Effective Cloud Solutions for Wildfire Management

by

Ujjwal KC

ME/MSc

ORCID: 0000-0002-5721-5534

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of ICT
College of Sciences and Engineering (CoSE)
UNIVERSITY OF TASMANIA

September 2021

Effective Cloud Solutions for Wildfire Management

Ujjwal KC
Supervisor: Dr. Saurabh Garg
Co-Supervisor: Dr. Jagannath Aryal, Dr. James Hilton

Abstract

Every year natural hazards such as wildfires cause massive destruction of physical infrastructure and loss of lives. A wide range of activities is carried out at different stages of a wildfire under wildfire management to minimize the associated risks. Representing the dynamics of wildfires with complex mathematical and empirical in the form of wildfire models is one of the effective ways to understand the behavior and form strategies against threatening wildfires. The current practice of wildfire management uses operational fire models such as Spark, Phoenix, FARSITE, and Prometheus, which are ideally expected to retrieve predictive information on the outspread of fires in as little time as possible. Risk metrics for a geographical location can be estimated by running multiple simulations, referred to as an *ensemble prediction*, with possible input factor conditions and conducting statistical analyses on simulation outputs. Such an approach of ensemble predictions can be analyzed in different ways to enable the estimation, analysis, and identification of the risks associated with wildfires. However, even a single simulation in an ensemble is a complex calculation based on interrelationships between different parameters and must also deal with geographical information data sets. Consequently, running ensembles on a single computer or a small cluster can result in bottlenecks due to data access and processing constraints and take longer than the window available for preparation for any imminent disaster.

Research carried out in recent years has put forward Cloud Computing frameworks as a possible solution to increase the efficiency of the ensemble results from the prediction tools and make these services available to many users in a scalable way. Cloud infrastructure itself does not decrease the computation time for individual simulation in an ensemble. But it provides a means to reduce the overall time of the ensemble as it allows elastic on-demand access to almost unlimited storage, network, and computational processing. However, this access to the Cloud resources must be coupled with an effective control mechanism and innovative solutions in the system design to manage the resources and support the ensemble predictions in optimal manners to rapidly estimate, identify, and analyze the associated risks.

As such, intending to enable ensemble predictions for rapid risk estimation, analysis, and identification, this thesis first presents the existing challenges through a comprehensive review of the adaptation of Cloud Computing in disaster modeling and management systems. To enable rapid ensemble wildfire predictions over Clouds for rapid risk estimation against the associated computational challenges, the thesis proposes a Cloud-based framework that offers ensembles of wildfire simulations as a service in a cost and time-efficient manner. As an improvement, the thesis extends the framework to facilitate running the fire simulations with sampled values of input parameters, referred to as *sensitivity analysis*, to perform a rapid risk analysis and determine the conditions with significant threats, which can be prohibitively time-consuming in local machines. Finally, against the naive comprehensive sweep methods in conventional ensemble predictions where simulations are run at all start locations to identify high-risk areas, the thesis proposes a novel quadtree-based search mechanism that can rapidly identify potential high fire-risk areas and produces an increasingly detailed risk map within a given time frame without running simulations at all start locations.

The wildfire model analyses carried out with real use cases in the Tasmanian region verify the efficacy and usability of the proposed solutions in a real operational environment. The solutions proposed in this thesis are model-agnostic and can be easily transferred to other natural hazard models.

This thesis adds to the body of the knowledge by making the following contributions:

1. A comprehensive survey that reflects the current research trends in utilizing ICT infrastructures including Cloud Computing to support different aspects of natural disaster management.
2. A validated foundation system design (framework) to deploy the ensemble of wildfire simulations as end services over the Clouds considering the user requirements with minimal cost for rapid risk estimation.
3. A brief report on parametric uncertainty quantification in Australian fire spread models used in Australian Fire Danger Rating System (AFDRS).
4. A comprehensive sensitivity analysis of input parameters in the widely used fire spread models with an insight into the implications of results on the understanding and interpretation of the fire models.
5. A cloud-based framework that can efficiently handle the high computational need of sensitivity analysis of operational disaster models for rapid risk analysis.
6. A novel and innovate quadtree-based mechanism for rapidly identifying areas of wildfire risk in operational management.

Declaration of Originality

I, Ujjwal KC, declare that this thesis titled, 'Effective Cloud Solutions for Wildfire Management' and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the PhD except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used, and
- the thesis is less than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices, as approved by the Research Higher Degrees Committee.

Signed: _____

Date: 29 April 2021

Authority of Access

This thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Signed: _____

Date: 29 April 2021

Statement of Co-Authorship

The following people and institutions contributed to the publication of work undertaken as part of this thesis:

Candidate –Ujjwal KC, School of Information and Communication Technology (ICT), University of Tasmania

Co-Author 1 (Supervisor) –Saurabh Garg, School of Information and Communication Technology, University of Tasmania

Co-Author 2 (Co-Supervisor) –Jagannath Aryal, School of Information and Communication Technology, University of Tasmania

Co-Author 3 (Industrial Supervisor) –James Hilton, Data61, CSIRO, Melbourne, Australia

Co-Author 4 –Andrew Sullivan, CSIRO, Canberra, Australia

Co-Author 5 –Matt Plucinski, CSIRO, Canberra, Australia

Co-Author 6 –Nicholas Forbes-Smith, CSIRO, Melbourne, Australia

Contribution of work by co-authors for each paper:

PAPER 1: KC, U., Garg, S., Hilton, J., Aryal, J., & Forbes-Smith, N. (2019). Cloud Computing in natural hazard modeling systems: Current research trends and future directions. International Journal of Disaster Risk Reduction, 38, 101188. (Located in Chapter 2)

Author contributions: Conceptualization, U KC, S Garg and J Hilton; Formal analysis, U KC; Investigation, U KC; Methodology, U KC and S Garg; Supervision, S Garg, J Aryal and J Hilton; Validation, U KC, S Garg, J Aryal and J Hilton; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, S Garg, J Aryal, J Hilton and N Forbes-Smith.

U KC - 75%, S Garg - 10%, J Hilton - 7%, J Aryal - 5% and N Forbes-Smith - 3%

PAPER 2: KC, U., Garg, S., & Hilton, J. (2020). An efficient framework for ensemble of natural disaster simulations as a service. Geoscience Frontiers, 11(5), 1859-1873. (Located in Chapter 3)

Author contributions: Conceptualization, U KC and S Garg; Formal analysis, U KC; Investigation, U KC and S Garg; Methodology, U KC and S Garg; Software, J Hilton; Supervision, J Hilton and S Garg; Validation, U KC, S Garg and J Hilton; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, S Garg and

J Hilton.

U KC - 80%, S Garg - 15%, J Hilton - 5%

PAPER 3: KC, U., Sullivan A., Hilton, J., Plucinski M., Garg, S., & Aryal, J. (2021). Assessing the sensitivity of Australian operational wildfire spread models. International Journal of Wildland Fire, (Under Review). (Located in Chapter 4)
Author contributions: Conceptualization, U KC and J Hilton; Formal analysis, U KC; Investigation, U KC; Methodology, U KC, J Hilton, A Sullivan and M Plucinski; Supervision, S Garg, J Hilton and J Aryal; Validation, U KC, J Hilton, A Sullivan and M Plucinski; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, J Hilton, A Sullivan, M Plucinski, S Garg and J Aryal.

U KC - 70%, A Sullivan - 8%, J Hilton - 7 %, M Plucinski - 7%, S Garg - 5%, J Aryal - 3%

PAPER 4: KC, U., Aryal, J., Garg, S., & Hilton, J. (2021). Global sensitivity analysis for uncertainty quantification in fire spread models. Environmental Modelling & Software, 143, 105110. (Located in Chapter 4)
Author contributions: Conceptualization, U KC and J Aryal; Formal analysis, U KC; Investigation, U KC; Methodology, U KC; Supervision, J Aryal, S Garg and J Hilton; Validation, U KC, and J Hilton; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, J Aryal, S Garg and J Hilton.

U KC - 75%, J Aryal - 10%, S Garg - 8%, J Hilton - 7%

PAPER 5: KC, U., Garg, S., Hilton, J., & Aryal, J. (2020). A cloud-based framework for sensitivity analysis of natural hazard models. Environmental Modelling & Software, 134, 104800. (Located in Chapter 4)
Author contributions: Conceptualization, U KC and S Garg; Formal analysis, U KC; Investigation, U KC; Methodology, U KC and S Garg; Software, J Hilton; Supervision, S Garg, J Hilton and J Aryal; Validation, U KC and J Hilton; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, S Garg, J Hilton and J Aryal.

U KC - 75%, S Garg - 10%, J Hilton - 10%, J Aryal - 5%

PAPER 6: KC, U., Garg, S., Hilton, J., & Aryal, J. An adaptive quadtree-based approach for rapidly determining areas of wildfire risk. Nature Sustainability (Submission Draft). (Located in Chapter 5)
Author contributions: Conceptualization, U KC and J Hilton; Formal analysis, U KC; Investigation, U KC, J Hilton and S Garg; Methodology, U KC, J Hilton, S Garg and J Aryal; Software, J Hilton; Supervision, S Garg, J Hilton and J Aryal; Validation, U KC, J Hilton, S Garg and J Aryal; Visualization, U KC; Writing – original draft, U KC; Writing – review & editing, U KC, S Garg, J Hilton and J Aryal.

U KC - 75%, S Garg - 10%, J Hilton - 10%, J Aryal - 5%

We, the undersigned, endorse the above stated contribution of work undertaken for each of the published (or submitted) peer-reviewed manuscripts contributing to this thesis.

Signed:

Dr. Saurabh Garg School of ICT, University of Tasmania Date: 27 April 2021	Associate Professor Quan Bai School of ICT, University of Tasmania Date: 28 April 2021

Ujjwal KC
Candidate, School of ICT,
University of Tasmania

Date: 27 April 2021

Statement of Published Works

The publishers of the papers comprising Chapters 2 to 4 hold the copyright for that content and access to the material should be sought from the respective journals/publishers. The remaining non-published content of the thesis may be made available for loan and limited copying and communication in accordance with the Statement of Access and the Copyright Act 1968.

Due to the inclusion of published material there is unavoidable repetition of material between chapters in this thesis.

Signed: _____

Date: 30 April 2021

Preface

Main contributions:

The research work for this thesis was carried out in the Big Systems Lab, School of Technology, Environment and Design, University of Tasmania. The main contributions presented in this thesis in Chapters 2-5 are based on the following publications:

- ★ **KC, U.**, Garg, S., Hilton, J., Aryal, J., & Forbes-Smith, N. (2019). Cloud Computing in natural hazard modeling systems: Current research trends and future directions. International Journal of Disaster Risk Reduction, 38, 101188.
- ★ **KC, U.**, Garg, S., & Hilton, J. (2020). An efficient framework for ensemble of natural disaster simulations as a service. Geoscience Frontiers, 11(5), 1859-1873.
- ★ **KC, U.**, Sullivan A., Hilton, J., Plucinski M., Garg, S., & Aryal, J. (2021). Assessing the sensitivity of Australian operational wildfire spread models. International Journal of Wildland Fire, (Under Review).
- ★ **KC, U.**, Aryal, J., Garg, S., & Hilton, J. (2021). Global sensitivity analysis for uncertainty quantification in fire spread models. Environmental Modelling & Software, 143, 105110.
- ★ **KC, U.**, Garg, S., Hilton, J., & Aryal, J. (2020). A cloud-based framework for sensitivity analysis of natural hazard models. Environmental Modelling & Software, 134, 104800.
- ★ **KC, U.**, Garg, S., Hilton, J., & Aryal, J. An adaptive quadtree-based approach for rapidly determining areas of wildfire risk. Nature Sustainability (Submission Draft).

Acknowledgements

Words cannot express how grateful I feel to my supervisors (Dr. Saurabh Garg (primary), Dr. Jagannath Aryal, and Dr. James Hilton) for their generous guidance and supervision.

Additionally, I would like to acknowledge everyone who inspired and helped me at various stages of my PhD studies.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

According to the records of Emergency Events Database, natural hazards have cost around 3 trillion dollars and caused 1.3 million casualties between 1998 and 2017 [2]. There is a wide range of activities that can be specifically directed and carried out at different stages of a natural disaster but effective management of these activities, commonly referred to as *Natural Disaster Management (NDM)*, is required to ensure the least damages are inflicted by the disaster. Various time windows can be categorized for NDM. Several activities can be carried as a pre-planning step to mitigate the dangerous impacts of a potential hazard. Such preparedness before the occurrence of a disaster as well as rapid damage assessment after a disaster can be hugely important in minimizing the disaster risks.

Natural hazards such as wildfires are phenomena that can be described by models in terms of dynamical relationships between their driving factors using mathematical and empirical relationships. Natural hazard modeling systems use such models to estimate the outspread of natural phenomena and pinpoint the activities that can be carried out at different stages to minimize the associated impacts. A wide range of models has been constructed for predicting natural hazards and effective disaster management. These include wildfire propagation models (Spark [3], Phoenix [4], FARSITE [5], Prometheus [6]), flood spread models (Swift [7], Rapid Flood Spreading Model (RFSM) [8]), dust storm forecasting model [9], landslide prediction model (Landslide Hazard Assessment

for Situational Awareness (LHASA) model [10]), cyclone models (Hurricane Weather Research and Forecasting (HWRF) model [11], Beta and advection (BAMM) model [12]), earthquake models (Kanai –Tajimi model [13], Dilatancy-diffusion model [14]) and many others. These models are necessarily complex as many environmental factors must be taken into account. For example, wildfire models require several input parameters such as the fuel condition, local weather, the type of land coverage, and local topography [15, 16].

In the current natural hazard modeling and management systems, risk metrics are derived from hazard models by running multiple simulations, referred to as an *ensemble prediction*. The usual practice of ensemble predictions for *risk estimation* includes running the disaster simulations at all possible input conditions and conducting statistical analyses to estimate the risks associated with the disaster. With the increasing complexity of hazard models, every simulation in the ensemble is computationally intensive itself as it includes a complex calculation and must also deal with geographical information data sets. Consequently, running ensembles on a single computer or a small cluster can result in bottlenecks due to data access and processing constraints and can take longer than the window available for preparation against any imminent disaster. Additionally, each of the input parameters in disaster simulations is subject to uncertainties that affect the outcome of the model. Running the simulations with sampled values of input parameters, referred to as *sensitivity analysis*, helps quantify the associated uncertainties and perform a *risk analysis* to determine the conditions with significant threats. But, such sensitivity analyses necessitate a large number of disaster simulations to be run before deriving any analytical results on the disaster thereby making such analyses prohibitively time-consuming on conventional local systems. Additionally, the identification of high-risk areas for a geographical location *(risk identification)* in an ensemble prediction is achieved by running the simulations in a sequential manner that covers all the possible locations. However, it has been observed that only a small fraction of the possible fires over a geographical area will be high-risk fires. Any solutions that could identify these high-risk areas with as few simulations as possible can help retrieve risk information and identify the most vulnerable regions for further operational measures. But, such an approach is not used within current operational fire management systems and tools, despite the potential benefits.

Researchers have turned their attention to *Cloud Computing* to address the computation-related complexities of various applications. Cloud Computing is a new computing paradigm that exploits the principle of distributed computing in multiple virtual machines. The National Institute of Standards and Technology (NIST) has defined Cloud Computing as "*a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction* [17]". The introduction of Cloud Computing has revolutionized the way computation is carried out in organizations and research. Computation is now considered as a utility service, rather than the traditional model of owning and utilizing resources for different applications. This shift of computing paradigm facilitates the users to focus more on their application and spend lesser time on repairing and maintaining the resources. Cloud Computing provides an almost unlimited capacity for computation, storage, and networking through its vast chain of virtualized resources ensuring key features of on-demand service. These include ubiquitous network access, independent resource pooling, rapid elasticity, and a service-based approach. With all these features, Cloud Computing seems to be an attractive solution to the challenges associated with the computational complexities of ensemble predictions and their sensitivity analyses. However, the sharing of intermediate data sets between different simulations and the specific geoscience requirements of the models make the ensemble predictions different from the well-studied application, such as bag-of-tasks, on Cloud infrastructure. Consequently, accommodating natural hazard models and simulations in a Cloud environment for risk estimation, identification, and analysis requires in-depth exploration of innovative and novel solutions that are yet to be realized.

