (5.7)$$

where $+_i$ $(i = 2,3, \cdots, n)$ indicates that a function-node can take i arguments, whereas, the leaf node (terminal node) receives no arguments. Figure 5.1 illustrates a function-node/computational-node of an FNT.

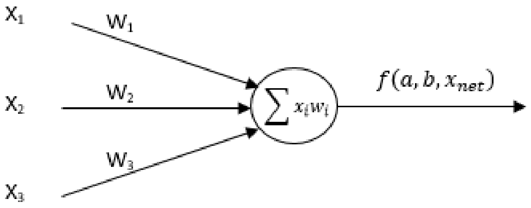


Figure 5.5.A computational node of a flexible neural tree

In Figure 5.5, the computational node $+_i$ receives $i$inputs through i connection weights (random real values) and two adjustable parameters/arguments a_i and $b_i$of the squashing (transfer) function, that limits the total output of the function-node within a certain range. A transfer-function used at the function-node is:

$$f(a_i, b_i, net_n) = e^{-\left(\frac{(net_n - a_i)}{b_i}\right)}, \qquad (5.8)$$

where net_n is the net input to the ith function-node also known as excitation of the node . It is computed as:

$$net_n = \sum_{j=1}^{n} w_j x_j \ , \qquad (5.9)$$

Where $j = 1,2,3 \ldots$ is the input to the i-th node. Therefore, the output of the i-th node is given as:

$$out_n = f(a_i, b_i, net_n) = e^{-\left(\frac{(net_n - a_i)}{b_i}\right)}, \qquad (5.10)$$

Figure 5.2 illustrates an example of a typical FNT. The root node of the FNT given in Figure 2 indicates the output of the entire tree-based model. The leaf nodes of the tree indicate the selected input feature and the edges of the tree indicate the underlying parameters (or the weights) of the model.

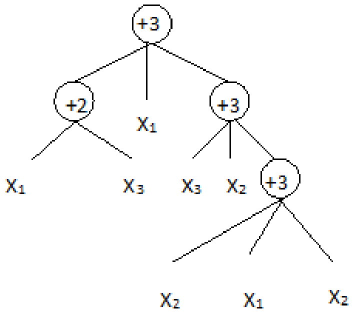


Figure 5.6. A typical FNT with instruction set $F = \{+_2, +_3,\}$ and $T = \{x_1, x_2, x_3\}$

Meta-heuristics are the stochastic algorithms that uses the exploration and exploitation of a given search space to find a global optimum solution for an optimization problem. Two different classes of meta-heuristic were used for the optimization for two different parts of the FNT: (a) genetic programming was used for the optimization of the structure [154]; and (b) swarm based meta-heuristics was used for the optimization of the parameters[182].

5.7 Construct the ensemble model

The idea of ensemble methodology is to build a predictive model by integrating multiple models. It is well-known that ensemble methods can be used for improving prediction performance. We Conduct an ensemble model by combing the prediction algorithms, using different meta learning methods such as vote and multischeme.

5.7.1 Meta Learning Ensemble

Utilizing Meta schemes available in WEKA we found out possible combinations of ensemble, using vote and multischeme. We constrcut the ensemble by using Four different algorithms were used as base predictors, each of which is the obtained outcome of learning algorithm applied to different dataset. We selected these methods based on the performance during the preliminary experiments.The base predictors were used for empirical testing of vote and multischeme.

5.7.2 FNT Ensemble

We used weighted mean combination method, where the weights for the models were computed by using meta-heuristic algorithm. In this work, we used, genetic algorithm for searching weights of the predictors (FNTs). Hence, ensemble output was computed as:

$$RMSE^{F'}(w_1, w_2, \cdots, w_k) = \sqrt{\frac{1}{N}\sum_i^N \sum_j^k w_j f_j (x_i - y_i)^2} \qquad (5.11)$$

Where x_i, andy_i denote the i-th input-target pair in the learning set that consists of total of Nsamples and $w_j$is the weight of j-th predictor.

5.8 Model validation

Test data is used to validate the performance of such models and to evaluate the performance of the model.

5.9 Summary

In a grid environment, the risk assessment is critical to ensure high security facilitation based on the way of its development. A risk assessment model to facilitate confidentiality, availability and integrity of collaborative grid environment is proposed in this chapter. To simulate the risk factors assessment model for collaborative grid security, the components on risk factors and grid environment are compiled from various literatures. An initial model of modified risk factors for collaborative grid environment is proposed. The relationships between these components are used to construct the questionnaire, which were tested in a pilot study. Item reliability is found to be poor and a few respondents and items were identified as misfits with distorted measurements. Some problematic questions are revised and some predictably easy questions are excluded from the questionnaire. Chapter six will explain how the model gives good performance and show the results.

CHAPTER 6

EXPERIMENTAL RESULTS AND ANALYSIS

This chapter starts with an experimental design and survey used. Next, we describe how requirements specified in the different experiments can be realized in our model and finally, we elaborate the results from our evaluation.

6.1 Experimental Design and Survey

The Previous Chapter handled the details about the Data simulation and the conducted survey for evaluating the risk factors. The data set was prepared with twenty factors to be the base attributes for all the experiments.

6.2 Feature Selection and Experimental Results

Data needs to be preprocessed before applying any data mining algorithm. In this phase, as a preprossing phase, the data is filtered to remove irrelevant and redundant features and to improve the quality. Different selection method is applied using WEKA platform and platform independent software tools to represent FNT model.

6.2.1 Feature Selection using Machine Learning Tool

Relief Attribute Evaluation and Correlation based Feature Selection Subset Evaluator (CFS Subset Eval), were adopted with different search methods which are Evolutionary Search, Best first search, and Exhaustive search for the risk data set. As a result eight different sub datasets were obtained, as illustrated in Table 6.1

Table 6.1: Attributes Selection Methods

Dataset	Evaluator	Search method	Selected Attributes	Number of Attributes
Original dataset	-	-	SLAV, CDA, JS, RF, RA, PA, CB, IV, DDoS, DA, DE, CV, MMA, PV, SA, HIC, SIO, ShUTh, SS, PM	20
1	ReliifF Attribute Evaluation	Ranker	DDoS, PM, DE, SA, ShUTh, HIC, CV, RA, SIO, CDA, RF, SLAV, JS, MMA, SS, PA, PV, IV, CB,DA	20
2	Reliff Attribute Evaluation	Ranker	DDoS, PM, DE, SA, ShUTh, HIC, CV, RA, SIO, CDA, RF, SLAV, JS, MMA, SS, PA, PV, IV	18
3	Reliff Attribute Evaluation	Ranker	DDoS, PM, DE, SA, ShUTh, HIC, CV, RA, SIO, CDA, RF, SLAV, JS, MMA, SS	15
4	Reliff Attribute Evaluation	Ranker	DDoS, PM, DE, SA, ShUTh, HIC, CV, RA, SIO, CDA, RF, SLAV	12
5	Reliff Attribute Evaluation	Ranker	DDoS, PM, DE, SA, ShUTh, HIC, CV, RA, SIO	9

6	CFS Subset Eval	Evolutionary Search	SLAV, JS, RA, CV, HIC, SIO	6
7	CFS Subset Eval	Best first search backward	CV, HIC, SIO	3
8	CFS Subset Eval	Exhaustive search	RA, CV, HIC	3

6.2.2 Feature Selection Using FNT

The mechanism of feature selection in the perspective of FNT model follows the sequence of, giving all feature same probability to be selected for formulating the FNT model. Then by an evolutionary procedure, the features which have more affect to the objective function will be enhanced and have high chance to select in the next generation.

We conducted our experiments using a platform independent software tools that realize the mentioned methodology. We processed our dataset using the developed software tool for constructing a predictive model and for understating the significance of input feature selection.