1.2 Research Gap

In this section, to enable ensemble predictions efficiently over Cloud infrastructure, we present the existing research gaps in current natural hazard modeling and management systems that have prevented rapid risk estimation, analysis, and identification for operational management.

1.2.1 Rapid Risk Estimation with Ensemble Predictions

In an ensemble prediction, risk metrics are estimated from hazard models by running multiple simulations for possible input factors' combinations, which may take longer than the time window available during an unfolding disaster. Yang et al. [18] conducted an extensive review of research conducted to address geoscience and Digital Earth needs for an integrated Earth system using Cloud Computing and demonstrated that Cloud Computing offers unprecedented new capabilities for such systems. Garg et al. [19] proposed a scalable framework over Cloud infrastructure to increase the efficiency of the bushfire prediction process by making the service available to several users within deadlines. The authors improvised by proposing another Cloud-based framework, *sparkCloud* [20], to demonstrate the ability of Cloud Computing to offer elastic and scalable Cloud solution for wildfire prediction model based on multiple user requests and deadline requirements. Additionally, Huang et al. [21] verified the capability of Cloud Computing to support ensemble simulations by deploying a complex dust forecasting model on an Amazon EC2 foundation with reduced cost when compared to using local resources. Li et al. [22] described a Model as a Service (MaaS) framework to support ensemble simulations of different Geoscience models over the Cloud infrastructure. Moreover, a cyberinfrastructure-based system developed by Behzad et al. [23] detailed the implementation of ensemble simulation of groundwater system modeling over the Cloud environment provided by Microsoft Windows Azure Cloud Platform.

These works have validated the readiness of Cloud infrastructure to support the complex ensemble simulations of different Geoscience models including wildfire models. Cloud infrastructure itself does not decrease the computation time for individual simulation in an ensemble. But, it provides a means to reduce the overall time of the ensemble as it allows elastic on-demand access to almost unlimited storage, network, and computational processing. However, this access to the Cloud resources must be coupled with an effective control mechanism and innovative solutions in the system design to manage the resources and support the ensemble predictions in optimal manners for rapid risk estimation. However, fewer developments have been made to offer these models as end services to the users. there are not any well-defined mechanisms to initiate and automate the multiple runs of simulations with minimal user interventions (a single user request) for an ensemble of disaster simulations. Moreover, cost and resource optimization for ensemble

simulations of wildfire models over the Cloud environment has not, to our knowledge, been previously considered and continues to be an open challenge.

1.2.2 Rapid Risk Analysis using Sensitivity Analysis

Risk analysis of any natural disaster helps identify the input factors' combination that pose significant threats. Natural hazard models such as wildfires are required both for risk analysis to identify vulnerable regions and assets and faster-than-real-time operational applications during an unfolding disaster. Risk analysis helps identify the input factors' combination that pose significant threats. These models are necessarily complex as many environmental factors must be taken into account. For example, wildfire models require input parameters such as the fuel condition, local weather, the type of land coverage, and local topography [15, 16]. Each of these input parameters is subject to uncertainties that affect the outcome of the model, such as fire area or maximum heat intensity [24]. To take these uncertainties into account, multiple simulations with input values sampled from these uncertain inputs can be used. However, the interrelation between the parameters and outputs from these models is complex and usually non-linear [25]. As such, running multiple simulations with different input combinations in an ensemble prediction, for risk assessment requires information on the sensitivity of the outcome of the model to various inputs.

Sensitivity analysis (SA) is one means to determine the influence of input parameters on a model outcome and its uncertainties [15, 26, 27]. In local SA, the impact of the parameters is studied around a specific point while in global SA, the entire range of the input parameters is considered [26]. Global SA methods allow the dominant factors driving the model to be identified and quantifies the relationship between uncertainties in input parameters, helping to understand the model. GSA has recently gained attention in environmental modeling in areas such as wildfire, hydrology, decomposition, and crops [27–30]. GSA helps to identify influential and non-influential factors in the model and fixing the non-influential factors to a known value, and treatment of uncertainties that contribute to better understanding and interpretation of the model [25, 26]. Operational disaster models such as wildfire models are computational models characterized by different, often complex, mathematical relationships that must be calculated multiple times for each combination of input parameters to produce a set

of outputs. As natural hazard models often require a large number of input parameters, accurate sensitivity analyses require a large number of combinations, making such analyses compute-intensive and time-consuming. These analyses can take several hours to days to complete for complex models. Such analyses also practically require a high degree of maintenance for data handling, orchestration, and management of results for the calculation of the final required metrics. The ability to automate SA and reduce the time taken for such analysis could benefit operational disaster management by rapidly determining the dominant factors affecting a particular local natural hazard to guide efficient response and planning. However, to the authors' knowledge, there are no systems or services that offer such analyses in a scalable, time-efficient, and convenient manner.

1.2.3 Rapid Identification of Areas of High Risks

Under the current state-of-the-art wildfire management systems, several methods have been in use to identify the areas with high fire risks. These methods include the use of satellite thermal images to detect fire hotspots (locations with active fires), fire-danger/severity rating calculation, and wildfire modeling. Satellite images have been primarily used to detect the fire hotspots and estimate the burned areas [31]. As reported in the study, sensors such as AVHRR [32], ATSR [33], MODIS [34], and MSG [35] have been used for various fire hotspot and burned area related applications. Lately, satellite images along with aerial images have been used in mapping fire severity for a region of interest [36, 37]. Using satellite images for identifying high fire-risk areas is possible only during the events of fires when the satellites are over-passing those areas. Moreover, the identified hotspots via satellite can be inferred to be cumulative of total fire occurrences around a region. Analyzing satellite or aerial images, on the other hand, can be computationally complex which can take a longer time on a limited pool of computing resources [38, 39]. As such, remote sensing techniques with satellites may not be one of the most effective methods to predict or identify high fire-risk areas before or during fire emergencies, especially when the satellite is not over-passing the area of interest. Nevertheless, such techniques are more suited for other applications such as burned area estimation [40], gas emission estimation [41] and analysis of fire regimes [42]. Fire danger rating calculation based on meteorological data has also been in practice to identify the areas with high fire risks. Canadian Fire Weather Index System (CFWIS) [43], US National Fire Danger Rating System (NFDRS) [44], Russian

Nesterov Index [45], the Italian RISICO (RISchio Incendi e Coordinamento) Index [46] all use weather data from weather station or weather forecast model to assess the risk of possible fires for any region of interest for any given day in a year. However, such fire danger ratings should be used as an approximate guide to expected fire behavior. The use of fire ratings at extremities should be handled with care as they are primarily derived using meteorological data [47].

Wildfire risk modeling has also been used to identify high fire-risk locations by predicting the fire spread rate or estimating various risk metrics in an operational framework. Such fire models have been integrated with landscape fire planning, fire suppression, and operational incidental fire management to provide more information to fire responders during emergencies [48–50]. Consequently, wildfire models have been one of the key decision-making tools for fire risk management during various stages of fire emergencies. While identifying the high fire risk areas using the risk metrics obtained from the wildfire models, under a conventional comprehensive sweep method, the models have to be run at all possible start locations. Such methods may not be scalable for a larger geographical location and may delay the identification of high risk areas beyond the available time window for preparation against an imminent disaster.

1.3 Problem Statement and Objectives

This thesis is centered around the following research problem.

Q. How to enable ensemble predictions efficiently over Cloud infrastructure for rapid risk estimation, analysis, and identification in current natural hazard modeling systems?

1.3.1 Research Questions

Based on the research problem, we derive the following research questions.

1. How has Cloud Computing been used in current natural hazard modeling systems? What are the conventional methods for disaster management?

2. How to enable ensemble predictions over Cloud infrastructure as a service in a resource and time-efficient manner?

3. How can Cloud solutions support the (sensitivity) analyses of simulations in ensemble predictions to identify the scenarios with significant risks?

4. Can search strategies in ensemble predictions facilitate rapid identification of high-risk areas?

1.3.2 Research Objectives

This thesis solves the research problem and answers the research questions by achieving the following objectives.

1. To reflect the picture of the current state-of-the-art of Cloud solutions in natural hazard modeling systems through a comprehensive survey.

2. To build an efficient Cloud-based framework for ensembles of natural disaster simulations in a convenient and resource-efficient manner.

3. To propose a cloud-based framework for sensitivity analysis of inputs to operational disaster models.

4. To devise novel and innovative mechanisms in ensemble predictions to rapidly identify the disaster risks.

1.4 Methodology

To achieve rapid risk estimation, analysis, and identification through efficient ensemble predictions over Cloud, we divide our work into four different phases. Phase 1 reflects the picture of current state of the art Cloud solutions and highlights the challenges for efficient ensemble prediction process, while Phase 2 proposes a cloud-based framework for efficient ensemble prediction for rapid risk estimation. Phase 3 enables rapid risk analysis by proposing a cloud-based framework for sensitivity analysis of wildfire simulations while Phase 4 achieves rapid risk identification by incorporating quadtree-based search

strategy in conventional ensemble prediction. The four phases are further described as follows.

Phase 1: Survey of existing cloud-based natural hazard modeling system

This phase provides background information on disaster management and discusses the general challenges in developing effective cloud-based systems for disaster models. In this phase, we continue to depict the current state of the art of the cloud-based natural hazard modeling system by providing a comprehensive survey of different works under well-defined categories. Furthermore, we highlight future directions where the current research can be focused on to realize more efficient solutions for disaster management.

Phase 2: An efficient framework for ensembles of disaster simulations as a service

To address the challenges of computing, data, concurrent-access intensiveness, and time requirements associated with ensemble predictions, we propose a system with two phases of optimization. In the first phase, the possible incurred cost of running ensemble predictions is minimized through the optimal distribution of the simulations among the cost-efficient workers while still complying with the user requirements. The second phase tries to further minimize the cost of operation by intelligently choosing the instances based on different pricing models - on-demand, reserved, and spot. The work validates the proposed system design using a real use case of the ensemble simulations using wildfire prediction tool 'Spark' as an end service under different scenarios of user requirements of time and request complexities over the real Cloud infrastructure.

Phase 3: A cloud-based framework for sensitivity analysis of operational disaster models

In this phase, based on the findings from the application of sensitivity analysis methods to empirical wildfire models, we propose a cloud-based framework to conduct the sensitivity analysis of operational disaster models as a service in a convenient manner for rapid risk analysis. For the demonstration of the applicability of the sensitivity analysis (SA) methods for risk analysis and comparison between different SA methods, the parameters and their ranges are chosen as such to cover all the operational conditions of a considered region and make fair comparisons between the models. The proposed framework considers different user inputs and an existing SA method to calculate the

sensitivity indices conveniently once the user submits the request to determine the conditions with significant threats.

Phase 4: An adaptive quadtree-based approach for rapidly identifying areas of wildfire risk

In this phase, we investigate the adaptation of search strategies in conventional sweep methods in ensemble predictions to identify the disaster risks for a given geographical region. We propose a novel quadtree-based mechanism that adaptively identifies potential high fire-risk areas and produces an increasingly detailed risk map within a given time frame.

1.5 Research Contributions

The research contributions of this thesis are listed as follows.

1. A comprehensive survey that reflects the current research trends in utilizing ICT infrastructures including Cloud Computing to support different aspects of natural disaster management.
2. A validated foundation system design (framework) to deploy the ensemble of wildfire simulations as end services over the Cloud considering the user requirements with minimal cost for rapid risk estimation.
3. A brief report on parametric uncertainty quantification in Australian fire spread models used in Australian Fire Danger Rating System (AFDRS).
4. A comprehensive sensitivity analysis of input parameters in the widely used fire spread models with an insight into the implications of results on the understanding and interpretation of the fire models.
5. A cloud-based framework that can efficiently handle the high computational need of sensitivity analysis of operational disaster models for rapid risk analysis.
6. A novel and innovate quadtree-based mechanism for rapidly identifying areas of wildfire risk in operational management

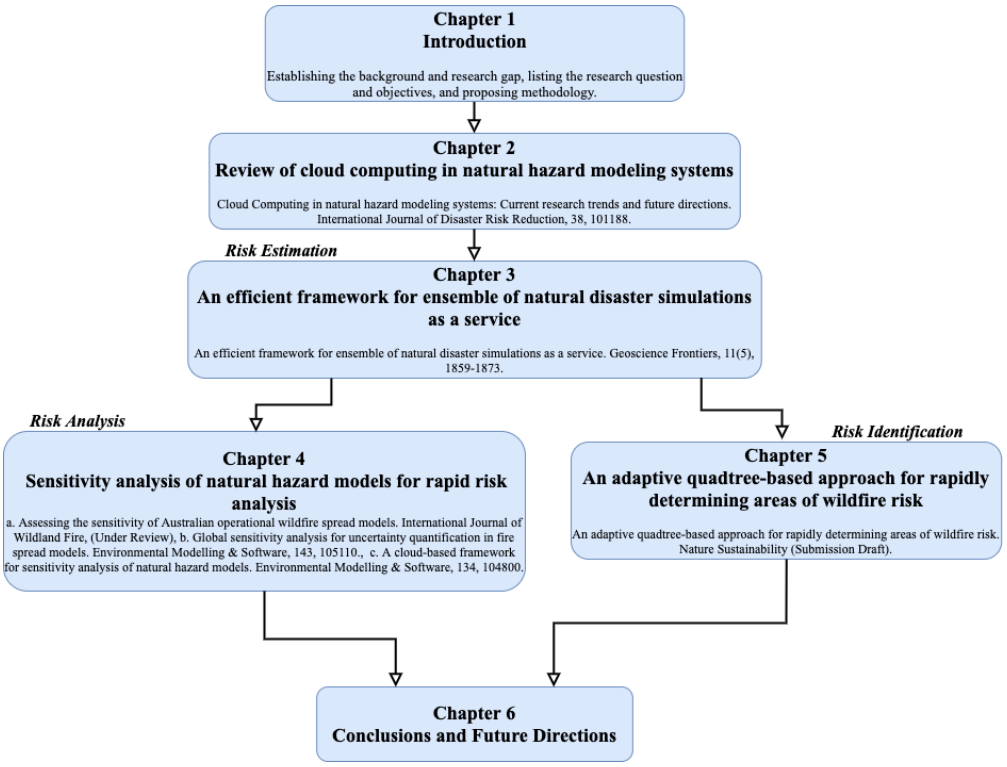


Figure 1.1: Thesis Organization

1.6 Thesis Outline

The thesis organization is outlined in Figure 1.1. Additionally, Figure 1.1 shows how each chapter was derived from research articles produced. Chapter 2 reflects the current state-of-the-art of Cloud Computing in natural hazard modeling systems while Chapter 3 describes the foundation system that enables the ensemble of disaster simulations over the cloud environment as a service for rapid risk estimation. Chapter 4 first describes how sensitivity analysis can be used for risk analysis and then explains a cloud-based framework for sensitivity analysis of operational disaster model for rapid risk analysis. Chapter 5 details an adaptive quadtree-based search mechanism in conventional ensemble predictions for rapid identification of the areas of wildfire risks while Chapter 6 concludes the thesis with future works.

Chapter 2

Literature Review

To better understand the challenges and knowledge gaps that have prevented rapid risk estimation, analysis, and identification in current ensemble prediction systems, this chapter conducts a broad and comprehensive review of technical solutions used in disaster management systems. This chapter aggregates all the challenges, reflects on the current research trends, and outlines a conceptual Cloud-based solution framework for more effective natural hazards modeling and management systems using Cloud infrastructure in conjunction with other technologies such as Internet of Things (IoT) networks, fog, and edge computing. Additionally, it draws a clear picture of the current research state in the area and suggests further research directions for future systems for efficient disaster management.

This chapter is derived from the following published work.

KC, U., Garg, S., Hilton, J., Aryal, J., & Forbes-Smith, N. (2019). Cloud Computing in natural hazard modeling systems: Current research trends and future directions. International Journal of Disaster Risk Reduction, 38, 101188.