25 FNT models were constructed. Our objective is to find significant input features. In other words, we determined significant risk factors. We calculated the score of risk factors $\mathrm{A_j}$, that is score of j-th as follows:

$$\mathrm{Score}(\mathrm{A_j}) = \frac{\sum_{\mathrm{i}}^{\mathrm{M}}\left(\mathrm{fm_i} \times \mathbb{I}(\mathrm{A_j})\right)}{\mathrm{M}}, \qquad (6.1)$$

where$\mathrm{fm_i}$ is the fitness of model i, M is total number of models (here it is 25), and function $\mathbb{I}(\mathrm{A_j})$ is an identity function that returns 1 if attribute (risk factor) $\mathrm{A_j}$is selected by model m_i, otherwise it returns 0.Once we calculate the score of all attributes, i.e., all N attributes (here N is 20), we calculated the final score by normalizing their values as follows:

$$\mathrm{Score}(A_j) = \frac{\mathrm{Score}(A_j)}{\max_{j=1\text{ to }N}(A_j)} \qquad (6.2)$$

Where max is a function that returns maximum value. Hence, our calculated score is given in Figure 6.1.

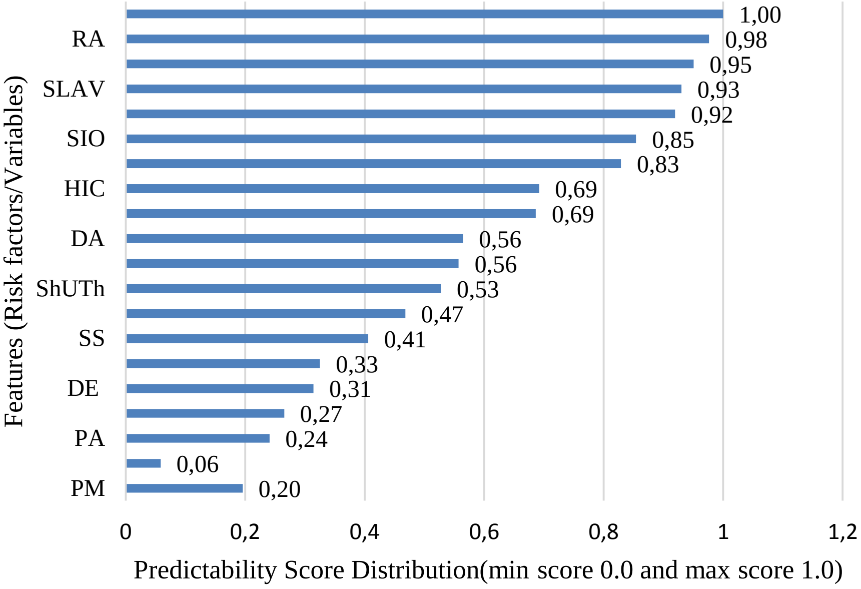


Figure 6.1. Predictability Score (influence of individual variables in risk assessment).

FNT for feature selection is used, which gave more accurate and clear features selection because it assigned each feature a score according to (2) that determined the significance level of the feature (input variable) to the prediction of risk. A clear understanding of risk variables significance level served the objective of giving priority to managing risk variables by the administrator who manages grid computing environment.FNT feature selection was based on evolutionary process and the selection is automatic. We obtained best features that attained predict ability score above 0.8: *SLVA, RF, RA, IV, CV, PV* and *SIO* (Figure 6.1).

6.3 Assessing Risk Algorithm and performance Evaluation

To provide accurate model to predict risk in grid computing, three data mining techniques were adopted and used to extract knowledge from risk dataset. Three predictive models, that able to predict risk for unseen data, were provided. The Following subsections provided details of the proposed models as well as performance measures.

6.3.1 Machine Learning based model

Utilizing WEKA platform, the experiments were performed. And the performance measures were calculated for all dataset using Correlation Coefficient (CC) and Root Mean Square Error (RMSE). As illustrated in Table 6.3, Isotonic Regression algorithm(IsoReg), IBK algorithm, Randomizable Filter Classifier algorithm (RFC) and the Extra tree algorithm (ETree) performed well for all the training and testing combinations and for the 9 different datasets (the original dataset in addition to 8 obtained dataset). All these algorithms exhibited the best performance in the case of all 9 datasets. It is noticed that the higher the percentage of training data (Dataset D) the better for achieving good results. However the empirical result shows that, the prediction algorithm required the least number of attributes (3 attributes only out of 20 attributes) to achieve high performance. The best result is accomplished with the Correlation Coefficient (CC) equal to 1 and the root mean squared error (RMSE) equal to 0.0015 for datasets 3 and 4. Table 6.2 reports the empirical results (for test data) illustrating the root mean squared error (RMSE) for the Nine datasets, since CC is equal to 1 for all dataset we didn't include it in the table.

Table 6.2: RMSE for 9 dataset with their higher performance algorithms

	Data Split	Original dataset	20 Attributes	18 Attributes	15 Attributes	12 Attributes	9 Attributes	6 Attributes	3 Attributes	3 Attributes
		RMSE								
IsoReg	A	0.0024	0.0023	0.0023	0.0023	0.0023	0.0023	0.0024	0.0023	0.0023
	B	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
	C	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018
	D	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
IBk	A	0.0023	0.0023	0.0023	0.0023	0.0023	0.0023	0.0022	0.0021	0.0021
	B	0.002	0.002	0.002	0.0019	0.002	0.0019	0.0019	0.0018	0.0018
	C	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0016	0.0016
	D	0.0017	0.0017	0.0017	0.0017	0.0017	0.0016	0.0016	0.0015	0.0015
RFC	A	0.0023	0.0023	0.0023	0.0023	0.0023	0.0023	0.0023	0.0022	0.0022
	B	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.0019	0.0019
	C	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018	0.0019	0.0017	0.0017
	D	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0018	0.0016	0.0016
Etree	A	0.0045	0.0048	0.0048	0.0046	0.0046	0.0046	0.3677	0.0045	0.0045
	B	0.0041	0.0041	0.0039	0.0039	0.0041	0.0038	0.0039	0.0038	0.0038
	C	0.0035	0.0035	0.0035	0.0037	0.0035	0.0033	0.0034	0.0033	0.0033
	D	0.0032	0.0031	0.0032	0.0031	0.003	0.0032	0.0031	0.0029	0.0029
	C	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161
	D	0.0153	0.0154	0.0154	0.0154	0.0154	0.0154	0.0153	0.0155	0.0155

On the other hand, Bagging, Ensemble Selection, Random subspace, and Random forest algorithms performed slightly well with the CC equal to 0.9999 and the RMSE varied according to the used algorithm and splitting of data. With Bagging and Random subspace algorithms the higher performance is achieved with 70% training and 30% testing, while Random forest gives the best performance with 60% training and 40% testing. Table 6.3 reports the empirical results (for test data) illustrating the root mean squared error (RMSE) for the Nine datasets, since CC is equal to 0.999 for all dataset we didn't include it in the table.

Table 6.3: RMSE for 9 dataset with less performance algorithms

	Data Split	Original dataset	20 Attributes	18 Attributes	15 Attributes	12 Attributes	9 Attributes	6 Attributes	3 Attributes	3 Attributes
		RMSE								
Bagging	**A**	**0.0128**	**0.013**	**0.013**	**0.013**	**0.013**	**0.013**	**0.0128**	**0.0128**	**0.0128**
	B	**0.0128**	**0.0127**	**0.0127**	**0.0127**	**0.0127**	**0.0127**	**0.0128**	**0.0126**	**0.0126**
	C	**0.013**	**0.0133**	**0.0133**	**0.0133**	**0.0133**	**0.0133**	**0.013**	**0.0132**	**0.0132**
	D	**0.0158**	**0.0159**	**0.0159**	**0.0159**	**0.0159**	**0.0159**	**0.0158**	**0.0159**	**0.0159**
EnsmS	**A**	**0.0164**	**0.0164**	**0.0164**	**0.0164**	**0.0164**	**0.0164**	**0.0164**	**0.0165**	**0.0165**
	B	**0.0157**	**0.0155**	**0.0155**	**0.0155**	**0.0155**	**0.0155**	**0.0157**	**0.0156**	**0.0156**
	C	**0.0161**	**0.0161**	**0.0161**	**0.0161**	**0.0161**	**0.0161**	**0.0161**	**0.0161**	**0.0161**
	D	**0.0153**	**0.0154**	**0.0154**	**0.0154**	**0.0154**	**0.0154**	**0.0153**	**0.0155**	**0.0155**
RsubS	**A**	**0.0157**	**0.0148**	**0.0154**	**0.0155**	**0.015**	**0.0173**	**0.0167**	**0.0161**	**0.0161**
	B	**0.015**	**0.0154**	**0.0169**	**0.0161**	**0.015**	**0.0155**	**0.0147**	**0.0157**	**0.0157**
	C	**0.0153**	**0.0162**	**0.0164**	**0.0155**	**0.015**	**0.0148**	**0.0168**	**0.0161**	**0.0161**
	D	**0.0173**	**0.0155**	**0.0175**	**0.0171**	**0.0175**	**0.017**	**0.017**	**0.0166**	**0.0166**
Rforest	**A**	**0.0129**	**0.0132**	**0.0131**	**0.0131**	**0.0127**	**0.0127**	**0.0124**	**0.0126**	**0.0126**
	B	**0.0142**	**0.0143**	**0.0143**	**0.0143**	**0.0129**	**0.0129**	**0.0127**	**0.0128**	**0.0128**
	C	**0.0133**	**0.0133**	**0.0134**	**0.0134**	**0.0125**	**0.0125**	**0.0124**	**0.0124**	**0.0124**
	D	**0.0158**	**0.0157**	**0.0157**	**0.0158**	**0.0145**	**0.0145**	**0.0143**	**0.0144**	**0.0144**

6.3.2 Adaptive Neuro Fuzzy Inference System Based Model

Fuzzy inference system is a process of using fuzzy logic for formulating a nonlinear mapping from input to output, where this system has three parts. (1) A rule base containing fuzzy rules, which are selected. (2) Data base, which defines membership functions applied for the fuzzy rules. (3) A logical system performing the way of inference based on the rules and facts.

The empirical result shows that, the prediction algorithm required the least number of attributes (3 attributes only out of 20 attributes) to achieve high performance.

In this part of experiment, to verify the efficiency of the proposed method, we used three features (CV, HIC, and SIO). To achieve the experimental result, different ANFIS parameters were tested as training parameters to maximize the prediction accuracy. Table 6.4 illustrates the ANFIS performance using different numbers of membership function (MF) shapes with different data splits. The lowest average testing error was obtained using Triangular Membership Function (Trimf) with Dataset that contains three attributes.

Table 6.4: ANFIS Performance for different membership functions (MF)

	Data Split	2 MF		3 MF		4 MF	
		RMSE					
		Train	Test	Train	Test	Train	Test
Trimf	A	0.0425	0.0433	0.0379	0.0381	**0.0139**	0.0146
	B	0.0424	0.0418	0.0325	0.0320	0.0143	**0.0137**
	C	0.0428	0.0424	0.0376	0.0374	0.0143	0.0141
	D	0.0428	0.0431	0.0372	0.0355	0.0144	0.0143
Gbellmf	A	0.0353	0.0357	0.0260	0.0271	0.0177	0.0195
	B	0.0355	0.0355	0.0255	0.0262	0.0189	0.0197
	C	0.0354	0.0353	0.0270	0.0263	0.0188	0.0198
	D	0.0352	0.0378	0.0222	0.0280	0.0184	0.0252
Guaussmf	A	0.0443	0.0444	0.0227	0.0250	0.0202	0.0217
	B	0.0409	0.0402	0.0229	0.0235	0.0216	0.0221
	C	0.0403	0.0409	0.0265	0.0282	0.0201	0.0214
	D	0.0401	0.0460	0.0235	0.0311	0.0188	0.0233
Trapmf	A	0.0398	0.0413	0.0381	0.0395	0.0290	0.0279
	B	0.0402	0.0406	0.0386	0.0389	0.0455	0.0453
	C	0.0396	0.0428	0.0379	0.0411	0.0451	0.0472
	D	0.0389	0.0507	0.0371	0.0497	0.0276	0.0255

Three risk factors and the ANFIS model was selected based on the minimum value of root mean square error equal to 0.0137, which is constructed using four triangular-shaped membership function for each input variable and linear membership function for output. Hence we have developed a risk prediction model for computational grid environment using ANFIS.

6.3.3 Flexible Neural Tree (FNT) prediction model

Several experiments with the parameter settings as per Table 6.5 are conducted. Since, the computation model mentioned is stochastic in nature, each instance of experiment offers distinct results in terms of accuracy and feature selection. We used RMSE to measure the accuracy, in other words, fitness of approximation model. Additionally, we use correlation coefficient to measure the correlation that tells the relationship between two variables (here, the two variables:

the actual output, and the models' output) reveals the quality of the constructed model.

Table 6.5: Parameter settings of the HFNT tool

	Parameter Name	Parameter Utility	Values
1	Tree Height	Maximum number of levels that a tree model can acquire during evolution.	5
2	Tree Arity	Maximum number of siblings a function-node can acquire during evolution.	4
3	Tree Node Type	Indicates the type of transfer-function a node can acquire during evolution.	Gaussian
4	GP Population	Number of candidates taking a part in the process of the evolution.	30
5	Mutation Probability	Probability that a candidate will take part in the mutation process to form a new candidate.	0.4
6	Crossover probability	Probability that a candidate will take part in the crossover process to form a new candidate.	0.5
7	Elitism	Probability that a fittest candidate will survive/propagate to the next generation.	0.1
8	Tournament Size	It indicates the size of the pool used for the selection of the candidates that will take part in evolutionary process.	15
9	MH Algorithm Population	The initial size of the swarm (population).	50
10	MH Algorithm Node Range	Defines search-space of transfer-function.	[0,1]
11	MH Algorithm Edge Range	Defines the search-space for the edges.	[-1.0,1.0]
13	Structure Iteration	Iteration of structure optimization.	100000
14	Parameter Iteration	Iteration of parameter optimization	10000

Using FNT 25 different models were constructed, we selected four highly accurate and divers FNT models for making ensemble. In Table 6.6, we present FNT model results over 10-fold cross validation dataset.

Table 6.6: FNT results based on 10 folds cross validation.

Exp.	Training		Test	
	RMSE	r	RMSE	r
1	0.03648	0.999	0.05861	0.998
2	0.04546	0.999	0.04952	0.998
3	0.04609	0.999	0.07277	0.931
4	0.05292	0.998	0.0835	0.907

6.4 Ensemble models

In this stage, we combine the four based prediction algorithms (IsoReg, IBK, RFC, and ETree) to conduct an ensemble using vote and multischeme as combination methods. We use nine different datasets with four different splitting categories for training and testing.