2.1 Introduction

According to the record of Emergency Events Database, natural hazards have cost about 3 trillion dollars of economic destruction and 1.3 million casualties with more than 4.4 billion people injured between 1998 and 2017 [2]. Despite the development of various

technology aided systems to understand and mitigate the effects of natural hazards, effective disaster prediction and management continues to be a worldwide issue. Various time windows can be categorized for Natural Hazard Management. A wide range of activities can be carried as a pre-planning step to mitigate the dangerous impacts of a potential hazard. Such preparedness before the occurrence of a disaster, as well as rapid damage assessment after a disaster, can be hugely important in ensuring the least damage is inflicted in terms of lives and infrastructure. Activities carried out during a disaster, such as faster and real-time modeling, allow effective operational strategies to be developed and implemented to decrease the impacts of the disasters.

A wide range of models have been constructed for predicting natural hazards and effective disaster management. These include wildfire propagation models (Spark [3], Phoenix [4], FARSITE [5], Prometheus [6]), flood spread models (Swift [7], Rapid Flood Spreading Model (RFSM) [8]), dust storm forecasting model [9], landslide prediction model (Landslide Hazard Assessment for Situational Awareness (LHASA) model [10]), cyclone models (Hurricane Weather Research and Forecasting (HWRF) model [11], Beta and advection (BAMM) model [12]), earthquake models (Kanai-Tajimi model [13], Dilatancy-diffusion model [14]) and many others. On the other hand, many studies have investigated and integrated various aspects of ICT in Geospatial Science and Disaster Management so as to work efficiently for the prevention and management of natural hazards. Satellite Remote Sensing, along with various monitoring and alerting tools, had been effectively used to study and manage the natural disasters. The recent advancements in technological aspects have made Geospatial Science face multiple challenges related to computation, storage and network. Geospatial Science collects, stores, analyzes, processes and simulates data from different regions of the world. The workload and scope of this have exponentially increased with the development of new sensors, the sophisticated information collecting methods and further understanding of Geospatial processes. This proceeding has made Geospatial applications and services data-intensive, compute-intensive and concurrent access-intensive. Hugely massive data sets collected from large regions in multi-temporal and spectral dimension, by using high-end resolution sophisticated sensors, have contributed to a huge bottleneck of data in Geospatial Sciences [51]. The algorithms and models developed in Geospatial Sciences are becoming more complex with an improved understanding of spatio-temporal principles driving those phenomena [52]. These models may require ensembles of simulations

for better disaster risk metrics, which is computationally intensive to implement. The recent rise in popularity of web and wireless devices has made it possible for numerous end users to access the services concurrently. These models, when offered as end services, invite various challenges of having to keep up with as fast as possible access and respond to sudden change in the number of concurrent users [53].

The implementation of Geospatial and natural hazard models over the traditional ICT foundation has become non-trivial and the researchers have turned their attention to Cloud Computing. Evolved from the principles of distributed computing, Cloud Computing possesses the ability of pooling, sharing, integrating the latest computing technologies and physically distributed computer resources [54]. Cloud Computing provides an on-demand and elastic access to an almost unlimited storage, network and computational resources. These features directly address the challenges of data, compute and concurrent-access intensiveness in the implementation of Geospatial models for disaster management. The adaptation of Cloud Computing in Geospatial Science for Natural Disaster Management(NDM) is one of the least explored areas despite the fact that Cloud Computing has a tremendous potential to revolutionize the disaster management with its neat shared architecture of infinite storage and computing resources.

There are a few research areas where Cloud Computing has been used in Geospatial applications for NDM to enhance the performance of the system with reduced cost and complexities. The exemplars presented by Yang et al. [18] provide a brief insight into how Cloud capabilities were used to support specific requirements of different applications. The work done so far has been able to initiate and verify the suitability of the use of Cloud Computing in Geospatial Science for NDM. The ability to offer the functionalities of NDM as end services is attractive to researchers and is now relatively easier to achieve. However, a neat and effective approach is yet to be determined to enable this so as to replicate the success of Cloud environment achieved in general computing, in NDM. Moreover, due to huge dependency of Cloud Computing on internet connectivity and regular power supply, the use of Cloud services can be difficult during the actual occurrences of the disasters when the communication and electricity infrastructure may break down. Cloud Computing offers better solution for disaster modeling and simulation but easy and efficient access to Cloud infrastructure, specially during the disaster, is still one of the key challenges.

A wide range of work to integrate Cloud Computing technologies with disaster management is found across the research domain. A comprehensive reflection of current research trends to highlight the existing research gaps, needs and problems is required in this research area. This clear picture of the research trends is expected to lay a strong foundation for well-directed future works for more effective NDM to ensure minimal losses inflicted by natural hazards to the global community. There are some attempts made to highlight the current research trends in adaptation of ICT in NDM. Yang et al. [52] explained how Cloud Computing could shape the future of Geospatial Science for advanced functionalities and capabilities taking four works as use cases in their work. Hristidis et al. [55] presented a comprehensive survey of data management and analysis in disaster situations to present the current state of knowledge, challenges and future research directions. A survey along with five papers are presented by Yang et al. [18] showing how Cloud technologies were capable of addressing the issues of Geospatial Science. None of the studies done so far have summarized the work done to integrate evolving Cloud technologies to support various aspects of disaster management, giving a clear picture of the current research state. As such, this proposed study aims to fill this gap by presenting a synthesized and comprehensive summary of relevant works to reflect the current research trends and future research directions.

2.2 Background

This section briefly explains the basic concepts of Disaster Management, Geospatial Science and Natural hazards, use of ICT tools in NDM, Cloud Computing, and use of Cloud Computing in Geospatial Science for NDM.

2.2.1 Natural Disaster Management and Its Aspects

Natural disasters, whether caused by natural or human-induced factors, cause large-scale destruction of the environment and physical infrastructure and directly threaten lives. It is a difficult task for authorities to formulate and implement effective strategies to minimize the dangerous impacts of the disasters. There is a wide range of activities that can be specifically directed and carried out at different stages of a natural disaster but an effective management of these activities, commonly referred to as *Natural Disaster*

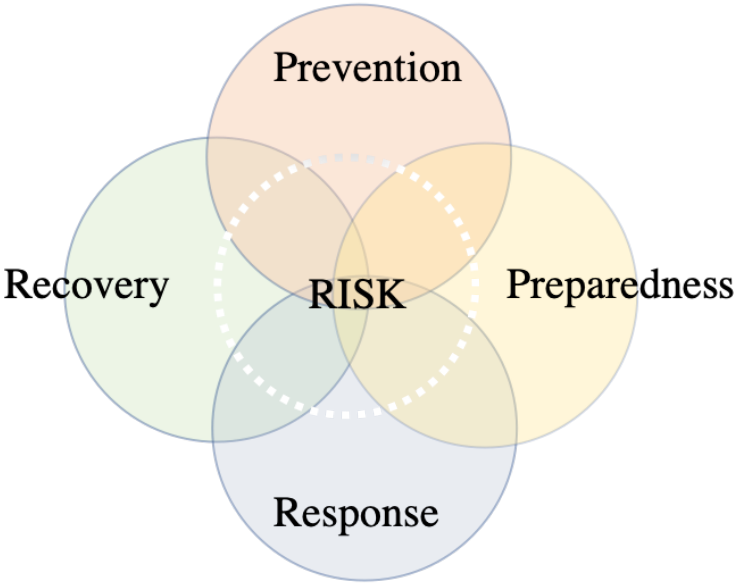


Figure 2.1: Comprehensive Approach to NDM [1]

Management(NDM), is required to ensure least damages are inflicted by the disaster. The comprehensive approach [1] has been widely used in NDM. This approach comprises of four phases namely - prevention, preparedness, response and recovery and is commonly referred to as PPRR framework for disaster management. Figure 2.1 show the four phases in PPRR framework which are not linear and independent as they overlap and support each other for a better balance between risk reduction and community resilience for better response and effective recovery.

2.2.1.1 Prevention

The risks of some natural disasters can actually be reduced or eliminated by carrying out proactive and counter-effective measures before the occurrence of the disasters. The possibility of prevention of the disasters is based on the factors contributing to the outburst of the disaster. The occurrence of flooding events can be prevented by erecting and reinforcing dams around the rivers or finding an alternate way out for the water in case of increased water level as suggested in the work [56]. For the disasters whose occurrences can be prevented, necessary actions can be taken after analyzing relevant information so as not to concede any loss to the disasters.

2.2.1.2 Preparedness

For disasters which cannot be mitigated, responses can be prepared by analyzing current information on the disaster to reduce potential impacts. For example, faster than real time models of disaster outbreak can predict which areas will be impacted as done by Cohen et al. using Swift citeswift for urban flood prediction and Miller et al. using Spark [3] for wildfires. Evacuation strategies can subsequently be developed accordingly. For earthquakes, preparatory actions could include managing open spaces for communities and forming effective strategies for deployment of earthquake-response units as highlighted by Allan and Bryant [57]. Co-ordinated action and plans as emphasized in [1] are necessary for an effective preparedness against any natural disaster.

2.2.1.3 Response

Response and resource mobilization during a disaster is critical in saving human lives and reducing physical losses. Authorities can acquire, collect and analyze real-time information about the disaster to form effective strategies for effective response. For example, search and rescue operations carried out during a disaster can be improved by making effective use of technical tools, like monitoring tools and communication methods as studied by Fiedrich et al. in [58] against earthquakes.

2.2.1.4 Recovery

It can be very complicated and protracted to recover and return to normal life once the disaster has inflicted damages to the community. The recovery efforts should align with the need of the area affected by the disaster for best outcomes. Post-disaster, the damage in terms of lives and economic value must be assessed using appropriate cost assessment methods for better reconstruction phase after the disaster as summarized in [59]. Authorities employ various methods for collating data from the event which is used for the prioritization of infrastructure repair as in [60] and to guide future management strategies and plans.

2.2.2 Geospatial Science and Natural Hazards Models

Geospatial Science, also referred to as Earth Science, is the study of various physical constitution and components of the planet and its atmosphere. Geospatial Science comprises the studies of the earth's physical characteristics ranging from the raindrops to fossils including earthquakes and floods. The scope of Geospatial Science can be huge, with complex interactions between different components. To study and understand these complex phenomena that occur around the planet, Geospatial Science uses different models that provide a picture of the past, present and future of the natural systems and processes.

Modeling is crucial in Geospatial Science as it helps the researchers to simulate the complex physical processes of earth systems [61]. For example, climate models simulate the future climatic conditions and changes for years to come simply by simulating the interactions among different factors such as atmosphere, land surfaces, biosphere, ice, and oceans using the past climatic condition records [62]. The same approach of modeling is used to study the phenomena of natural hazards to predict their outbursts. For example, a model can be constructed for predicting the spread of a wildfire in a particular region by studying and simulating the complex interactions with several factors including vegetation, climatic conditions, fuel models, altitudes, chemical reactions and turbulent interactions with the atmosphere.

The implementation of simulations in Geospatial models possesses a number of challenges. The highly complicated nature, compute-intensive nature, specific time requirements, need for scalability for ensembles of simulations and data-intensive nature of Geospatial models are what make the implementation a complex process [52]. The complex interaction between all the influencing factors to the natural phenomenon makes the task of setting up a Geospatial model intensive with respect to computation and data. Most of the models require the complex simulation to be repeated a number of times for different points in the region being considered. The natural models for weather, hazards and dust predictions should run and complete within a specific time requirement as these predictions are time-sensitive and could make delayed predictions obsolete. These Geospatial models usually employ an ensemble of numerous simulations for more accurate risk metrics. These specific runs require scalable computing resources which can adapt to the ever-changing requirements of the models. Moreover, given the

recent advancements in the data collection techniques and number of inputs, a model could be dealing with a huge volume of data even for the little duration of time [51]. For example, the weather prediction model could be generating terabytes of data just for few days of prediction thereby making the data handling a challenging task in Geospatial models.

2.2.3 ICT In Natural Disaster Management

There are different aspects of natural disasters where effective management is required before, during and after the occurrence of the disaster. Depending upon the phase of the disaster, a wide range of ICT tools can be used for different activities so as to minimize the impacts of the disasters. The foremost step in the disaster management is to collect the relevant statistical data related with the particular disaster and correctly analyze and identify the risks and dangers associated with the disaster [1]. The next step is to look out for the measures that can be taken in order to prevent, mitigate and prepare for the emergencies caused by the disasters.

The use of ICT technologies can significantly improve the management of the disaster by performing different activities in efficient and convenient ways [63]. The use of Geographic Information System (GIS) allows the potential risks and dangers of a disaster can be identified and the geographical areas to be classified into different level of vulnerabilities for effective mitigation planning. The technological foundation of ICT can also be helpful in early warning systems which can help authorities and people save lives. The use of ICT tools can be crucial in the collection of information from multiple system and sources during the occurrence of the disasters and forming operational plans during the emergency. The transfer of critical information during the emergency can be implemented using various ICT tools for effective mobilization of resources. Along with the Remote Sensing and satellite data, ICT can contribute through visualization of real-time information after the disaster has struck. Moreover, the foundation of ICT can allow the execution of different simulations to predict the nature and spread of different natural hazards and make necessary arrangements and preparations accordingly to minimize the impacts of the disaster.

The use of web, web-based applications, communication tools and visualization platforms are pivotal in providing useful information about the disasters [64]. However, the

evolution of different Geospatial models for different natural disasters and parallel development of sophisticated data collection methods, the conventional methods of using ICT for disaster management have become outdated. The challenges of data-intensiveness, compute-intensiveness and concurrent access-intensiveness have been added to the disaster management making it a hugely complex task to handle. Cloud Computing has emerged as an attractive alternative to address the new challenges in the field of disaster management.

2.2.4 Cloud Computing

Given the need for elastic on-demand resources for parallel and distributed computation in various application, Cloud Computing has emerged as a new technology that exploits the principle of distributed computing in multiple virtual machines. The National Institute of Standards and Technology (NIST) has defined Cloud Computing as "*a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction* [17]".

The introduction of Cloud Computing has revolutionized the way computation is carried out in organizations and research. Computation is now considered as a utility service, rather than the traditional model of owning and utilizing resources for different application. This shift of computing paradigm facilitates the users to focus more on their application and spend lesser time on repairing and maintaining the resources. Irrespective of the ways Cloud Computing is defined, there are some inherent features which Cloud Computing is expected to possess. Cloud Computing provides an almost unlimited capacity for computation, storage and networking through its vast chain of virtualized resources ensuring key features of on-demand service. These include ubiquitous network access, independent resource pooling, rapid elasticity and a service-based approach. The concepts of different Cloud service models and deployment methods are summarized in Figure 2.2 to show how the Cloud environment can be used under different configurations.

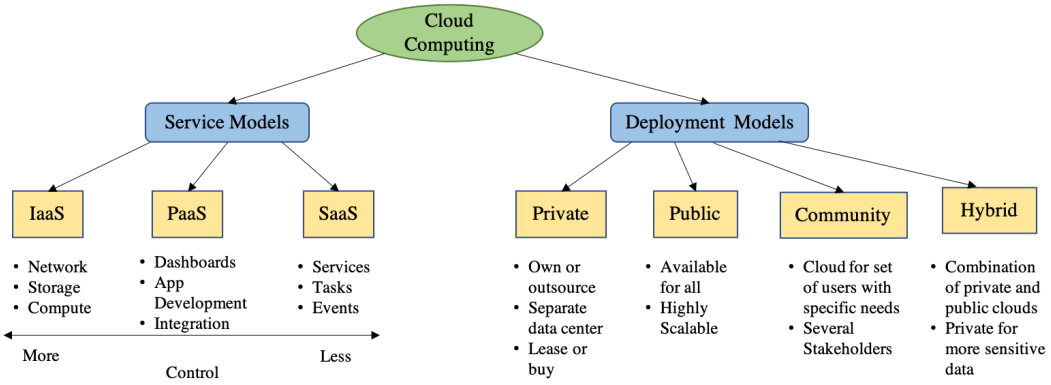


Figure 2.2: Cloud Computing: Service and Deployment Models

2.2.4.1 Cloud Service Model

Cloud Computing facilitates the consumption of Cloud services and utilities at different levels. As such, Cloud Computing has been classified into three distinct categories based on the services and abstraction levels at which it offers the advantages to its users. The three categories of service models are explained below:

Infrastructure as a Service (IaaS) IaaS stands on the lowermost layer of a managed Cloud service ecosystem providing virtualized and pre-configured hardware services. It provides the services of networking, servers, virtualization components and storage and the users have to take care of all other aspects of hardware including the installation and maintenance of the operating system, applications, databases and security components. Amazon Elastic Compute Cloud (EC2) is a good example of IaaS.