6.4.1 Ensemble with two and three base Prediction Algorithm

Table 6.7: RMSE of ensemble with Two base predictors

Combination Method	**Ensemble with 2 base predictors**									
	Base predictors	Original dataset	20 Attributes	18 Attributes	15 Attributes	12 Attributes	9 Attributes	6 Attributes	3 Attributes	3 Attributes
	RMSE									
Voting	**IsoReg** **IBK**	0.0014	0.0014	0.0014	**0.0013**	0.0014	**0.0013**	0.0014	0.0014	0.0014
	IsoReg **RFC**	**0.0013**	0.0014	0.0014	**0.0013**	0.0014	**0.0013**	**0.0013**	0.0014	0.0014
	IsoReg **ETree**	0.0017	0.0019	0.0019	0.0017	0.0017	0.0019	0.0017	0.0017	0.0017
	IBK **RFC**	**0.0013**	0.0014	0.0015	0.0014	0.0014	0.0015	0.0014	**0.0013**	**0.0013**
	IBK **ETree**	0.0018	0.0018	0.0018	0.0017	0.0018	0.0018	0.0017	0.0017	0.0017
	RFC **Etree**	0.0020	0.0019	0.0019	0.0018	0.0018	0.0019	0.0018	0.0017	0.0017
Multi-scheme	**IsoReg** **IBK**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
	IsoReg **RFC**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
	IsoReg **ETree**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
	IBK **RFC**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0016	0.0016	0.0015	0.0015
	IBK **ETree**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0016	0.0016	0.0015	0.0015
	RFC **Etree**	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0018	0.0016	0.0016

According to Table 4, comparing the possible combinations of two predictors for risk prediction process, the best performance is achieved with dataset 2, dataset 4, and dataset 5 with RMSE equal to 0.0013 and 90% for training and 10% for testing

Table 6.8: RMSE of ensemble with Three base predictors

Combination Methods	**Ensemble with 3 base predictors**									
	Base Predictors	Original dataset	20 Attributes	18 Attributes	15 Attributes	12 Attributes	9 Attributes	6 Attributes	3 Attributes	3 Attributes
		RMSE								
Voting	IsoReg IBK RFC	**0.0012**	0.0013	0.0013	0.0013	**0.0012**	0.0013	**0.0012**	0.0013	0.0013
	IsoReg IBK Etree	0.0014	0.0014	0.0015	0.0013	0.0014	0.0014	0.0013	0.0014	0.0014
	IsoReg RFC Etree	0.0014	0.0015	0.0015	0.0013	0.0014	0.0015	0.0013	0.0014	0.0014
	IBK RFC Etree	0.0015	0.0015	0.0015	0.0013	0.0015	0.0015	0.0014	0.0014	0.0014
	IsoReg IBK RFC	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
	IsoReg IBK Etree	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017

Multi-Scheme	IsoReg RFC Etree	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
	IBK RFC Etree	0.0017	0.0017	0.0017	0.0017	0.0017	0.0016	0.0016	0.0015	0.0015

Tables 6.7 and 6.8 show that vote works better than multischeme in all possible combinations of the four base predictors in all selected datasets. For combining the four base algorithms the best result is achieved is 0.0012 with original dataset and dataset with 3 attributes, with vote Meta methods.

6.4.2 Ensemble of FNT models

To construct the ensemble of FNT, we selected four highly accurate and divers FNT models for making ensemble. In Table 6.9, we present FNT model results over 10-fold cross validation dataset. We constructed an ensemble of FNT model that shows significant improvement over using individual FNT model.

Table 6.9: Ensemble of FNT model

Exp.	Training		Test		Ensemble weights
	RMSE	r	RMSE	r	
1	0.03648	0.999	0.05861	0.998	0.589584
2	0.04546	0.999	0.04952	0.998	0.359202
3	0.04609	0.999	0.07277	0.931	0.053049
4	0.05292	0.998	0.0835	0.907	0.000001
Ensemble	-	-	**0.0311**	**0.999**	-

6.4.3 Comparison between different prediction models output and actual output

Figure 6.2 illustrated the comparison between different risk factors prediction models output and actual output. The differentiation is based on prediction model. From Figure 6.2 we conclude that there is a significant performance by using prediction models for risk assessment.

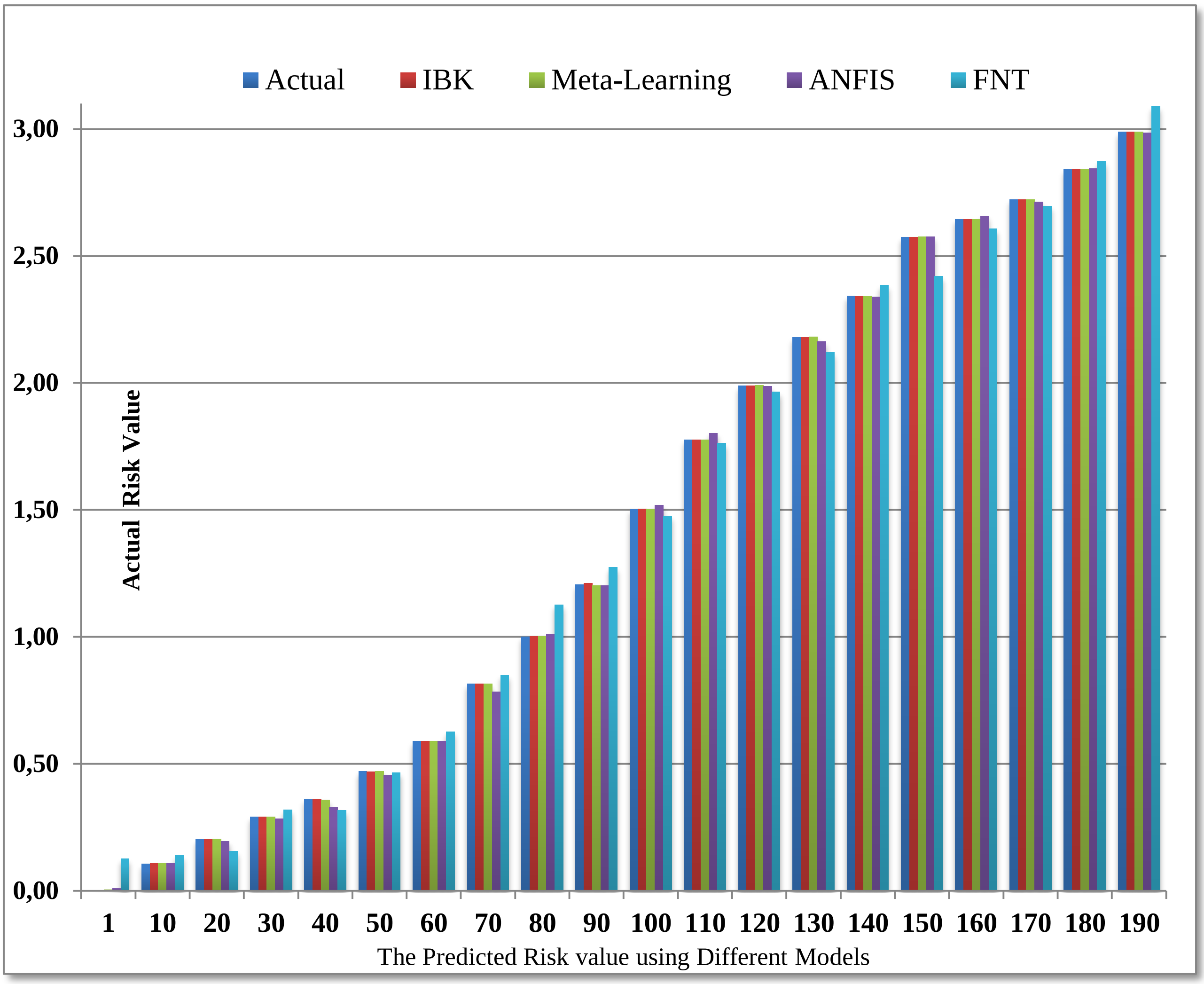


Figure 6.2. Comparison between different prediction models

6.5 Summary

In this chapter, the results observed from the prediction models and the simulating methods are described. The model is then evaluated based on the different prediction models. The model is validated by comparing the model with different prediction models. Finally, the use of the model to plan future research is presented.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

The work of this research had proposed three objectives concerning the prediction risk assessment to facilitate confidentiality, availability and integrity of grid computing. This Chapter summarizes the main findings of the research. Three main Objectives have been achieved and presented in three different chapters. The important results are summarized below:

1. In this research, we investigated the problem of data security in grid environment, to ensure the confidentiality, availability and integrity of users' data in the grid. We proposed a prediction risk model in computational grid environment. This model consists of risk factors and development of predictive data mining techniques in a computational grid environment. To formulate the proposed model for collaborative Computational Grid System (CGS) security, the components are compiled from various literatures. An initial model of modified components for collaborative grid environment is proposed. The relationships between these components are used to construct the questionnaire, which were tested in a pilot study. Item reliability is found to be poor. Few respondents and items were identified as misfits with

distorted measurements. Some problematic questions are revised and some predictably easy questions are excluded from the questionnaire.