Platform as a Service (PaaS) PaaS manages all the hardware-oriented functionalities such as operating system installation and updates and security patches maintenance and provides a versatile foundation for developers to develop, test and deploy applications with a wide range of functionalities. It includes various APIs and tools to facilitate monitoring of services, version control of systems and work division. Microsoft Azure and Google Cloud Platform are well known PaaS solutions.

Software as a Service (SaaS) SaaS is a service offered to end users through a web-based interface over the internet where the users have the least flexibility in terms of the environment and hardware over which the services are running. The users do

not have to worry about development, update, backup, support or maintenance of the services as the service provider takes care of everything. Gmail, Dropbox, and Netflix are popular existing SaaS services.

2.2.4.2 Cloud Deployment Model

There are different ways how the Cloud services are deployed to offer various services to its users.

Private Clouds In a private Cloud, a business firm is the only entity that has access to the Cloud services as the Cloud services are not shared with anyone else. The firm deploys its own applications and services that are accessed by the personnel inside the company through intranet over secured connections. The payment system is often a fee-per-unit-time based scheme.

Public Clouds In a public Cloud, the business firms access the Cloud services provided by a Cloud service provider and hence, multiple business firms can access the same Cloud infrastructure based on the subscription schemes. The Cloud service provider maintains the security in the Cloud services to deny any unauthorized access to the services. The payment scheme is usually a pay-as-you-go model based scheme.

Community Clouds In community Clouds, specific business communities can have access to a complete Cloud solution provided by a Cloud service provider. The Cloud infrastructure are shared by the business firms but they have their own private Cloud space so as to meet the common privacy, security and compliance needs of the community. This model can be helpful in providing the complete Cloud solutions to business entities with a common interest to meet their specific needs.

Hybrid Clouds In hybrid Clouds, the Cloud deployment lies between public and private where sensitive and critical data are stored in private Cloud for the highest level of security while other operations are carried out in public Clouds. Hybrid Clouds can help business to reduce the costs by providing the option of running all their services over the public Clouds without comprising their sensitive data.

Table 2.1: Pros and Cons of Cloud Computing

Pros	Cons
- Near-infinite capacity of compute, storage and network - Reduced Capital Expenses - Ubiquitous Access - Redundant Data Storage - Scalable resources - Flexibility and Mobility - Reliable services	- Latency related Issues - Security Issues - Compliance and Regulatory Issues - Interoperability Issues

2.2.5 Pros and Cons of Cloud Computing

With its vast network of physically distributed data centers, Cloud Computing has advantages of reduced capital costs, robust and redundant data storage, ubiquitous access and on-demand and scalable resources [65]. But, Cloud Computing is hugely dependent on the internet connectivity and the power supply that operates the data centers. Because of the remote location of the Cloud servers, there may be latency and bandwidth related issues [66]. In addition, there may be issues related to security, compliance and regulation [66]. Because of multiple Cloud platforms, developing services may have interoperability issues. Despite these cons, Cloud Computing offers a more robust, reliable, scalable and cost efficient solution compared to local computers and small cluster of computers. Specially for the disaster management, because of its features, Cloud Computing stands as an indispensable entity, which can be used in conjunction with other evolving technologies for the most effective use. The pros and cons of Cloud Computing are summarized in Table 2.1.

2.2.6 Cloud Computing in Geospatial Science for Natural Disaster Management

The challenges of the compute, data and concurrent-access-intensive nature of Disaster Management models as end services make traditional computing infrastructures less fit for purpose than Cloud Computing. The additional needs of scalability, dynamic reconfiguration, easy access, and distributed operation of the models have also make a Cloud Computing foundation an attractive choice as Cloud technologies have the potential to provide support for all those needs. Given the rise of Cloud Computing infrastructures for the deployment of various services and applications, researchers have

been looking to Cloud Computing to address the challenges and issues associated with Geospatial Science for Disaster Management. Cloud Computing provides new capabilities to Geospatial Science with its almost unlimited capacity of computation, storage, and networking resources to handle the associated challenges.

Geospatial Science encompasses sectors such as energy and mineral science, climate science, ecology, environmental health, water management, disaster management and traffic management. Cloud Computing has had limited success in these areas due to the low levels of current integration between Cloud Computing and Geospatial Science. Li et al. [67] used features of Cloud Computing to address the complex demands of data, storage, and processing for energy information management. The challenges of large-scale data management, analysis and processing of climate Science were handled using Cloud Computing by the introduction of community defined services such as Earth System Grid [68]. The need of real-time capabilities to solve data-intensive problems and offer on-demand services to a dynamic number of end users in traffic management and surveillance was addressed by Li et al. [69] by using Cloud Computing. The inherent challenges of ecology in regard to storage, scalability, platform integration and deployment were addressed by the use of Cloud Computing in conjunction with Geospatial Science [70]. The Cloud Computing also facilitated the support for ensemble runs for predicting and forecasting the availability of freshwater and spread of different natural hazards [23]. The needs for flexibility and extensibility in visualization, monitoring, warning, preparing and responding to fire disasters were also met with the introduction of Cloud Computing [71].

The work carried out so far has illustrated how the use of Cloud Computing technologies has brought in various advanced capabilities in the implementation of models in Geospatial Science for NDM. Further work can be developed on this foundation to offer functionalities of NDM as services to close the gap between these Geospatial models and their users.

2.2.7 Challenges in Implementation of Disaster Models as Services

Natural hazard models developed using Geospatial principles can be contribute to understanding the complex nature of natural disasters and reducing their impacts. However,

the ability to easily and conveniently use these models as end services are prevented by a number of challenges which are described below:

2.2.7.1 Compute-Intensive Nature

The models and algorithms are generally very complex as they are based on physical models with additional relationships between various model components. The development of new technologies has contributed to better understanding the phenomenon [52] but has increased the implementation complexity due to the large datasets produced. The computational power required to support these models has also drastically increased and consequently, traditional sequential computing techniques and single machine are not able to keep up with the increased computation demands. Natural Hazard models now require a high-performance computing scheme to be able to meet the increased computation demands, which is not possible for every organization wishing to use such models.

2.2.7.2 Data-Intensive Nature

The scale of recent advancements in data sensing technologies means that Geospatial Science must now handle massive data sets. Cui et al. [51] highlighted the support of massive data as one of the long-term bottlenecks in Geospatial Science due to the amount of data accumulated by in situ sensors and satellites. Satellites currently collect petabytes of Geospatial data annually(more than 4 petabytes in 2019) [72]. Moreover, the scattered nature of data, non-uniform formats, diverse temporal scale of incoming data and service types of Geospatial models result in significant challenges in the organization, administration and processing of the data.

2.2.7.3 Concurrent-Access-Intensive Nature

The rise and success of web and wireless devices has enabled a large mass of end users to access Disaster Management services concurrently from a diverse range of geographical locations [73]. These web-based services must offer customized services to end users based on user requirements and sets of user inputs. Additionally, these services must have the ability to provide fast access and respond to sudden change in the number of

concurrent accesses to the services. The number of users of a disaster model service can peak during the event of occurrence of the disaster while the number of the users may be low during other time. The ability to respond to these access spikes is a key requirement for a disaster management service [53]. Effective management of the resources must be realized for optimized and uncompromised user experience while facilitating concurrent access to the services.

2.2.7.4 Time-Critical Requirements

Based on the complexity and level of interactions between different factors for particular natural disasters, the implementation of Geospatial models and processing can be time-consuming. Specially, for predictions from natural disaster models the time taken for producing the results and relevant alerts are highly critical to operational management. During the occurrence of natural disasters, any prediction results obtained quickly about the spread of the disaster could be crucial in saving and preventing further damage and loss. Given the complex natures of the models in Geospatial Science, it is a challenging task to handle the resources so as to be able to meet the strict time-critical requirements of the models and services.

2.2.7.5 Inaccessibility of Cloud Infrastructure during Disasters

Cloud Computing is hugely dependent on the internet connectivity and the regular power supplies that keep the data centers running. Depending upon different forms of the disasters, the infrastructure for communication and electricity can be significantly damaged. The communication infrastructure was non-functional for a prolonged time due to an earthquake in 2011 in Japan [74]. Similarly, the regular power supply was reported to be interrupted frequently because of different natural disasters such as hurricane, earthquakes and so on [75]. In such disaster circumstances, despite the fact that Cloud Computing offers attractive solution for disaster modeling and simulations, Cloud services cannot be easily accessed. As such, determining the effective ways to either make Cloud services accessible through other alternate methods or integrate other related technologies for better response during the disasters, is still an open challenge.

2.3 Proposed Cloud-based Conceptual Solution

This section proposes a conceptual Cloud-based solution for easier integration of Cloud Computing technologies with Geospatial Science for delivering NDM capabilities. The proposed solution aims to expose the capabilities of Cloud Computing to complex disaster management models in order to address the challenges associated with offering Disaster Management as end services. There are three major blocks in the proposed concept that handle different tasks independently focusing on specific aspects of the entire system. The User-interface is the one and only point of contact between the users and the Cloud-based system in which users can initiate requests and get a desired output after suitable processing and execution. The Cloud Infrastructure block provides all the hardware capabilities (compute, storage and networking) required for the execution of any processes and simulations as initiated by the users. The Control Mechanism block is central to the proposed solution as it governs all the mechanism for handling and managing the user requests and Cloud infrastructure to produce the desired output in an optimized manner. The block of other related technologies such as IoT network, fog and edge computing, is an extension under the umbrella of Cloud Computing that offers some time-critical and less compute and data-intensive disaster-related services and acts as a transitional data ingestion point during the disaster due to the realistic fact that the Cloud services may not be accessible because of communication and power supply breakdown during the disasters. The composition of the proposed solution is shown in Figure 2.3 while the desirable features of each block are summarized in Figure 2.4. Each components and how the proposed Cloud-based solution can be used effectively, are described below.

2.3.1 Component Overview

2.3.1.1 User Interface

The first block in the proposed conceptual solution is a user interface block that offers an interface to different end services as facilitated in the entire system. This is the front-end of the Cloud Computing based system and accessible to the users through the use of web services and different application program interfaces (APIs). The user interface block is critical to the system as the block encapsulates the entire operation of the system and is

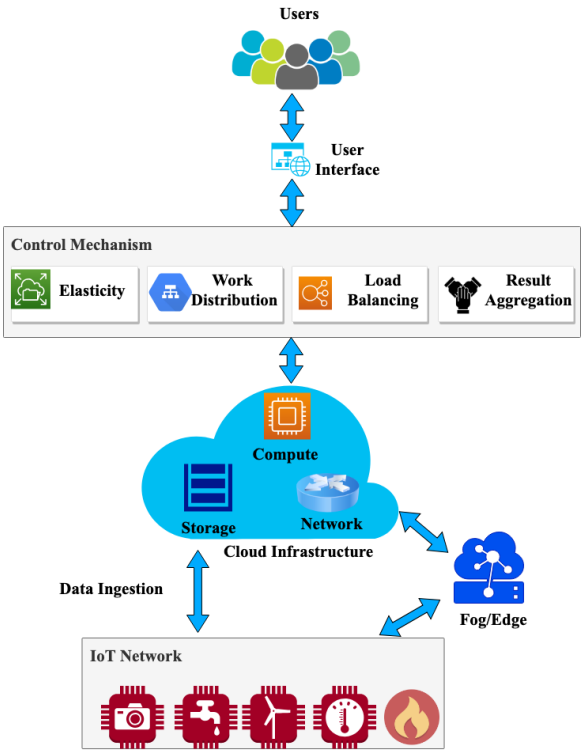


Figure 2.3: Proposed Cloud-based Conceptual Solution

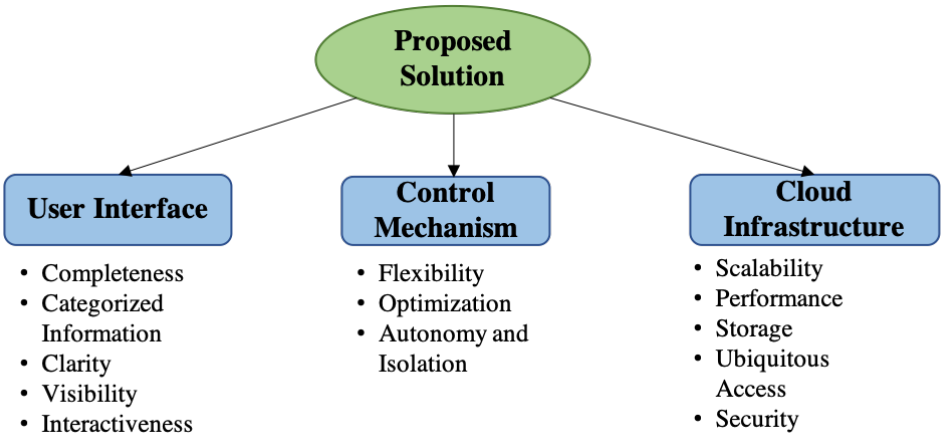


Figure 2.4: Desired Features for the blocks in Proposed Solution

the single point of interaction between users and the system. The block should facilitate the initiation of the user request and collection of the results. The user should be able to use the functionalities of the entire system and execute models through this block, including entering input parameters and displaying results obtained from the models.

Desirable Features

- Completeness. For a system implementation of NDM models, the user interface acts as the single point of the control for the users and the system that runs the required operation to produce relevant sets of outputs. It is therefore critical that the block remains complete at any instant. The block should be able to provide complete information to the system and users. The block should ensure the system can acquire the complete set of input parameters for running models in the system. This also holds whenever the block is interpreting the results obtained from the system to users and the interface block should be able to provide complete information about results obtained after complex runs of the model. It is desirable that all the information held over the block before, during and after the run of the models is complete.

- Categorized Information. The users of NDM models can comprise a diverse range of people with varying level of knowledge about the related phenomena. When offered as end services to these users, the user interface should be able to represent the relevant information that is useful and understandable to any categories of users. The dispense of information related to any Geospatial process or any disaster should be managed under different categories and reflected in the user interface with proper isolation for any confidential or sensitive information through proper security measures. Users should be able to derive and understand the important and desired information from the system through the use of user interface irrespective of the role they play during the disasters or any earth processes.

- Clarity. Depending upon the processes of various Natural Hazards, the system may take up a range of inputs to produce a large set of outputs. The types of operation that are carried out to produce the desired outputs can also significantly vary. From a user perspective, the interface is the only point of contact with the system and hence the interface should try to maintain a clear line between different aspects

of the model. There must be a clear picture of input parameters and how they are likely to govern the operation of the entire system. The user interface of the system should have a distinct line of clarity to dispense any information related to the status of the system or results obtained from the model run so as to make them easily understandable and readable.

- Visibility. Given the compute-complexities of the Natural Hazard models, the system may not be able to produce the desired results instantaneously and the users might have to endure significant waiting time. As the user interface block is the sole representative of the entire system architecture, it is desirable that the block represents the operational states of the system at different instances of time. Along with the user inputs and the results obtained after the run of the model, the user block should ensure visibility of the system status to facilitate easier display and interpretation of any information related to the system to improve the quality the user experience.

- Interactiveness. The models for NDM comprise of a large set of parameters and input data sets that are considered during the construction and implementation of a particular scenario. The ways in which inputs are entered into the system and how the results are displayed over the user interface are important as users of these services may have customized steps and visualisations of the results. Given the crucial nature of the user interface block in the conceptual solution, it is important to create and maintain an interactive experience for the users of the service. The block should add some elements of interactiveness while accepting inputs from users and interpreting the results obtained from models. The system should have a wide range of interactive options for displaying the results. The block could add options to toggle between various visualization options for the users to interpret the results from different perspectives and display any required useful information.

2.3.1.2 Control Mechanism

This block is the central component of the proposed solution as it employs various methods to prepare the existing Cloud infrastructure for complex NDM models. The different aspects of Disaster Management need to be managed effectively using relevant control

methods so as to implement the models in a distributed fashion over the Cloud environment. Moreover, to effectively deal with the complexities related to the implementation of the models, there have to be different mechanisms for handling different aspects of the system independently. This block is basically a compilation of different functionalities that enables the smooth run of various operations in the system. The wide range of methods ranging from achieving the distributed mode of operations to optimization of the performance in terms of cost, time and resources is defined in this block.