2. When choosing a methodology for a problem, the consideration on the complexity of Methodologies is necessary. Methodologies, which propose large and complex models in their development phases, or methodologies with lots of dependencies between their models may be unsuitable for analyzing and designing a system. It can be concluded that by using machine learning methods and based on the characteristics of the grid environment, adding a security layer based on risk assessment supports faster and reliable security for grid computing.
3. In this research, the model based on prediction of risk factors to facilitate the confidentiality, availability and integrity of grid is simulated. To simulate the model, three different techniques are used for simulation. Our proposed model is evaluated using questionnaire; to determine which risk factor is associated to security policy is more successful. If the chosen risk factor is significant then directly the security policy is deemed useful to facilitate the security of grid. Our model is significant and deemed useful to facilitate the grid security upon acceptance of its significance.

7.2 Future Work

Further work can be carried out to generate more data to understand the security of grid for the different types of tools and different parameters. Some suggestions for future works are listed as follows:

1. Developers would be wise to design and develop their next generation system to be deployed in grid computing environment due to the fast evolution of

grid computing security in terms of number of users and number of grid applications, which makes them more complex and therefore more vulnerable to various kinds of complex grid attacks.

2. It is a great idea to develop a feedback model to acquire implicit knowledge from security professional teams to develop suitable criteria for reminding and recommending useful information to grid users. We hope that the proposed model will be a trigger for discussions leading to even more detailed and acceptable models in the area of grid computing security.

3. Several limitations are observed for the model evaluation. The validity of the availability policy may not be truly established on the basis of a single study. We shall need to exercise caution when generalizing findings. This is due to the fact that validation of measurement requires the assessment over different grid environments (external validity). However, this is not impossible due to all of grid environments have the same system and network architecture. More importantly, different grid architectures may have different security domains.

REFERENCES

1. Foster, I.K., Carl, The grid in a nutshell, in Grid resource management. 2004, Springer. p. 3-13.
2. Foster, I.K., Carl Tuecke, Steven, The anatomy of the grid: Enabling scalable virtual organizations. International journal of high performance computing applications, 2001. 15(3): p. 200-222.
3. Foster, I.K., Carl Nick, Jeffrey M Tuecke, Steven, The anatomy of the grid. Berman et al.[2], 2003b: p. 171-197.
4. Foster, I.K., Carl Nick, Jeffrey M Tuecke, Steven, The physiology of the grid. Grid computing: making the global infrastructure a reality, 2003a: p. 217-249.
5. Schwiegelshohn, U.B., Rosa M Bubak, Marian Danelutto, Marco Dustdar, Schahram Gagliardi, Fabrizio Geiger, Alfred Hluchy, Ladislav and D.L. Kranzlmüller, Erwin, Perspectives on grid computing. Future Generation Computer Systems, 2010. 26(8): p. 1104-1115.
6. Foster, I.Z., Yong Raicu, Ioan Lu, Shiyong. Cloud computing and grid computing 360-degree compared. in Grid Computing Environments Workshop, 2008. GCE'08. 2008: Ieee.
7. Chakrabarti, A.D., Anish Sengupta, Shubhashis, Grid computing security: A taxonomy. Security & Privacy, IEEE, 2008. 6(1): p. 44-51.
8. Butt, A.R., et al., Grid-computing portals and security issues. Journal of Parallel and Distributed Computing, 2003. 63(10): p. 1006-1014.
9. Cody, E., et al., Security in grid computing: A review and synthesis. Decision Support Systems, 2008. 44(4): p. 749-764.
10. Damoah, D., et al., Improving Security Measures on Grid Computing. International Journal of Computer Applications, 2013. 83(9): p. 6-11.
11. Kazemi, A., Review of Grid Computing Security and Present a New Authentication Method for Improving Security. 2014.

12. Rosmanith, H. and J. Volkert, Interactive techniques in grid computing: A survey. Computing and Informatics, 2012. 27(2): p. 199-211.

13. Schwiegelshohn, U., et al., Perspectives on grid computing. Future Generation Computer Systems, 2010. 26(8): p. 1104-1115.

14. Czajkowski, K., et al. Grid information services for distributed resource sharing. in High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on. 2001: IEEE.

15. Foster, I., What is the grid? a three point checklist, July 2002. ThreePoint-Check. pdf, 2006.

16. Humphrey, M. and M.R. Thompson, Security implications of typical grid computing usage scenarios. Cluster Computing, 2002. 5(3): p. 257-264.

17. Gupta, M. and G. Gupta, Security Requirements For Increasing Reliability In Grid Computing. International Journal of Engineering Computers & Applied Sciences, 2014. 3(12): p. 6-10.

18. Demchenko, Y., et al. Web services and grid security vulnerabilities and threats analysis and model. in Proceedings of the 6th IEEE/ACM international workshop on grid computing. 2005: IEEE Computer Society.

19. Krauter, K., R. Buyya, and M. Maheswaran, A taxonomy and survey of grid resource management systems for distributed computing. Software: Practice and Experience, 2002. 32(2): p. 135-164.

20. Vieira, K., et al., Intrusion detection for grid and cloud computing. IT Professional Magazine, 2010. 12(4): p. 38.

21. Smith, M., et al., Countering security threats in service-oriented on-demand grid computing using sandboxing and trusted computing techniques. Journal of Parallel and Distributed Computing, 2006. 66(9): p. 1189-1204.

22. Foster, I., The grid: a new infrastructure for 21st century science. Phys. Today, 2002. 55(ANL/MCS/JA-42173).

23. Chakrabarti, A., A. Damodaran, and S. Sengupta, Grid computing security: A taxonomy. IEEE Security & Privacy, 2008(1): p. 44-51.

24. Hu, X. and M. Zhou, Research on the Information Security Problems in Cloud Calculation's Environment. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013. 11(12): p. 7316-7323.

25. Nagaratnam, N., et al., The security architecture for open grid services. Open Grid Service Architecture Security Working Group (OGSA-SEC-WG), 2002: p. 1-31.

26. Broadfoot, P.J. and A.P. Martin, A critical survey of grid security requirements and technologies. Programming Research Group, PRGRR-03-15, Oxford University Computing Laboratory, 2003.

27. Foster, I. and C. Kesselman, The Grid 2: Blueprint for a new computing infrastructure. 2003: Elsevier.

28. Naqvi, S. and M. Riguidel, Threat model for grid security services, in Advances in Grid Computing-EGC 2005. 2005, Springer. p. 1048-1055.

29. Sangrasi, A.D., Karim. Component level risk assessment in grids: A probablistic risk model and experimentation. in Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on. 2011: IEEE.

30. Foster, I.K., Carl Nick, Jeffrey M and S. Tuecke, The physiology of the grid. Grid computing: making the global infrastructure a reality, 2003: p. 217-249.

31. Djemame, K., et al. Introducing risk management into the grid. in e-Science and Grid Computing, 2006. e-Science'06. Second IEEE International Conference on. 2006: IEEE.

32. Bellotti, T., et al., Chapter 6 - Feature Selection. Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications, ed. V. Balasubramanian, S.-S. Ho, and V. Vovk. 2014, Morgan Kaufmann, Boston: Newnes. 115-130.

33. Aha, D.W. and R.L. Bankert. Feature selection for case-based classification of cloud types: An empirical comparison. in AAAI-94 Workshop on Case-Based Reasoning. 1994: Seattle, WA.

34. Buyya, R., C.S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. in High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on. 2008: Ieee.

35. Syed, R.H., M. Syrame, and J. Bourgeois, Protecting grids from cross-domain attacks using security alert sharing mechanisms. Future Generation Computer Systems, 2013. 29(2): p. 536-547.

36. Sangrasi, A., K. Djemame, and I.A. Jokhio, Aggregating Node Level Risk Assessment in Grids Using an R-out-of-N Model, in Emerging Trends and Applications in Information Communication Technologies. 2012, Springer. p. 445-452.