Functionalities

- Elasticity. The access of the service models offered with different functionalities of Disaster Management models can vary significantly based on the time within a year. For example, access to services related to bushfire would see a spike in the access and usage over the fire season and lower usage during other seasons. For flood, the maximum scale of access and usage of the services will be during rainy periods of the year with lower access and usage during other periods. As such, there should be an effective mechanism to handle this irregular pattern of usage and access so as to ensure better usage of the resources within the system. During a spike in user access, the mechanism should be able to add more resources in the system pool to provide an uncompromising system performance to the users and during the minimal usage, the mechanism should scale down and cut down on the resources to eliminate wasted resources.

- Work Distribution. The end services of delivering the functionalities of a Disaster Management System are complex as they comprise of a wide range of aspects related to the natural disasters and related processes. The system must not only handle multiple tasks at any given instant of time, but also take into account the diverse nature of the tasks. The tasks can vary from visualization to complex ensemble runs and handling the wide range of particular tasks types can be a complex process. A mechanism must be defined as to how multiple tasks of similar natures are to be grouped, where tasks of particular nature are to be carried out in the particular computing section of the system and how multiple tasks with diverse natures can be divided to ensure minimal cost. There should be an effective and efficient control mechanism for dividing the work obtained at any instant of time

in the system to different nodes in the Cloud infrastructure in such a way that the desired outputs are obtained with optimized use of resources and minimum cost and time.

- Aggregation of Results. Geospatial processing can employ ensemble run of simulations with inputs values drawn from statistical distributions. These ensemble runs assign different jobs to a large number of computing nodes. Outputs from each of the processing nodes are crucial for the accurate presentation of the result. There must be an effective mechanism to keep track of the order of the outputs generated by each computing nodes as the jobs during the operations can be distributed in highly parallel fashion. Intermediate results may have to be stored for a final reduction step at the end stage of the ensemble run. An effective control mechanism must be integrated into the system to handle the large sets of output files generated during the operation and subsequent post-processing.

- Load-Balancing. The ensemble runs of Geospatial models employ a number of computing nodes for a single operation. Outputs from each of the processing nodes are crucial for the accurate delivery of the final result. Whenever a fixed number of computing nodes are assigned to a number of ensemble runs, there may be instances where one of the nodes completes the jobs early while the other takes more time due to a particularly complex set of input conditions. Given the nature of the computing devices used in the Cloud infrastructure, the failure of machines during the operation must be taken into account, even though the rate is quite low. The control mechanism block should adopt an effective measure to balance the load by migrating the jobs from one computing node to another in case any of the nodes finish the job early or fails.

Desirable Features

- Flexibility. For a diverse range of users from different geographical locations, the system must be flexible enough to switch between operations and produce desired outputs without compromising performance. There should be flexibility in the control mechanisms as this determines how the entire system runs. Depending upon the availability of the resources in the system and the requests made by the users, the control mechanism should have the ability to vary and configure the

operation of the system to produce the suitable outputs. If any changes in the control mechanisms are required at any instant of time, the control mechanism should be able to be changed without significantly altering any other components of the system [76]. The control mechanisms employed in the system cannot take static or rigid forms as adaptive techniques have to be incorporated to the control mechanisms to make the system operable under any dynamic conditions [77].

- Optimization. Whenever an ensemble of runs for a Geospatial process are carried out, a large amount of computing resources must be utilized [21, 23]. These computing resources can be expensive and using them in a non-optimal way can result in a significant waste of resources if used to offer end services. The control mechanism block should employ an effective measure to optimize the computing resources used in each run of the system. The facilitation of the centralized result storage system can help in further optimization of the resources in the system. Filtration and sorting out mechanism can also be helpful for optimization of the resources in the system [78].

- Autonomy and Isolation. The number of the processes that may be run during the implementation of a Geospatial process can be very high. Repeated interaction between the processes can slow make the entire system due to the large-scale exchange of data. Moreover, the operation of the system for Disaster Management is highly distributed and it is desirable to have the least exchange of data between the computing nodes during processing. As far as possible, each run of a simulation in an ensemble in a computing node should be made as independent and autonomous as possible to decrease the overhead and network bottleneck in the system architecture [79]. The control mechanism block should define functionalities to maintain the independent mode of operation.

2.3.1.3 Cloud Infrastructure

The Cloud infrastructure is the foundational block in the system architecture that eliminates the need of having local computers or powerful servers to be able to simulate a scenario. NDM models require compute-intensive machines to support and run the ensemble simulations and aggregate the results for better interpretation. This block of the system architecture transfers the complexities to the Cloud infrastructure. This tier

should provide the foundation to conduct massive computation with huge data sets with advanced networking requirements. The Cloud Computing infrastructure can be chosen from any Cloud service providers as long as the service provides the basic features of the Cloud Computing in terms of scalability, fault-tolerance, security and other related aspects. Based on the functionalities, Cloud infrastructure can be used for different applications listed below:

Functionalities

- Computational Applications. Geospatial processes and NDM models are complex and the hardware requirements to support these models can be significant, making the local desktops and computers obsolete in terms of time performance. On the other hand, Cloud Computing can create a virtual pool of any number of computing nodes connected together to address the large computational needs of any system. For any Geospatial and Natural Hazard model, Cloud infrastructure can easily handle the computational needs from simple analytical processing to large-scale ensemble runs of simulations. The computing nodes in the Cloud infrastructure can be easily scaled for better utilization of the computing resources. The scaling can either be upwards in horizontal or vertical fashion for compute-intensive applications, or downwards in the same manner for less compute-intensive applications. An additional mechanism can be integrated with the Cloud infrastructure to ensure the optimized utilization of computing resources in terms of cost and time.

- Visualization. A wide range of visualization tools can display the important information and outputs obtained from Geospatial processes and models for disaster management. Depending upon the tools used, the hardware required to support the visualization of the results and information can be significant. As such, Cloud infrastructure can be used to visualize a number of components ranging from simple analytical results to complex statistical result sets obtained as outputs from ensemble runs. Clouds can provide a flexible, scalable and dynamic solution for visualization of different components of various models when it comes to user-focused service models. Customized and sophisticated techniques can independently be integrated into the system to provide quick and complete information to users and

authorities for effective decision making. Cloud infrastructure can be used to develop a visualization platform for result data, location-based resources information and resources mobilization for better decision support. The visualization platform over the Cloud infrastructure can compliment the rigorous operations of different aspects of the model in an interactive way.

- Storage. The recent advancement of data collection technologies and new Disaster Management models can result in massive datasets for various operations. Moreover, the nature of frequency of access to these datasets can be irregular as there might be a spike in the access of the disaster data during the peak occurrence periods compared to other times of the year. The near-unlimited capacity of storage of Cloud infrastructure is an ideal solution to address the data intensiveness of Geospatial models. The frequency of occurrences of natural disasters and associated data information can easily generate a huge volume of data that do not just require a storage media but also analytical and complex processing. When a number of users require seamlessly access to the data from different locations, Cloud infrastructure can provide a solid solution for an effectively managed data archive system for any Disaster Management models.

- Data Management. For the implementation of a Disaster Management model, data from a diverse sources have to be collected to be able to run the simulation for producing the relevant and important results. For example, wildfire prediction models require data for topography, fuel characteristics and land coverage of the considered area, as well as a range of meteorological information such as wind, air temperate and related factors. As such we require a strong hardware base that can effectively handle all the distributed chunks of data required for the model. Cloud Computing can create a pool of virtually connected data centers for storing massive sets of data. The act of handling and managing the large chunks of data stored over the Cloud can be non-trivial and the effectiveness of data management techniques is determined by how quickly the data can be fetched from the storage for further processing and representation. Because of the distributed architecture of Cloud Computing, effective and advanced data management techniques can deliver faster and accurate representation of data to the concurrent users located at different geographical locations for use in their models.

Desirable Features

- Scalability. Given the compute-intensive nature of models, the system architecture may require a number of computing nodes over the Cloud infrastructure. Rather than just increase the time taken for the operation to be completed, the Cloud infrastructure should facilitate the easy scaling out of the infrastructure in terms of the number of processing nodes following different constraints set by the control mechanism of the architecture [78]. The trade-off between the vertical and horizontal scaling of the Cloud infrastructure is handled by the control mechanism but, the Cloud infrastructure should have enough resources to provide the system architecture with that capability.

- Performance. It can be critical for specific disaster models to be able to produce results within designated time windows. Whenever such models are offered as end services there are various performance factors such as cost that need to be considered. The performance of the entire service model is dependent upon the performance of the Cloud hardware and hence the performance of the hardware should not just support the operations but also be consistent. The Cloud infrastructure should be able to provide superior performance under any range of service requests by employing proper control mechanisms [51]. The hardware in the Cloud infrastructure should maintain the same level of performance even when subjected to the higher traffic of user requests or tasks.

- Storage. NDM can result in massive datasets from models, sensors and tools [51]. During ensemble simulations this data might have to be held within the Cloud infrastructure. Moreover, the results produced by large number of simulations that run under a Natural Hazard model can significantly increase the storage needs of the Cloud infrastructure. The Cloud infrastructure should possess enough storage capacity to be able to address the data-intensiveness of Disaster Management model.

- Ubiquitous Access. Ubiquitous access to the Cloud infrastructure is critical to the entire system architecture as the system aims to offer the different functionalities as services. The access of the data and services should be possible from any location using any web services or APIs using a wide range of devices that have internet

connectivity [80]. The system should be able to access the Cloud resources easily so as to execute any necessary processing as required by the users.

- Security. The Cloud infrastructure needs to be secure as it should provide the results of simulation runs of disaster models to concurrent users. The user data should be kept intact and separate during concurrent handling of the user requests with the adaptation of various security measures [81]. The Cloud infrastructure should have security features to maintain the data integrity during the operation of the entire system. New security features and measures can be defined using the control mechanism block.

2.3.1.4 IoT Network and Fog/Edge Computing

The disaster scenarios can be best represented by different disaster models if the real-time data can be fed into those models. The updated data can help create a better situational awareness during the occurrence of the disasters for more effective response against the disasters. As such, collection of real-time and live data during the emergencies is possible with the evolving technology of IoT. An extensive network of different kinds of sensors can be created in an affected area to collect as much relevant information as possible. During the occurrence of disasters like fires, the real-time data can be collected from a wide network of different sensors like temperature, wind, humidity, rain and fuel-types and other types of connected devices carried by response teams and people, at a station closer to these devices or at Cloud servers depending upon the communication methods available.

In the proposed solution, Cloud infrastructure provides a robust solution to different disaster-related services in an effective way. This can be hindered during the actual occurrence of the disasters as the communication infrastructure and the regular power supply, on which Cloud services are primarily dependent on, may break down. As such, the new paradigms of computing, edge and fog computing have been considered under the umbrella of Cloud Computing for more time sensitive and critical services during the emergencies. In the proposed solution, based on the complexity and sensitivity of the services, processing of the data retrieved from the sensor can be pushed closer to the sensor network to trigger different actions prior or during the disasters. Specially, during the event of the disasters with limited connectivity and power supply, the end

devices like smart phones and routers can be used to create an ad-hoc network to collect critical data and perform computations to determine an effective way of responding to the emergencies. Moreover, whenever possible, the on-premise computing devices and local supercomputers or similar High Performance Clusters (HPCs) can be used. All the data and operations upheld at end or local devices because of limited connectivity, power supply and response time should be forwarded to Cloud infrastructure for a long-term storage and more intensive computation to further assess the disasters.

2.3.2 Effective Use of Proposed Solution

The main idea behind the proposed Cloud-based solution is to enable different functionalities of natural disaster management as end services to be used by different actors (users) during various phases of the disasters. Various studies [21, 22, 82] have proved that the cloud-based solutions are more cost-effective than the on-premise systems for running disaster prediction models. Moreover, in addition to the sequential operation of the disaster simulations in an on-premise setup, if the simulations are parallely executed over the Cloud as proposed in this work, the prediction results can be obtained in less time, thus giving us more time for better preparedness against the disasters [65]. Given the cost-effectiveness and efficiency of the proposed solution, the government should be willing to pay for the expenses of the solution. The description of how the proposed Cloud-based solution can be effectively used during various phases of disaster is given below.

2.3.2.1 Prevention

For the disasters whose occurrences can be prevented, the complex disaster simulations based on different disaster models can be run over the Clouds as end services to determine the key causes of the disasters [83, 84]. Accordingly, measures can be prioritized in a particular region for preventing the disaster. An archive of information system can be maintained using Cloud infrastructure that provides a comprehensive coverage for all the disasters [85, 86]. Alerting and notification services can be developed based on the processing and analysis of the data collected using different sensors, over Cloud [71].

2.3.2.2 Preparedness

During the preparedness phase, running disaster simulations for the determination of risk metrics [20], analysis of crowdsourced data [85, 87], enhanced visualization and monitoring of different aspects [88, 89], processing of sensor data for regular alerts and storage of crucial real-time and live data [90, 91] can be done over the Cloud environment to stay better prepared against the disasters. Specially for the preparedness against the disasters, local computing resources, supercomputers or similar HPCs can also be used in a hybrid fashion [92–94].

2.3.2.3 Response

For the better response, the co-ordination of the entire rescue and search operation can be centered around the Cloud infrastructure with remote operations, information collection, intuitive visualization, meaningful monitoring and efficient evacuation plans [71, 85, 94–96]. Depending upon the nature of the disasters, proper evacuation strategies and mobilization of response units can be achieved through the computations carried over the Clouds [97]. For some forms of the disaster like fire and flood, even the general public (at different locations closer to the disaster-affected location) can use different services under the proposed solution to run disaster simulations to develop more effective strategies at an individual level. But, for some forms of disasters, the communication infrastructure and reliable power supply may be interrupted, making the access to Cloud services difficult. Nevertheless, the effective use of the proposed solution can be ensured by overcoming the communication breakdown during the disasters and using Cloud infrastructure in conjunction with IoT network, edge and fog computing as described in detail below.

Overcoming Communication Breakdown during the Disasters Some forms of disasters can completely wipe out the infrastructure required for any Cloud services to be operational at the affected location. This is true for the private clouds whose data centers are located in the affected areas. For a public cloud infrastructure, geographically diversified location of the data centers, replicas of the data collected and the remote operation can create a more robust infrastructure to coordinate the activities during the actual occurrence of the disasters [65]. As such, even with no or unreliable power supply

during the disaster, the public Cloud infrastructure at a distant location can be used to process and simulate different disaster scenarios to produce critical results that could be relayed to the affected areas for better rescue operations. There are various studies that have tried to enhance the robustness of the communication infrastructure before and during the occurrences of the disaster. In the pre-disaster scenario, a redundant network design, enabled by improved fault tolerance and several backup links, has been discussed in [98] and [99] for survivable communication networks. During the disasters, rapid emergency networks, based on portable nodes and end-user devices, can be created to enhance the connectivity [100]. The internet connectivity can be made possible using satellite, optical fibers, robust wireless gateways and vehicular access points by creating a mesh network based on these transportable nodes [101]. An ad-hoc network created by different techniques involving the mobile devices can ensure the connectivity during the disasters for critical communication [102]. Moreover, the unmanned aerial vehicles (UAVs) like drones [103] and Autonomous networked robots [104] can play a significant role in providing the connectivity to an affected area so that the critical data can be transferred to the central infrastructure of Cloud for further assessments and planning. The results obtained from the further assessment in Clouds can be disseminated to the affected area for more effective steps during the disaster, similar to the faster than real-time evacuation steps during emergencies calculated over Clouds [105].

Conjunction with other Computing Paradigms The proposed solution is an overview of how different disaster-related services, from complex ensembles to evacuation plans, can be centered around the Cloud infrastructure along with different other technologies(IoT, Fog and Edge) for better preparedness and response during the disasters. For critical and time-sensitive services during the occurrence of the disasters, the end devices in fog and edge computing should provide various services related to alerting, evacuation plans and rescue resources mobilization. The intensive sensor networks in an IoT environment can help offer different disaster-related services with real-time and live data to best reflect and respond to the disaster scenarios. For the utmost efficiency of the proposed solution, different operations and services have to carried out in different devices under various computing paradigms based on complexity, time-sensitivity and critical nature. Cloud Computing still stands as an inseparable component that is required even for other computing paradigms for better assessment and interpretation

Table 2.2: Scenarios for effective use of Proposed Solution

Factors	Level		
	Low	**Medium**	**High**
Compute-Intensive	Fog/Edge	Fog/Edge/Cloud	Cloud
Data-Intensive	Fog/Edge	Cloud	Cloud
Time Critical	Fog/Edge	Fog/Edge/Cloud	Cloud
Internet Connectivity	Fog/Edge	Cloud	Cloud
Reliable Power Supply	Fog/Edge	Fog/Edge	Cloud

of the situations. An overview of using Cloud Computing in conjunction with other computing paradigm for more effective disaster management based on different factors is given in Table 2.2.