37. Alsoghayer, R. and K. Djemame, Probabilistic risk assessment for resource provision in grids. Proceedings of the 25th UK PEW, 2009: p. 99-110.

38. Alsoghayer, R.A., Risk assessment models for resource failure in grid computing. 2011: University of Leeds.

39. Carlsson, C.F., Robert, Risk Assessment in Grid Computing, in Possibility for Decision. 2011, Springer. p. 145-165.

40. Carlsson, C. and R. Fullér, Risk Assessment of SLAs in Grid Computing with Predictive Probabilistic and Possibilistic Models, in Preferences and Decisions, S. Greco, et al., Editors. 2010, Springer Berlin Heidelberg. p. 11-29.

41. Arunraj, N., S. Mandal, and J. Maiti, Modeling uncertainty in risk assessment: An integrated approach with fuzzy set theory and Monte Carlo simulation. Accident Analysis & Prevention, 2013. 55: p. 242-255.

42. Haslum, K., A. Abraham, and S. Knapskog. Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment. in Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on. 2008: ieee.

43. Poolsappasit, N., R. Dewri, and I. Ray, Dynamic security risk management using bayesian attack graphs. Dependable and Secure Computing, IEEE Transactions on, 2012. 9(1): p. 61-74.

44. Haslum, K., A. Abraham, and S. Knapskog. Dips: A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment. in Information Assurance and Security, 2007. IAS 2007. Third International Symposium on. 2007: IEEE.

45. Haslum, K., A. Abraham, and S. Knapskog. Fuzzy online risk assessment for distributed intrusion prediction and prevention systems. in Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on. 2008: IEEE.

46. López, D., O. Villalba, and L.J. García. Dynamic risk assessment in information systems: state-of-the-art. in Proceedings of the 6th International Conference on Information Technology, Amman. 2013.

47. Parrilli, D.M., Legal Issues in Grid and cloud computing, in Grid and Cloud Computing. 2010, Springer. p. 97-118.

48. Bishop, M., What is computer security? Security & Privacy, IEEE, 2003. 1(1): p. 67-69.

49. Usop, N.S.M., A. Abdullah, and A.F.A. Abidin, Performance evaluation of AODV, DSDV & DSR routing protocol in grid environment. IJCSNS International Journal of Computer Science and Network Security, 2009. 9(7): p. 261-268.

50. Joseph, J., M. Ernest, and C. Fellenstein, Evolution of grid computing architecture and grid adoption models. IBM Systems Journal, 2004. 43(4): p. 624-645.

51. Das, S., Investigations into Performance Evaluation of Fabric level and Application level QoS Guarantee in Grid Environment. 2005, Dissertation Report, 71p, BITS Pilani, India.

52. Bote-Lorenzo, M.L., et al., A tailorable collaborative learning system that combines OGSA grid services and IMS-LD scripting, in Groupware: Design, Implementation, and Use. 2004, Springer. p. 305-321.

53. Baker, M., R. Buyya, and D. Laforenza, Grids and Grid technologies for wideâ€•area distributed computing. Software: Practice and Experience, 2002. 32(15): p. 1437-1466.

54. Johnston, W.E., D. Gannon, and B. Nitzberg, Information power Grid implementation plan: research, development, and testbeds for high performance, widely distributed, collaborative, computing and information systems supporting science and engineering. NASA Ames Research Center, http://www. nas. nasa. gov/IPG, 1999.

55. Woodward, P.R., et al., Cluster Computing in the SHMOD Framework on the NSF TeraGrid. 2004, LCSE internal report, April, 2004, available on the Web at www. lcse. umn. edu/turb2048.

56. Hoschek, W., et al., Data management in an international data grid project, in Grid Computingâ€”GRID 2000. 2000, Springer. p. 77-90.

57. Buyya, R., D. Abramson, and J. Giddy. A case for economy grid architecture for service oriented grid computing. in null. 2001: IEEE.

58. Casanova, H. and J. Dongarra, NetSolve: A network-enabled server for solving computational science problems. International Journal of High Performance Computing Applications, 1997. 11(3): p. 212-223.

59. Scardamalia, M. and C. Bereiter, Knowledge building. The Cambridge, 2006.

60. Schwiegelshohn, U., et al., Perspectives on grid computing. Future Generation Computer Systems, 2010. 26(8): p. 1104-1115.

61. Foster, I., C. Kesselman, and S. Tuecke, The anatomy of the grid. Berman et al.[2], 2003: p. 171-197.

62. Foster, I., et al. Cloud computing and grid computing 360-degree compared. in Grid Computing Environments Workshop, 2008. GCE'08. 2008: Ieee.

63. Foster, I. and C. Kesselman, The grid in a nutshell, in Grid resource management. 2004, Springer. p. 3-13.

64. Bhatia, R., Grid Computing and Security Issues. International Journal of Scientific and Research Publications, 2013: p. 554.

65. Foster, I., Globus toolkit version 4: Software for service-oriented systems. Journal of computer science and technology, 2006. 21(4): p. 513-520.

66. Foster, I., et al. A security architecture for computational grids. in Proceedings of the 5th ACM conference on Computer and communications security. 1998: ACM.

67. Wells, A.J., Grid application systems design. 2007: CRC Press.

68. Minoli, D., A networking approach to grid computing. 2004: John Wiley & Sons.

69. Nelson, E.K., et al., LabKey Server: an open source platform for scientific data integration, analysis and collaboration. BMC bioinformatics, 2011. 12(1): p. 71.

70. Tathe, M.V.A. and M.D.P. Patil, Next Generation Computing on the Internet (GRID). International Journal of Scientific and Research Publications, 2013. 2(2).

71. Bote-Lorenzo, M.L., Y.A. Dimitriadis, and E. GÃ³mez-SÃ¡nchez. Grid characteristics and uses: a grid definition. in Grid Computing. 2004: Springer.

72. Sharma, P., Grid Computing Vs. Cloud Computing. International Journal of Information and Computation Technology. ISSN, 2014: p. 0974-2239.

73. Iamnitchi, A. and I. Foster, A peer-to-peer approach to resource location in grid environments, in Grid Resource Management. 2004, Springer. p. 413-429.

74. Calheiros, R.N., et al., CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 2013. 41(1): p. 23-50.

75. Lee, C. and D. Talia, Grid programming models: Current tools, issues and directions. Grid Computing: Making the Global Infrastructure a Reality, 2003. 21: p. 555-578.

76. Kinhekar, A.M. and H. Gupta, A Review of Load Balancing in Grid Computing. 2014.

77. Foster, I., C. Kesselman, and S. Tuecke, The anatomy of the grid: Enabling scalable virtual organizations. International journal of high performance computing applications, 2001. 15(3): p. 200-222.

78. Misra, S.C. and A. Mondal, Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment. Mathematical and Computer Modelling, 2011. 53(3): p. 504-521.

79. Adabala, S., et al., From virtualized resources to virtual computing grids: the In-VIGO system. Future Generation Computer Systems, 2005. 21(6): p. 896-909.

80. Stankovski, V., et al., Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. Future Generation Computer Systems, 2008. 24(4): p. 259-279.

81. Weil, S.A., et al. Ceph: A scalable, high-performance distributed file system. in Proceedings of the 7th symposium on Operating systems design and implementation. 2006: USENIX Association.

82. Yang, J. and Z. Chen. Cloud computing research and security issues. in Computational intelligence and software engineering (CiSE), 2010 international conference on. 2012: IEEE.

83. Mollah, M.B., K.R. Islam, and S.S. Islam. Next generation of computing through cloud computing technology. in Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on. 2012: IEEE.

84. Talib, A.M., et al., Towards a comprehensive security framework of cloud data storage based on multi agent system architecture. Journal of Information Security, 2012. 3(04): p. 295.

85. Rana, O.F.W., Martijn Quillinan, Thomas B Brazier, Frances Cojocarasu, Dana, Managing violations in service level agreements, in Grid Middleware and Services. 2008, Springer. p. 349-358.