2.4 Current Research Trends

This section examines and categorizes work done in using the foundation of ICT in relation to NDM and the adaptation of Cloud Computing for supporting the different aspects of disaster models. There have been a number of studies carried out to provide a range of end services related to natural disasters using various features and tools of ICT including Cloud Computing. Figure 2.5 represents how the related works are categorized into different headings to reflect the current research trends. The related works are explained in detail under different categories as follow:

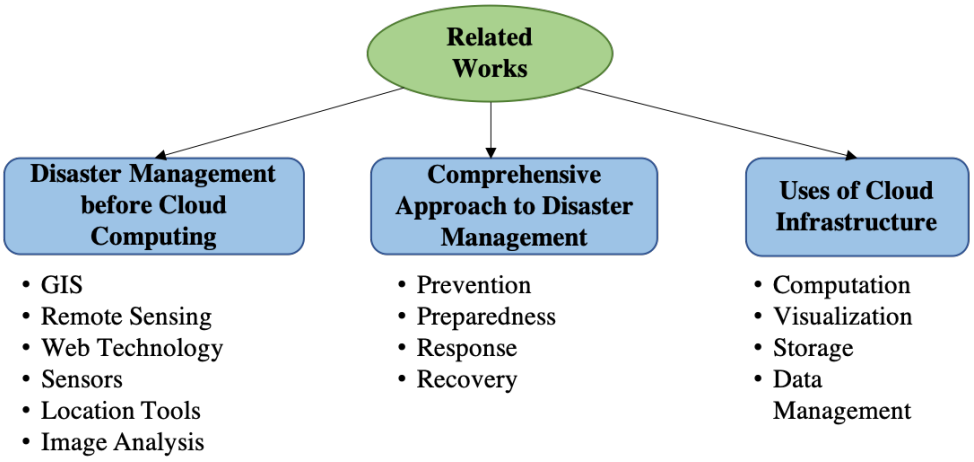


Figure 2.5: Categorization of Related Works

2.4.1 Disaster Management before Cloud Computing

A number of ICT tools ranging from Geographical Information System (GIS) tools to Image analysis were used to address various aspects of disaster management before the advent of Cloud Computing. GIS tools were used to produce and present the results obtained after spatial processing and analysis with additional geographical information for a better decision support. Pidd et al. [106] developed a prototype simulator capable of providing spatial decision support to emergency planners by integrating the geographical information within the simulator. Yong et al. [107] used GIS in conjunction with web technology to develop a decision support tool for identification of effective response strategies to strong earthquakes and assessment of expected damages and losses. Wex et al. [97] proposed a decision support model based on Monte-Carlo heuristics using geographical information for NDM that minimized the sum of completion times of incidents weighted by the severity of the incidents. The model was efficient during the emergency operations for allocation of available rescue units to any emergency incidents and scheduling the processing time of those incidents. Van Westen [108] demonstrated how Geographic Information System can be coupled with Satellite Remote Sensing to develop effective disaster management tools for prevention, preparedness, relief and reconstruction at different stages of the disasters. Laituri and Kodrich [109] added Internet GIS into the system to increase the effectiveness of the disaster response and management after high magnitude disasters. Jeyaseelan [84] validated the efficiency of using GIS intefrated with the Remote Sensing for early warning, real-time monitoring and damage assessment in any events of flood and drought. Manfré et al. [110] and Montoya [86] demonstrated the effectiveness of using GIS along with Remote Sensing and related technologies for better disaster and urban risks management respectively. Cutter [111] explained to what extent geo-information Science can be used by practitioner community for post disaster management.

The use of Satellite Remote Sensing was widely adapted to monitor the disasters and derive critical information before, during and after the occurrence of the disasters. Kerle and Oppenheimer [112] verified the ascendancy of Satellite RS over the use of sensors for better disaster management in Lahar. In a study carried out by Voigt et al. [113], efficient image analysis techniques were carried out on the multiple source satellite data to generate rapid maps for disaster and crisis management support. The study also

used the satellite data for rapid impact assessment after different disasters occurred at different corners of the earth. The work done by Tralli et al. [114] demonstrated how satellite Remote Sensing data can be effectively used in conjunction with multiple modeling for forecasting and visualizing the results for better decision support in case of the occurrence of natural hazards such as earthquakes,volcano, flood,landslide and coastal inundation hazards. The works [108], [84] and [110] explain the effectiveness of disaster management when Remote Sensing was coupled with other technologies such as GIS and Global Navigation Satellite System(GNSS). Montoya [86] developed a cost effective and rapid method of collection for an inventory based on Remote Sensing, global positioning system (GPS), digital video (DV) and GIS for urban risks management.

Web technologies have been used to accommodate different disaster related services for easier access and limited computation. Yong et al. [107] used web-technology for hosting the decision support system for disaster management that facilitated easier user access to the system. Different types of sensors were used to gather as much information as possible to derive better understanding of the disasters. Kerle and Oppenheimer [112] investigated the efficiency of using optical and radar sensors as tools for disaster management for lahars. 'People as sensors' was used as a concept in the system for effective response and management after high magnitude disasters. Geographic location information tools such as GPS and GNSS were used in [110] and [86] to annotate additional information of location to the information collected from other sources such as Remote Sensing and GIS tools. Efficient image analysis techniques were used in the work [113] to generate rapid maps on satellite data for better crisis management support. The summary of the ICT tools used for disaster management is represented in Table 2.3.

2.4.2 Aspects of Disaster Management

This section categorizes the research works based on different aspects of NDM. There are different aspects of natural disasters where activities can be focused in various ways so as to reduce the impacts of the natural disasters. Starting from the prevention of the occurrence of the disasters to the assessment of the damages caused by the disasters, every aspect is equally important to build effective strategies for better disaster management.

Table 2.3: Disaster Management Solutions before Cloud Computing

ICT Tools	Purpose	Related Works
GIS	Better Decision Support	Pidd et al., 1996[106], Wex et al., 2014[97], Van Westen, 2000[108] Laituri & Kodrich, 2008[109], Cutter, 2003[111] Jeyaseelan, 2003[84], Manfré et al., 2012[110], Montoya, 2003[86]
Remote Sensing	Monitoring and Data Collection	Kerle & Oppenheimer, 2002[112], Voigt et al., 2007[113], Van Westen, 2000[108], Tralli et al., 2005[114], Jeyaseelan, 2003[84] Manfré et al., 2012[110], Montoya, 2003[86]
Web Technology	Service Hosting and Wide access	Yong et al.,2001[107]
Sensors	Information Collection	Kerle & Oppenheimer, 2002[112], Laituri & Kodrich, 2008[109]
Location Tools	Advanced Monitoring	Manfré et al., 2012[110] Montoya, 2003[86]
Image Analysis	Map Generation	Voigt et al., 2007[113]

2.4.2.1 Prevention

Researchers have relied on risk identification, historical information, monitoring based on data processing and analyses and simulation of processes for preventing the actual occurrences of the disaster such as flood and droughts. The studies by Yu and Kim [83] as well as Jeyaseelan [84] identified the vulnerable regions for possible floods and droughts and helped concerned authorities to take effective measures to prevent the occurrence of the disasters. The historical information about the occurrences of the disasters was emphasized to make effective strategies to prevent the occurrence of flooding events by Wan et al. [85] and health issues related disasters by Shen et al. [115]. The extensive processing and analyses of multiple data were key to form effective strategies in the system proposed by Jiang et al. [116], Liu et al. [67] and Montoya [86]. The data processing framework proposed by Jiang et al. [116] facilitated convenient and highly available processing of the forest pest control data to build an effective strategy for forest pest control. The monitoring system developed by Liu et al. [67] focused on prevention of the disasters caused by magnetic storm facilitated by power system data, geomagnetic data, satellite data and other earth space observation data and their processing over the Clouds. Montoya [86] explored the use of low cost and rapid method of data collection for development of inventory based on combination of various technologies such as RS, GPS, Digital Video and GIS with multistage operations and analysis for prevention of disaster situations. Eriksson et al. [117] developed a Cloud-based architecture for simulating the pandemic influenza so as to be able to prevent the chaotic environment caused by the influenza.

2.4.2.2 Preparedness

The research works have adapted various methods ranging from monitoring enabled by geovisualization to running simulations for predicting the instants of disasters. The disaster monitoring enabled by visualization of data collected from different sources provided crucial information to general public about the spread of the disasters and helped them prepare against the impacts of the disaster. The web-based visualization service set up by Australia based on the Sentinel satellite [88] provides graphical information of wildfire events occurring all over Australia with well-categorized indexes based on time

to general public. The monitoring system developed by Zou [118] facilitated rapid information extraction from satellite RS data so as to stay prepared against possible disaster scenarios. Bohm et al. [119] proposed geovisual analytic solutions in public health sector for better planning processes to prepare and tackle the emergency situations. The Climate Engine developed by Huntington et al. [89] helped in visualization of climate data in an interactive GUI so as to stay prepared against any disasters caused by extreme climatic conditions. The work done by Tralli et al. [114] focused on the use of satellite RS data for construction of Geospatial models for monitoring the disasters for effective preparation against those disasters.

Many early warning systems have been developed to warn the people about the possible dangers of disasters and encourage them to stay alert [91], [90], [95], [71], [120], [84]. The system devised by Al-Dahash et al. [121] facilitated the early warning system based on efficient communication for preparing against dangers caused by terrorism in Iraq. Puthal et al. [90] presented a big data stream framework that supported the emergency event detection and generation of the alert by effectively analyzing the data stream. Rossi et al. [95] introduced a service-oriented Cloud based architecture that was capable of issuing early warning during the events of disasters. The web-based platform VirualFire [71] had the capability of issuing early warning in the event of a fire to general public for staying prepared against the disaster. The community-based Cloud system proposed by Li et al. [120] facilitated the issuing of early warning of disasters that was helpful in building preparatory strategies to minimize the impacts of the disaster. The study carried out by Jeyaseelan [84] was capable of issuing early warning for general public in case of any events related to flood and drought for better preparedness.

The importance of regular updates about the disaster along with regular exchange of information between different entities was highlighted in a system called CyberFlood developed by Wan et al. [85] that incorporated crowd sourcing technology for providing fresh updates on flooding events to enable general public to stay prepared against any water-related disasters. Furthermore, the architectural design of communication network proposed by Ali et al. [122] focused on effective flow of information for better preparedness against the disasters. The integrated approach devised by Zlateva et al. [123] performed the risk assessment of natural disasters to calculate the probability of occurrence of a particular disaster for the effective preparedness. The outspread of various disasters can be predicted to take better informed decisions to stay prepared against

the perilous disasters. SparkCloud developed by Garg et al. [20] facilitated the users to predict the spread of bushfires so as to form preparatory strategies to minimize the impacts of the disasters caused by fire. Huang et al. [21] formulated the forecasting of dust storm through ensemble run of the model to contribute to the preparedness against the emergency situations caused by dust storms. Li et al. [22] facilitated the run of ensemble simulation of different Geospatial Science models over the Cloud to predict the outburst of various disasters so as to develop effective preparatory strategies against the disasters. The Sentinel Hotspots system [88] maintained by Geospatial Science Australia in the Cloud environment provides visual information to public about the actual occurrence of the bushfire events in different time resolutions.

2.4.2.3 Response

Studies have focused on regular and quick information collection, better communication between response teams, efficient mobilization of rescue units and simulation of risks and evacuation plan for more effective response to the disasters to keep the loss of lives and physical structures to minimum. Wex et al. [97] proposed a decision support model based on different heuristics for effective allocation and scheduling of rescue units that formulated and solved the problem through the minimization of the sum of completion times for different events of natural disasters weighted by their severity.

Regular collection of information is necassary while responding to the occurrences of the disasters and many previous studies have examined this in the context of disaster response. The Collaborative Knowledge as a Service (CKaaS) proposed by Grolinger et al. [124] focused on the collection and integration of diverse sources of data over the Cloud environment for the disaster response management. Zou [118] devised a disaster monitoring system by proposing an interoperable framework to integrate a distributed model and data for rapid information extraction. Kerle and Oppenheimer [112] established the superiority of satellite imaging over optical and radar sensors for facilitating better disaster response management from lahars. The work done by Van Westen [108] advocated the use of Remote Sensing and geographic information systems for various phases of disaster management. A software called ERIC [125] was developed to automate situation reporting during any emergency situations by collecting data from a wide range of sources. The collected data was visualized using a web interface to respond in

better ways against any disasters. that weKlauck et al. [126] proposed flexible post-disaster management facilitated by continuous monitoring enabled by sensor data and interaction between the observers over the Cloud. The software architecture proposed by Rossi et al. [95] used data from different sources to produce observations for authorities and responders for Emergency Response services. Li et al. [120] maintained information repository for effectively handling the disaster response management with updated information collected from multiple sources. Voigt et al. [113] explained the use of satellite data collected from multiple sources and efficient image analysis for production of rapid-maps for better disaster and crisis management support. Manfré et al. [110] highlighted the use of technologies such as remote-sensing, GIS and GNSS for improvement in construction of effective emergency plans for post-disaster management at different levels. Based on the analysis of satellite RS data Tralli et al. [114] carried out reconstruction of land surface maps based on historical data for better mitigation and management in post-disaster situations. The users of cyberFlood [85] can access information about actual occurrences of floods from the data collected through crowd-sourcing technology. The work done by Jeyaseelan [84] highlighted the importance of Remote Sensing and GIS for real-time monitoring to provide rapid updates during the occurrence of the disasters such as floods.

Several studies have emphasized the importance of effective communication between different entities involved in disaster management for better disaster response [122], [71], [127], [121]. Ali et al. [122] proposed a network architecture with reinforced layers for effective communication to facilitate better post-disaster management. The VirtualFire developed by Kalabokidis et al. [71] incorporated web-based platform to share and utilize information and tools among firefighters for better coordination of firefighting efforts in the events of the fire. The metamodel for disaster management proposed by Othman and Beydoun [127] described how the semantic domain models could be built into an artifact for better knowledge sharing thereby facilitating the combination of different activities to manage the disaster on the hand in better ways. The work done by Al-Dahash et al. [121] provided concerned agencies to make better decision by providing properly managed communication during the emergency situation caused by terrorism in Iraq.

The simulation of risk scenarios, evacuation plans and risk assessment in various studies can provide important information for making better decision for responding to the

disasters. Qiu et al. [128] developed a smart evacuation system over the Clouds using smart phones and data centers to facilitate emergency decision system for faster disaster response. Alazawi et al. [129] proposed the modeling of impact of various disasters on the real transportation system of the cities for improving the flow of the traffic and smooth evacuation during the events of disasters. Pidd et al. [106] developed a prototype of decision support for use by emergency planners for effective evacuations from the disaster areas in the post-disaster scenarios. The stakeholders could input different desired risk scenarios into the platform developed by Aye et al. [130] for possible mitigation measures for better disaster response management. The risk assessment result obtained in the integrated approach proposed by Zlateva et al. [123] could provide the government with crucial information for taking more informed decision regarding the mobilization of the resources in post-disaster scenario.

2.4.2.4 Recovery

The use of an actuarial model combined with Remote Sensing has been used to assess the damages caused by disasters such as flood and droughts. The aggregated loss after a disaster could be accounted for using an actuarial model in the approach explained by Zlateva et al. [123] which was helpful in distribution of the available fundings for the population affected by the disasters. Jeyaseelan [84] emphasized on the use of Remote Sensing for quick damage assessment of drought and flood disasters.