86. Kar, S.S., Bibhudatta, An Anamaly Detection System for DDOS attack in Grid Computing. International Journal of Computer Applications in, 2009.

87. Syed, R.H.S., Maxime Bourgeois, Julien, Protecting grids from cross-domain attacks using security alert sharing mechanisms. Future Generation Computer Systems, 2013. 29(2): p. 536-547.

88. Parrilli, D., Legal Issues in Grid and Cloud Computing, in Grid and Cloud Computing, K.W. Stanoevska-Slabeva, Thomas Ristol, Santi, Editor. 2010, Springer Berlin Heidelberg. p. 97-118.

89. Selvi, R.K. and V. Kavitha, Authentication in grid security infrastructure-survey. Procedia Engineering, 2012. 38: p. 4030-4036.

90. Boehm, B., Software risk management. 1989: Springer.

91. Fairley, R.E., Software risk management. IEEE Software, 2005(3): p. 101.

92. Boehm, B.W., Software risk management: principles and practices. Software, IEEE, 1991. 8(1): p. 32-41.

93. Keil, M., et al., A framework for identifying software project risks. Communications of the ACM, 1998. 41(11): p. 76-83.

94. Wallace, L. and M. Keil, Software project risks and their effect on outcomes. Communications of the ACM, 2004. 47(4): p. 68-73.

95. Catteddu, D., Cloud Computing: benefits, risks and recommendations for information security, in Web Application Security. 2010, Springer. p. 17-17.

96. Humphreys, E., Information security management standards: Compliance, governance and risk management. information security technical report, 2008. 13(4): p. 247-255.

97. Saleh, M.S. and A. Alfantookh, A new comprehensive framework for enterprise information security risk management. Applied computing and informatics, 2011. 9(2): p. 107-118.

98. Fernaeus, Y., J. Tholander, and M. Jonsson. Towards a new set of ideals: consequences of the practice turn in tangible interaction. in Proceedings of

the 2nd international conference on Tangible and embedded interaction. 2008: ACM.

99. Jrad, F., J. Tao, and A. Streit, SLA based Service Brokering in Intercloud Environments. CLOSER, 2012. 2012: p. 76-81.

100. Theilmann, W., R. Yahyapour, and J. Butler, Multi-level sla management for service-oriented infrastructures. 2008: Springer.

101. Abraham, A., C. Grosan, and V. Snasel. Programming Risk Assessment Models for Online Security Evaluation Systems. in UKSim 2009: 11th International Conference on Computer Modelling and Simulation. 2009: IEEE.

102. Haslum, K.A., Ajith Knapskog, Svein. Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment. in Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on. 2008: ieee.

103. Krautsevich, L.L., Aliaksandr Martinelli, Fabio Yautsiukhin, Artsiom. Risk-aware usage decision making in highly dynamic systems. in Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on. 2010: IEEE.

104. Feng, N. and M. Li, An information systems security risk assessment model under uncertain environment. Applied Soft Computing, 2011. 11(7): p. 4332-4340.

105. Marhavilas, P.K., D Gemeni, V, Risk analysis and assessment methodologies in the work sites: on a review, classification and comparative study of the scientific literature of the period 2000–2009. Journal of Loss Prevention in the Process Industries, 2011. 24(5): p. 477-523.

106. Sangrasi, A. and K. Djemame, Risk Assessment Modeling in Grids at Component Level: Considering Grid Resources as Repairable, in Distributed Computing and Artificial Intelligence, S. Omatu, et al., Editors. 2012, Springer Berlin Heidelberg. p. 321-330.

107. YADAV, J.S.J., MOHIT YADAV ANKIT, Risk Assessment Models And Methodologies. International Journal Of Scientific Research And Education, 2014. 1(06).

108. Wickboldt, J.A., et al., A framework for risk assessment based on analysis of historical information of workflow execution in IT systems. Computer Networks, 2011. 55(13): p. 2954-2975.

109. Pak, C. The near real time statistical asset priority driven (nrtsapd) risk assessment methodology. in Proceedings of the 9th ACM SIGITE conference on Information technology education. 2008: ACM.

110. Djemame, K., et al. Introducing risk management into the grid. in e-Science and Grid Computing, 2006. e-Science'06. Second IEEE International Conference on. 2006: IEEE.

111. Sangrasi, A. and K. Djemame. Component level risk assessment in grids: A probablistic risk model and experimentation. in Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on. 2011: IEEE.

112. Negoita, C., L. Zadeh, and H. Zimmermann, Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems, 1978. 1: p. 3-28.

113. Gourlay, I., K. Djemame, and J. Padgett. Reliability and risk in grid resource brokering. in Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on. 2008: IEEE.

114. Iosup, A., et al. On the dynamic resource availability in grids. in Proceedings of the 8th IEEE/ACM International Conference on Grid Computing. 2007: IEEE Computer Society.

115. Schroeder, B. and G. Gibson, A large-scale study of failures in high-performance computing systems. Dependable and Secure Computing, IEEE Transactions on, 2010. 7(4): p. 337-350.

116. Asnar, Y., et al. From trust to dependability through risk analysis. in Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. 2007: IEEE.

117. Zhang, Y., et al. Performance implications of failures in large-scale cluster scheduling. in Job Scheduling Strategies for Parallel Processing. 2005: Springer.

118. Gottumukkala, N.R., et al. Reliability analysis in HPC clusters. in Proc. of High Availability and Performance Computing Workshop. 2006.

119. Lingrand, D., et al., Optimization of jobs submission on the EGEE production grid: modeling faults using workload. Journal of Grid Computing, 2010. 8(2): p. 305-321.

120. Pinheiro, E., W.-D. Weber, and L.A. Barroso. Failure Trends in a Large Disk Drive Population. in FAST. 2007.

121. Jones, J., An introduction to factor analysis of information risk (fair). Norwich Journal of Information Assurance, 2006. 2(1): p. 67.

122. Alberts, C.J.D., Audrey, Managing information security risks: the OCTAVE approach. 2002: Addison-Wesley Longman Publishing Co., Inc.

123. Sangrasi, A.D., Karim, Risk Assessment Modeling in Grids at Component Level: Considering Grid Resources as Repairable, in Distributed Computing and Artificial Intelligence. 2012, Springer. p. 321-330.

124. Sangrasi, A.D., Karim Jokhio, Imran Ali, Aggregating Node Level Risk Assessment in Grids Using an R-out-of-N Model, in Emerging Trends and Applications in Information Communication Technologies. 2012, Springer. p. 445-452.

125. Carlsson, C. and R. Fullér, Probabilistic versus possibilistic risk assessment models for optimal service level agreements in grid computing. Information Systems and e-Business Management, 2013. 11(1): p. 13-28.

126. Alsoghayer, R. and K. Djemame, Resource failures risk assessment modelling in distributed environments. Journal of Systems and Software, 2014. 88: p. 42-53.

127. Carlsson, C.F., Robert, Risk Assessment of SLAs in Grid Computing with Predictive Probabilistic and Possibilistic Models, in Preferences and Decisions. 2010, Springer. p. 11-29.

128. Kira, K. and L.A. Rendell. A practical approach to feature selection. in Proceedings of the ninth international workshop on Machine learning. 1992.

129. Gutesman, E. and A. Waissbein. The impact of predicting attacker tools in security risk assessments. in Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research. 2010: ACM.

130. Butt, A.R., et al., Grid-computing portals and security issues. Journal of Parallel and Distributed Computing, 2003. 63(10): p. 1006-1014.

131. Chakrabarti, A., Taxonomy of grid security issues, in Grid Computing Security. 2007, Springer. p. 33-47.

132. Butt, A.R.A., Sumalatha Kapadia, Nirav H Figueiredo, Renato J Fortes, José AB, Grid-computing portals and security issues. Journal of Parallel and Distributed Computing, 2003. 63(10): p. 1006-1014.

133. Smith, M.F., Thomas Engel, Michael Freisleben, Bernd, Countering security threats in service-oriented on-demand grid computing using sandboxing and

trusted computing techniques. Journal of Parallel and Distributed Computing, 2006a. 66(9): p. 1189-1204.