2.4.2.5 Holistic Aspects of Disaster Management

Some previous work has addressed the facilitation of a range of services before, during and after the occurrence of a disaster, thereby addressing every aspect of disaster management. Adam et al. [131] examined the combination of social media and spatial computing for effective disaster management with different services including issuing alerts, data streaming, location services and data services. Habiba and Akhter [132] proposed a Cloud-based framework for enabling multiple services to facilitate better and effective disaster management. Tralli et al. [114] highlighted the importance of satellite RS data in reconstruction of land surfaces based on recent history for predicting the hazards due to various disasters such as flood, landslide, flood and coastal inundation

Table 2.4: Categorization of Related Works based on various aspects of Disaster Management

Aspects	Methods	Related Works	Disasters
Prevention	Risk Identification	Yu et al., 2018[83], Jeyaseelan, 2003[84]	Health Disaster,
	Historical Information	Wan et al., 2014[85], Shen et al., 2012[115]	Floods, Droughts
	Data Processing & Analyses	Jiang et al., 2010[116], Liu et al., 2012[67] Montoya, 2003[86]	Disaster Risks Forest Pest Disaster,
	Simulation	Eriksson et al., 2011[117]	Magnetic Storm
Preparedness	Monitoring	Australia, 2018[88], Zou, 2017[118] Böhm et al., [119], Huntington et al., 2017[89] Tralli et al., 2005[114]	Dust Storm,
	Early Warning System	Al-Dahash et al., 2017[121], Puthal et al., 2016[90] Rossi et al., 2017[95], Kalabokidis et al., 2013[71] Li et al., 2011[69], Jeyaseelan, 2003[84]	Fire, Terrorism, Flood, Health Disaster, Other Risks
	Effective Communication	Wan et al., 2014[85], Ali et al., 2015[122]	
	Risk Assessment	Zlateva et al., 2013[123]	
	Simulation	Garg et al., 2018[20], Huang et al., 2017[54] Li et al., 2017[22]	
Response	Rescue Unit Mobilization	Wex et al., 2014 [97]	
	Information Collection	Grolinger et al., 2015[124], Zou, 2017 [118] Kerle & Oppenheimer, 2002[112], Van Westen, 2000[108], Klauck et al., 2011[126] Rossi et al., 2017[95], Li et al., 2011[120] Voigt et al., 2007 [113], Manfré et al., 2012[110] Tralli et al., 2005[114], Wan et al., 2014[85] Jeyaseelan, 2003[84], ERIC, 2018[125]	Flood, Fire Droughts,
	Effective Communication	Ali et al., 2015[122], Kalabokidis et al., 2013[71] Othman & Beydoun, 2013 [127] Al-Dahash et al., 2017[121]	Other Risks
	Evacuation Plan	Qiu et al., 2014[128], Alazawi et al., 2011[129] Pidd et al., 1996[106]	
	Risk Assessment\Simulation	Aye et al., 2015[130], Zlateva et al., 2013 [123]	
Recovery	Model	Zlateva et al., 2013[123]	Flood, Droughts
	GIS and RS	Jeyaseelan, 2003[84]	Disaster Risks

for preparation, mitigation and management of the disasters. Manfré et al. [110] highlighted the importance of remote-sensing, GIS and GNSS for effective NDM through the establishment of spatial data infrastructure and participation of organization and government to facilitate proper exchange of information. Bessis et al. [133] explained the visionary opportunity in integrating various emerging paradigms including grid, Cloud, pervasive and situated computing for a collective intelligence model for effective disaster management. Laituri and Kodrich [109] introduced the 'people as sensors' concepts using online disaster response community for effective and quick circulation of information using blogs and pictures for better response during every phases of natural disasters.

A comparative summary of related work based on different aspects of disaster management is shown in Table 2.4.

2.4.3 Cloud Infrastructure

The related works are categorized under different categories based on how they have used Cloud environment to address different aspects of the disasters as follow.

2.4.3.1 Computational Application

Ensemble Simulations Some Geospatial and hazard models require a large number of simulations to be run to derive statistical metrics rather than a single deterministic result. This approach is often used when inputs into models are subject to uncertainty and can only be expressed as probabilistic distributions rather than fixed quantities. Examples include the amount of rainfall over a particular area for flood models, or weather conditions in wildfire models. By sampling from a probabilistic distribution and running an ensemble of simulations the results can be combined through a reduction step into a probabilistic output, for example for risk metrics such as probability of flooding or wildfire impact. The Cloud environment is well-suited to support compute-intensive ensembles of hundreds to thousands of simulations. However, few studies have used Cloud infrastructure to run ensembles for predicting the outspread of various disasters. Garg et al. [20] examined the possibility of using Cloud Computing for ensemble run of Geospatial Science models by developing SparkCloud for the wildfire prediction software Spark. Huang et al. [21] verified the readiness of Cloud infrastructure for ensemble run

of complex dust forecasting model by deploying the parallel mode of dust model over Amazon EC2 foundation with reduced costs as compared to local resources. Li et al. [22] developed an MaaS that conducted an ensemble run in parallel with single requests from the users. All the required data for the ensemble run are uploaded by the users using the web-interface. A cyberinfrastructure based geographic information system was developed by Behzad et al. [23] that was able to support ensemble run of groundwater system modeling over the Cloud environment provided by Microsoft Windows Azure Cloud Platform.

Simulation/Modeling Various studies that have implemented models within the Cloud environments to simulate different aspects of disaster management. The smart evacuation system proposed by Qiu et al. [128] performed various modeling of evacuation plan, threats and cities over the Cloud infrastructure. Alazawi et al. [129] performed modeling of impacts of disasters on the traffic flow of the city over the Clouds based on the data collected by multiple sources for better disaster response. Eriksson et al. [117] developed a simulator over the Cloud environment of Amazon EC2 to understand the process of outbreak of pandemic influenza at a particular place. Kalabokidis et al. [71] also simulated the spatiotemporal spread and intensity of a forest fire using FARSITE [5]. Pajorová and Hluchý [134] developed a platform for HPC over the Cloud environment for complex Earth and astrophysics simulations. Ji et al. [135] used Clouds to implement a Geospatial workflow application based on the weights of Evidence Method Metallogenic Prediction for mineral prediction with improved execution time and scalability. Vöckler et al. [136] developed an application to process the astronomical data released by Kepler project across the multiple Clouds of FutureGrid, NERSC's Magellan Cloud and Amazon EC2 using the Pegasus Workflow Management System and generate computationally complex periodograms of the data.

Geospatial/Data Analysis Cloud Computing has been used extensively in Goespatial processing for calculating various indexes, spatial and statistical processing and data analysis for better decision support. The computational task of calculating standardized precipitation index, drought index and vegetation index was carried out by Yu et al. [83] within the Cloud environment. Zou [118] provided a Cloud solution of MapReduce for data analysis of massive RS data including the preprocessing such as

Table 2.5: Categorization based on different computation applications

Computation Applications	Related Works	Purpose/Aspects	Deployment Models/ Cloud Providers
Ensemble Simulations	Behzad et al., 2011[23], Garg et al., 2018[20] Li et al., 2017[22], Huang et al., 2013[21]	Wildfire, Dust Storm, Ground Water System, Geospatial Models	Public Clouds, Nectar Cloud, Azure Cloud Platform Amazon EC2
Simulation/ Modeling	Alazawi et al., 2011[129], Eriksson et al., 2011[117] Kalabokidis et al., 2013[71], Qiu et al., 2014[128] Pajorová and Hluchý, 2011[134], Ji et al., 2012[135], Vöckler et al., 2011[136]	Earth Simulations, Evacuation Disaster Impacts & Outbreak, Mineral Prediction Astronomical Applications	Public Clouds Private CLouds Amazon EC2
Geospatial/ Data Analysis	Yu et al., 2018[83], Di Martino et al., 2011[82] Pajorová and Hluchý, 2011[134], Zou, 2017[118] Wan et al., 2014[85], Montgomery et al., 2010[70] Golpayegani and Halem, 2009[137] , Fujioka et al., 2012[138], De Luca et al., 2017 [139], Wan et al., 2018[140], Li et al., 2011[69] Wang et al., 2010[141], Huang et al., 2017[54] Al-Dahash et al., 2017[121], Böhm et al., 2011[119] Zlateva et al., 2013[123], Huntington et al., 2017[89] Liu et al., 2012[67], Kalabokidis et al., 2013[71]	Indexes Calculations, Spatial Interpolation, Statistical Processing, Data Classification, Climatological Calculations, Risk Identification	Public Clouds Private Clouds Amazon EC2

radiometric correction, geometric correction, mosaic and fusion and information extraction processes such as classification, transformation and index calculation. Zlateva et al. [123] used the Cloud infrastructure to perform risk assessment of the natural disasters using a joint application of fuzzy logic models and an actuarial model. Various biodiversity indices at different resolutions were calculated using the marine life data in the system developed by Fujioka et al. [138]. VirtualFire [71] computed fire ignition probability for identification of high-risk areas.

Wang et al. [141] used the Cloud infrastructure to perform high performance and distributed spatial interpolation of hugely massive spatiotemporal data sets that included climate data, census survey data and Remote Sensing images. The study carried out by Golpayegani and Halem [137] developed a high end compute clusters over the Cloud infrastructure with a distributed file system and MapReduce framework integrated into the cluster for speedy large-scaled processing of largely massive Remote Sensing datasets. A number of Geospatial analysis and statistical processing was facilitated in the work done by Huang et al. [54] who integrated diffferent Geospatial models into their system. The HCC platform was capable of supporting ensemble runs of Geospatial models which was validated by run of dust storm forecast model over the primary Cloud infrastructure of Amazon EC2 Clouds. Al-Dahash et al. [121] maintained a separate layer in the Cloud environment to perform various processing and analysis over the data collected in the database to draw significant conclusions for better disaster management. A global flood infrastructure built by Wan et al. [85] used a Google chart API for creating analytic chats for statistical analysis of flood events. Moreover, the system classified the data into different levels based on severity and fatalities of the flood. The environmental monitoring developed by Montgomery et al. [70] performed various processing of Geospatial data for prediction of the changes in various environmental resources so as to ensure proper adaptation for sustainability. Li et al. [69] used Cloud infrastructure to perform various complex Geospatial computing tasks such as FCD query, FCD map matching and speed computation for roadlinks for urban traffic monitoring. The power grid storm disaster monitoring system developed by Liu et al. [67] used the features of Cloud Computing to solve the processing difficulties associated with largely massive geomagnetic data, satellite data and other earth space observation data.

A geovisualanalytics system proposed by Böhm et al. [119] in the public health sector implemented an innovative geo-business intelligence methods and procedures of public

heath over the Cloud for better decision support in planning and analysis processes. Climatological calculations and statistical analyses were carried out on the climate and observation data in the climate engine [89]. De Luca et al. [139] used Cloud infrastructure to form a processing chain of differential Synthetic Aperture Radar (SAR) interfermetry (DInSar) Parallel Small Base-line Subset (P-SBAS) for unsupervised processing of large volumes of SAR data. The processing of large volume RS data was dealt with using Virtual Processing System for RS (VS-RS) over the Cloud in pipsCloud [140].

A comparative analysis of use of Cloud Computing for different computational applications is given in Table 2.5.

2.4.3.2 Visualization

A wide range of visualization techniques and functionalities have been offered over Cloud infrastructure in conjunction with web technologies for better interpretation of spatial results obtained after computational processes. Interactive mapping tools, advanced animations and 3D visualization have been integrated along with Cloud technologies to provide elaborated and classified information about disasters to take better informed decisions. Moreover, a visual interface hosted over the Cloud provides users of the system the ability to customize and keep track of any processes in operation. Categories of functionalities and support through the use of different visualization methods are further described as follow:

Interactive Mapping Services The mapping tools used in various studies enabled better understanding of the results obtained after analysis and simulation. Researchers have extensively used a wide range of interactive mapping tools to better visualize the processes that govern different disasters and understand the possible damages caused by those disasters. The studies have made use of existing mapping tools, such as the Google Maps API, while some studies have integrated mapping services of servers such as TeraGrid, and GeoServer. Some of the studies offered real-time mapping services for various Geospatial processes while some offered advanced capabilities by integrating mapping tools into the system after completion of simulation runs or analyses. The related works are categorized into two categories based on their real-time mapping capabilities as follow:

- *Real-Time Mapping Services.* The Sentinel Hotspots Australia and AUSCORS Australia, 2018 [88] built by GeoScience Australia used Cloud infrastructure to develop an interactive visualization interface for bushfire events and 1 Hz data streaming from GNSS stations respectively throughout Australia, Antarctica and the Pacific. Wan et al. [85] used a public Cloud-based flood cyber-infrastructure to develop a tool called CyberFlood that could collect, organize and manage global flood data for providing real-time location-based eventful visualization to authorities and the public. The visualization enabled by Cloud Computing included various statistical and graphical capabilities. They used the Google Map API and interactive combination of color codes to represent data under different categories for concise and useful information related to the floods. Montgomery et al. [70] integrated a collaborative visualization along with mapping tools in the user interface of the system for visualization of data for monitoring. The traffic surveillance system described by Li et al. [69] evaluated the utility of Cloud Computing for visualization of urban traffic data obtained after computing tasks namely Floating Car data (FCD) query, FCD map matching and speed computation for road links using different interactive map tools. The system described by Li et al. [120] used a web-based interface powered by mapping services to enable the users to access information, collaborate and communicate efficiently. The work done by Zou [118] proposed a web platform for disaster monitoring where users could visualize the data produced over the portal using maps or download the files in KML or vector file format.

- *Non Real-Time Mapping Services.* Ji et al. [135] incorporated interactive mineral maps for visualization of results obtained after relevant data analysis. Yu et al. [83] used the Cloud environment for visualization of processed data with mapping tools integrated into the system for better understanding of the results. The web-interface in the HCC developed by Huang et al. [54] was capable of displaying the results obtained after spatial analysis or run of Geospatial models using maps. Qiu et al. [128] used maps of a city integrated with results obtained from evacuation model to visualize the evacuation plans. Alazawi et al. [129] made use of interactive maps to visualize the optimized plan for smooth evacuation and better flow of traffic during the events of disasters. Fujioka et al. [138] used a mapping engine powered by GeoServer 2.1 to interactively visualize the search results obtained

from the marine life census system maintained over the Clouds. Eriksson et al. [117] facilitated the visualization of simulation results through a simple easy-to-use GUI interface enabled with mapping service in the Cloud-based simulator built for pandemic influenza. Böhm et al. [119] facilitated the visualization of important health data over the Clouds in a scalable manner through the use of JavaScript and HTML along with XML to form a map widget for better understanding. The service layer in Service Oriented Architecture (SOA) proposed by Rossi et al. [95] used a website to geographically and interactively visualize all data handled by service layer in a mapping layout. The visualized data could also be downloaded in different user formats as desired by the users in the system. Li et al. [22] used various interactive visualization tools to display the output of ensemble run of the Geospatial models using different maps in the Cloud environment without downloading the output files. VirtualFire [71] used Bing Map Services and other APIs as web services for interactive visualization of fire spread and weather data. De Luca et al. [139] used advanced mapping services for visual representation of geographical regular grid and deformation velocity maps generated by P-SBAS processing. Wang et al. [140] used interactive map services in their web-based interface for visualization of the data obtained after user queries. The Cloud environment was used to visualize the disaster data sets and information collected over time by Al-Dahash et al. [121] for better decision support.

Animations and Advanced Visualization Somestudies have utilized the capabilities of Cloud Computing to build advanced animations, rendering of data in 3D and facilited advanced visualization in augmented reality. The HCC system developed by Huang et al. [54] can interactively visualize results after spatial analysis using animations. GI-Solve [141] integrated visualization services supported by TeraGrid into a system for spatial visualization of data through self-guided user interfaces. Qiu et al. [128] used advanced visualization features over the Cloud to display results from evacuation and threat models in 3D scenarios. Montgomery et al. [70] integrated a wide range of communication media including forums, news, blogs and videoconferencing over a custom web page portal for effective collaboration. The study also integrated InteleView for displaying high resolution maps on global scale, in 3D and in real-time. The climate engine [89] allowed users to perform on-demand mapping and time series visualization

over the Cloud. Vöckler et al. [136] integrated the features of FutureGrid into their system to facilitate the visualization of astronomical data released by NASA. Miska and Kuwahara [142] made use of WebDAV protocol in their Cloud-based system to allow interactivity in a web-based interface and allow users to create, change and move the documents on a remote server. Moreover, the system used OpenSIM to facilitate the visualization of a 3D environment within the web-based interface. Visualization Tool (VT) with advanced displaying capabilities was developed by Pajorová and Hluchý [134] as an e-Science gateway over the Cloud for visualization of simulations related to the Earth and astrophysics. The architecture proposed by Di Martino et al. [82] also integrated augmentation module that received global navigation satellite system (GNSS) data and computed augmented and validated GNSS position for advanced visualization features.