134. Smith, M.E., Michael Friese, Thomas Freisleben, Bernd. Security issues in on-demand grid and cluster computing. in Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on. 2006b: IEEE.

135. Cody, E.S., Raj Rao, Raghav H Upadhyaya, Shambhu, Security in grid computing: A review and synthesis. Decision Support Systems, 2008. 44(4): p. 749-764.

136. Kussul, O.K., Nataliia Skakun, Sergii, Assessing security threat scenarios for utility-based reputation model in grids. Computers & Security, 2013.

137. Hassan, S.R.S., Maxime Bourgeois, Julien, Protecting grids from cross-domain attacks using security alert sharing mechanisms. Future Generation Computer Systems, 2012.

138. Lee, H.M.C., Kwang Sik Jin, Sung Ho Lee, Dae-Won Lee, Won Gyu Jung, Soon Young Yu, Heon Chang, A fault tolerance service for QoS in grid computing, in Computational Science—ICCS 2003. 2003, Springer. p. 286-296.

139. Oreski, S. and G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert systems with applications, 2014. 41(4): p. 2052-2064.

140. Hall, M., et al., The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 2009. 11(1): p. 10-18.

141. Wang, H.K., Taghi M Seliya, Naeem. How many software metrics should be selected for defect prediction? in FLAIRS Conference. 2011.

142. Bäck, T.S., Hans-Paul, An overview of evolutionary algorithms for parameter optimization. Evolutionary computation, 1993. 1(1): p. 1-23.

143. Tarvainen, M., Recognizing explosion sites with a self-organizing network for unsupervised learning. Physics of the Earth and Planetary Interiors, 1999. 113(1–4): p. 143-154.

144. Wu, C.-H.S., Wei-Han Ho, Ya-Wei, A study on GPS GDOP approximation using support-vector machines. Instrumentation and Measurement, IEEE Transactions on, 2011. 60(1): p. 137-145.

145. Désir, C.P., Caroline Heutte, Laurent Salaun, M Thiberville, Luc, Classification of endomicroscopic images of the lung based on random

subwindows and extra-trees. Biomedical Engineering, IEEE Transactions on, 2012. 59(9): p. 2677-2683.

146. Rokach, L., Ensemble methods in supervised learning, in Data mining and knowledge discovery handbook. 2010, Springer. p. 959-979.

147. Zhang, L.S., Ponnuthurai Nagaratnam, Random Forests with ensemble of feature spaces. Pattern Recognition, 2014.

148. Hossain, S.J. and N. Ahmad, Adaptive neuro-fuzzy inference system (ANFIS) based surface roughness prediction model for ball end milling operation. Journal of Mechanical Engineering Research, 2012. 4(3): p. 112-129.

149. Abraham, A., Adaptation of fuzzy inference system using neural learning, in Fuzzy systems engineering :Theory and Practice, N.N.e. al., Editor. 2005, Studies in Fuzziness and Soft Computing Springer: Verlag Germany. p. 53-83.

150. Abraham, A., Neuro fuzzy systems: State-of-the-art modeling techniques, in Connectionist models of neurons, learning processes, and artificial intelligence. 2001, Springer. p. 269-276.

151. Bey, K.B., et al. CPU load prediction model for distributed computing. in Parallel and Distributed Computing, 2009. ISPDC'09. Eighth International Symposium on. 2009: IEEE.

152. Karthika, B. and P.C. Deka, Prediction of Air Temperature by Hybridized Model (Wavelet-ANFIS) Using Wavelet Decomposed Data. Aquatic Procedia, 2015. 4: p. 1155-1161.

153. Kubat, M., Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. The Knowledge Engineering Review, 1999. 13(04): p. 409-412.

154. Golberg, D.E., Genetic algorithms in search, optimization, and machine learning. Addion wesley, 1989. 1989( ): p. 95-99.

155. Chen, Y., B. Yang, and J.A. Dong, Ajith, Time-series forecasting using flexible neural tree model. Information Sciences, 2005. 174(3–4): p. 219-235.

156. Mendes-Moreira, J., et al., Ensemble approaches for regression: A survey. ACM Computing Surveys (CSUR), 2012. 45(1): p. 10.

157. Vilalta, R., C. Giraud-Carrier, and P. Brazdil, Meta-Learning-Concepts and Techniques. Data Mining and Knowledge Discovery Handbook, 2010: p. 717-731.

158. Dietterich, T.G., Ensemble methods in machine learning, in Multiple classifier systems. 2000, Springer. p. 1-15.

159. Breiman, L., Bagging predictors. Machine Learning, 1996. 24(2): p. 123-140.

160. Hubbard, D.W., The failure of risk management: Why it's broken and how to fix it. 2009: John Wiley & Sons.

161. Modarres, M., Risk analysis in engineering: techniques, tools, and trends. 2006: CRC press.

162. Aven, T., Risk analysis. Assessing uncertainties beyond expected values and probabilities, 2008. 2008, Chichester, England: Wiley. x.

163. Bartlett, J., Project risk analysis and management guide. 2004: APM Publishing Limited.

164. Simon, P., D. Hillson, and K. Newland, PRAM: Project risk analysis and management guide. 1997: Association for Project Management High Wycombe.

165. Koller, G., Risk assessment and decision making in business and industry: A practical guide. 2005: CRC Press.

166. Bennett, J.C., et al., Risk analysis techniques and their application to software development. European Journal of Operational Research, 1996. 95(3): p. 467-475.

167. White, D., Application of systems thinking to risk management: a review of the literature. Management Decision, 1995. 33(10): p. 35-45.

168. Merna, T. and F.F. Al-Thani, Corporate risk management. 2011: John Wiley & Sons.

169. Douceur, J.R., The sybil attack, in Peer-to-peer Systems. 2002, Springer. p. 251-260.

170. Kar, S. and B. Sahoo, An Anamaly detection system for DDOS attack in grid computing. 2009.

171. Kussul, O., N. Kussul, and S. Skakun, Assessing security threat scenarios for utility-based reputation model in grids. Computers & Security, 2013. 34: p. 1-15.

172. Carlsson, C. and R. FullÃ©r, Risk Assessment of SLAs in Grid Computing with Predictive Probabilistic and Possibilistic Models, in Preferences and Decisions. 2011, Springer. p. 11-29.

173. Mukhin, V. and A. Volokyata, Integrated Safety Mechanisms Based on Security Risks Minimization for the Distributed Computer Systems. IJ Computer Network and Information Security, 2013. 2: p. 21-28.

174. Menasce, D.A. and E. Casalicchio, QoS in grid computing. Internet Computing, IEEE, 2004. 8(4): p. 85-87.

175. Lee, H.M., et al., A fault tolerance service for QoS in grid computing, in Computational Scienceâ€"ICCS 2003. 2003, Springer. p. 286-296.

176. Smith, M., et al., Secure on-demand grid computing. Future Generation Computer Systems, 2009. 25(3): p. 315-325.

177. Rana, O.F., et al., Managing violations in service level agreements, in Grid Middleware and Services. 2008, Springer. p. 349-358.

178. Lee, H.M., et al., A fault tolerance service for QoS in grid computing, in Computational Science—ICCS 2003. 2003, Springer. p. 286-296.

179. Kar, S. and B. Sahoo, An Anamaly detection system for DDOS attack in grid computing. International Journal of Computer Applications in Enggineering, Technology, and Sciences (IJ-CA-ETS), 2009. 1(2): p. 553-557.

180. Garner, S.R. Weka: The waikato environment for knowledge analysis. in Proceedings of the New Zealand computer science research students conference. 1995: Citeseer.

181. Barua, A., L.S. Mudunuri, and O. Kosheleva, Why trapezoidal and triangular membership functions work so well: Towards a theoretical explanation. Journal of Uncertain Systems2013. 8(3).

182. Kennedy, J., Particle swarm optimization, in Encyclopedia of Machine Learning. 2010, Springer. p. 760-766