Customization The visual interface for different Cloud-based systems can be customized based on user preferences and scenarios for the operation of the system. The visual interface allows users to make changes to parameters, timelines and models that are needed to produce relevant results in advanced visual forms. Elements within the user interface can be customized in order to graphically represent various results. Categories of different customisation types for aspects of the user interface are given below:

- *Operation Customization.* The system devised by Li et al. [22] gave users the option of configuring a particular job before initiating the entire run of a model. Users were provided with the ability to customize analytic steps in the climate engine Huntington et al. [89] to produce map and time series results including product types, datasets, variables, calculations and statistics. The stakeholders could input different desired risk scenarios into the platform developed by Aye et al. [130] for possible mitigation measures for disaster management. Huang et al. [143] allowed users to configure the system in any way after an authentication step. Users could customize the monitoring operations through the user interface provided by the system in the work proposed by Montgomery et al. [70]. Shen et al. [115] developed a system where users could find health services for their conditions that was customisable according to their needs through a user interface. Böhm et al. [119] developed a Business Intelligence (BI)-GIS system which offered geovisualanalytic solutions through a customisable user interface. The simulation

execution of pandemic influenza could be controlled using a front end interface in Cloud-based architecture proposed by Eriksson et al. [117]. In the Hybrid Cloud Computing (HCC) platform developed by Huang et al. [54], users could customize and choose any desired models from a group of models that were integrated into the platform.

- *Parameters Customization.* SparkCloud [20] allowed users to customize wildfire simulations based on igntion locations and timelines. The users of VirtualFire [71] could change the inputs to the system to visualize desired sets of output in the web-based services. The system developed by Ramachandran et al. [144] for distribution of NASA collected datasets allowed users to pick sets of required datasets. Users of the system developed by Yu et al. [83] could customize the location parameters for calculation of different index and visualization of the result data. The users of monitoring system in the work described by Li et al. [69] could change the traffic monitoring based on different parameters.Fujioka et al. [138] developed a marine life census system that could customize a search operation using a number of different parameters for well-refined search results. The cyberGIS framework developed by Wang et al. [141] allowed users to customize various parameters to run the system in desired way for data processing or visualization. The ground water system developed by Behzad et al. [23] allowed user to change ensembles parameters of the model that simulated the flow of ground water.

- *Result Customization.* Many systems have facilitated the customization of visual forms for the representation of the results [89], [82], [83] and [143]. Wan et al. [85] allowed users to select a range of years and causes of the flood in the web-interface hosted by an Apache web server to visualize the customized results. The system devised by Wang et al. [141] was capable of customizing the visualization of analyzed data in a number of ways.

Job Status The system developed by Huang et al. [54] facilitated the users to keep track of the status of the jobs through a web-based interface. The web-based interface in sparkCloud [20] allowed user to monitor the status of the job requested by the users. The web-interface developed by Li et al. [22] could show the status of each jobs that were being processed for the ensemble run of Geospatial models over the Cloud.

Table 2.6: Categorization based on different visualization tools and functionalities

Visualization Tools and Functionalities	Related Works	Tools/Aspect
Mapping Tools	Australia, 2018[88], Ji et al., 2012[135] , Yu et al., 2018[83] Huang et al., 2017[54], Wan et al., 2014[85] Qiu et al., 2014[128], Alazawi et al., 2011[129] Eriksson et al., 2011[117], Böhm et al., 2011[119] Huntington et al., 2017[89], Li et al., 2017[22] Kalabokidis et al., 2013[71], Wang et al., 2018[140] , De Luca et al., 2017[139] Li et al., 2011[69], Al-Dahash et al., 2017[121] Montgomery et al., 2010[70], Fujioka et al., 2012[138] Di Martino et al., 2011[82], Rossi et al., 2017[95]	Google Map API, Color Combinations, GeoServer Map Engine, JavaScript, HTML with XML, Bing Maps
Animations and Advanced Tools	Huang et al., 2017[54], Miska and Kuwahara, 2010[142] Di Martino et al., 2011[82], Rossi et al., 2017[95] , Montgomery et al., 2010[70], Qiu et al., 2014[128] Pajorová and Hluchý, 2011[134]	3D Scenarios, InteleView, TeraGrid Augmentation Module FutureGrid, WebDAV, OpenSIM
Customization	Ramachandran et al., 2017[144], Wan et al., 2014[85] , Kalabokidis et al., 2013[71], Liu et al., 2012[67] , Li et al., 2017[22] Huntington et al., 2017[89], Garg et al., 2018[20] Wang et al., 2010[141], Yu et al., 2018[83] , Li et al., 2011[69] Behzad et al., 2011[23], Di Martino et al., 2011[82] Montgomery et al., 2010[70], Shen et al., 2012[115] Eriksson et al., 2011[117], Huang et al., 2017[54] Huang et al., 2010[143], Fujioka et al., 2012[138] Böhm et al., 2011[119]	Preferences, Parameters, Risk Scenarios, Deadlines Models,
Job Status	Li et al., 2017[22], Garg et al., 2018[20] Huang et al., 2017[54]	Status Bar
Decision Support about Tools	Chavan et al., 2016[145], Li et al., 2011[120] Shen et al., 2012[115], Behzad et al., 2011[23] Habiba and Akhter, 2013[132]	Smart phones, Messages, Web portals, Monitoring

Decision Support Various work has facilitated better decision support at different stages of natural disasters by developing systems delivered over Cloud infrastructure using a diverse range of technologies. The Social Media Alert and Response to Threats to Citizens (SMART-C) developed by Adam et al. [131] focused on developing participatory sensing capabilities for better decision support throughout the life-cycle of a disaster using multiple devices such as smartphones and modalities such as messages, web portals, tweets and blogs. The system architecture proposed by Chavan et al. [145] facilitated the use of Graphical Processing Unit (GPU) for displaying the results of spatial queries by the users. Shen et al. [115] used Clouds for scalable, customizable and robust visualization of health services data obtained after various data processing and clustering steps. The monitoring system developed by Liu et al. [67] and delivered through a wide range of devices allowed user access through a web browser, where desired services could be selected. The outputs of the ensemble run of a ground water model could be saved in Blob Storage after compression and downloaded by the user through a web-based interface in the system developed by Behzad et al. [23]. Habiba and Akhter [132] developed web portals for visualization of data from different modules in a framework proposed for effective disaster management. CUMULUS [144] developed Cumulus-API for a protected GUI allowing users to gain insight into operations taking place and management of the platform.

The categorization of visualization based on tools and functionalities is shown in Table 2.6.

2.4.3.3 Storage

The almost unlimited capacity of the Cloud infrastructure has been well used in different studies to store large and diverse spatial data sets in Structured and Unstructured forms. Some studies have not clearly defined the form in which the data sets were stored, but the Cloud environment was utilized to store large data sets.

Structured Databases Most work to date has dominantly used SQL, PostgreSQL and PostGIS to store the spatial data in a structured form. The system developed by Chavan et al. [145] used SQL to store spatial data in the Clouds. Montgomery et al. [70] stored various Geospatial data sets over the Cloud using traditional SQL

design to monitor water supply, weather, ocean to predict and adapt to their changes for sustainable development of the environment. Fujioka et al. [138] stored more than 31.3 million observations of marine life data as the Marine Life Census within the Cloud using PostgreSQL and PostGIS, with free access to the users through a Geospatial portal. Rossi et al. [95] used Azure SQL for storing all the user textual information over the Cloud infrastructure. Huang et al. [143] used Postgresql with PostGIS to support spatial datasets for deployment and maintenance of GEOSS Clearinghouse on an Amazon EC2 platform.

Unstructured Databases The unstructured forms such as NoSQL along with Graph databases, Hadoop Distributed File System (HDFS), Blob services, big-table and geo-databases were widely used to store and analyze the spatial data. Grolinger et al. [124] used Graph databases to represent and store the data using graphical structures with edges, nodes and properties. Huang et al. [54] implemented the concepts of distributed file-system, relational database and NoSQL database over the Cloud infrastructure for holding massively large Geospatial data for different models in the HCC platform. Zou [118] stored massive sets of satellite data over the Cloud in a more distributed approach using HDFS. Wan et al. [85] created the Flood Data Archive within the Cloud using Google Fusion table, containing all flood related data from 1998 to 2008. Grolinger et al. [146] proposed Knowledge as a Service (KaaS) for disaster Cloud data management for facilitating storage of massive datasets related to disasters in relational NoSQL databases. The framework developed by Jiang et al. [116] used HDFS to store massive sets of Geo-data related to forest pest control. Qiu et al. [128] stored the data collected from different sensors installed in disaster prone areas in the data centers for further processing. Puthal et al. [90] described the storage of the information over the Clouds citing the data-intensiveness nature of the collected data for batch processing in a store-and-process fashion. Rossi et al. [95] used Azure Blob service for storing all the user photos and logos over the Cloud infrastructure. The adaptation of HPGFS along with Hilbert-R+ tree based data indexing in NoSQL database over the Cloud foundation handled the vast amount of unstructured RS data in pipsCloud [140]. Schnase et al. [68] used the framework of Integrated Rule-Oriented Data System (iRODS) to store the disparate data over a distributed architecture. The fire data in VirtualFire [71] was stored over the Clouds using geo-database.The community-based Cloud developed by

Table 2.7: Categorization based on different structure of Cloud storage

Form	Related Works	Tools
Structured Form	Chavan et al., 2016[145], Montgomery et al., 2010[70] Rossi et al., 2017[95], Huang et al., 2010[143] Fujioka et al., 2012[138]	SQL, PostgreSQL PostGIS
Unstructured Form	Grolinger et al., 2015[124], Huang et al., 2017[54] , Zou, 2017[118] Wan et al., 2014[85], Grolinger et al., 2013[146] , Jiang et al., 2010[116] Puthal et al., 2016[90], Rossi et al., 2017[95] , Wang et al., 2018[140] Li et al., 2011[69], Schnase et al., 2011[68] , Li et al., 2011[120], Shen et al., 2012[115], Kalabokidis et al., 2013[71]	Graph databases, NoSQL, HDFS, Azure Blob Service, big-table, geo-database Google Fusion Table
No Description	Australia, 2018[88], Ji et al., 2012[135] , Ramachandran et al., 2017[144] Yu et al., 2018[83], Wang et al., 2010[141] , Qiu et al., 2014[128] Miska and Kuwahara, 2010[142], Habiba and Akhter, 2013[132] Böhm et al., 2011[119], Huntington et al., 2017[89] , Behzad et al., 2011[23] Al-Dahash et al., 2017[121], Klauck et al., 2011[126] , Li et al., 2017[22] Adam et al., 2012[131], Liu et al., 2012[67], Alazawi et al., 2011[129]	

Li et al. [120] maintained a virtual community database for physical and human resources information and social media database using semantic dimensions for real-time emergency situation through social medias over the Clouds for emergency management. Shen et al. [115] utilized the Cloud infrastructure to develop an effectively managed data archive systems in the form of historical diagnosis database and knowledge base to record medical resources in public health area for developing alternative practices.

No Detailed Information GeoScience Australia used the Cloud storage provided by AWS to store the massive data related to bushfire events and GNSS observation data in different systems of Sentinel Hotspots and AUSCORS [88]. The system developed by Ji et al. [135] stored the Geospatial data using the Cloud in a native but complex Geospatial type. The data archive developed by Ramachandran et al. [144] optimized the files based on the input configuration and distribution requirements before storing them in the Clouds. Climate precipitation data was stored in the Clouds during the study carried out by Yu et al. [83]. The cyberGIS framework developed by Wang et al. [141] integrated the data storage and management capabilities of middleware workflows into three core data services to handle the massive spatiotemporal data. The SMART-C system devised by Adam et al. [131] stored massive data sets within a Cloud database to keep track of demography, weather, traffic, hospitals, schools and so on for anticipating possible disasters. The framework proposed by Habiba and Akhter citehabiba2013Cloud used Cloud infrastructure for data record services to facilitate different functionalities after required processing and analysis. The intelligent disaster management system developed by Alazawi et al. [129] used the Cloud to store data collected from multiple sources and locations including the place of an event for better decision support. A high volume of public health data was stored in the Cloud environment by Böhm et al. [119] to form a business intelligence widget. Large datasets of climate and satellite Earth observations were saved over the Cloud environment for the climate engine [89]. Behzad et al. [23] integrated the Geospatial middleware in the Cloud for storing massive datasets related to ground water flows and maintaining the datasets in an archive over the Cloud infrastructure. Al-Dahash et al. [121] developed database over the Clouds to store all the information about the terrorism collected from multiple sources for further processing and analyses. Klauck et al. [126] stored the information over the Cloud to enable the collaborative work and reduce the acquisition and maintenance costs. The

MaaS framework proposed by Li et al. [22] used data servers over the Clouds to store the large number of output data produced after ensemble run of various Geospatial models. Miska and Kuwahara [142] developed an innovative idea to use the features of Cloud Computing to start project management framework by maintaining International Traffic Database project over the Clouds with new possibility of handling the entire project publishing and communication at a place. Liu et al. [67] used the Cloud infrastructures to store massively large amount of geomaganetic data, satellite data and other earth space observation data for power grid storm disaster monitoring.

A comparative analysis of the related works on Cloud storage on the basis of structure is given in Table 2.7.

2.4.3.4 Data Management

Researchers have widely used existing Cloud services to effectively handle and manage the data in their systems while some have developed their own data management framework to better suit their purposes. Ji et al. [135] deployed Hadoop for effectively handling and processing the Geospatial data in their Geospatial workflow application maintained over a Cloud environment. Chavan et al. [145] used basic spatial operators, computational geometry operators and Open Geospatial Consortium compliant operators for handling the user queries under an optimized plan given by Query Optimizer in the system that worked on based on a cost model. Grolinger et al. [124] used proprietary graph query language called Cypher to query the data stored in graph databases maintained within a Cloud environment. Ramachandran et al. [144] developed CUMULUS as a native data management system which generated granule-level metadata with collection-level metadata stored in the catalog pointing to the storage locations maintained over AWS Cloud environment. The system used Amazon Lambda, EC2, EC3, S3 and SQS services for data processing. Wang et al. [141] proposed a distributed data management services for storage, where the service kept track of metadata about spatial and computational features of every data set and results were fetched based on the requirements of the queries. The effective handling of the flood data was ensured using Google Fusion Table where additional location information was presented as MultiGeometry using Keyhole Markup Language (KML). The queries were similar to SQL queries and data was updated in the table only after satisfying some predefined criteria. The

Knowledge as a Service (KaaS) model proposed by Grolinger et al. [146] used a series of steps such as text extraction from images, file metadata separation, pattern processing and tagging for effective data management. The database for various environmental aspects data maintained by Montgomery et al. [70] were automatically connected to external databases, internet sites and other different sources using standard protocols of SQL, HTTP and FTP. Jiang et al. [116] used HBase and MapReduce to store Attribute Data and process data, respectively, in their framework. The smart evacuation system built by Qiu et al. [128] used MapReduce functions to perform required data analysis over the data collected by different sensors installed all over the cities. Puthal et al. [90] explained the methods of batch processing and data stream processing for analysis of the data collected from known and unknown sources where they focused on data stream processing of the data over the Clouds for real-time event detection. The solution proposed by Böhm et al. [119] consisted of different data layers for various visualization methods such as clustering, heat-map and polygons. Search queries were made to be based on attribute rather than spatial ones in the marine life data system developed by Fujioka et al. [138]. The data servers maintained over the Clouds in Maas framework proposed by Li et al. [22] handled metadata management for all the output data produced after ensemble run of the models. Rossi et al. [95] used .NET Entity framework as Object Relation Mapper and REST architecture for querying the stored data. Miska and Kuwahara [142] focused on storage of data with meta information for better handling and management with better understanding of the data.

Kalabokidis et al. [71] used ArcGIS server to effectively handle the stored data in VirtualFire. Li et al. [120] used a distributed hash tables (DHTs) to locate desirable data and resolve any queries efficiently in a community based emergency management system. PipsCloud [140] used HBase as metadata depository for handling the metadata management and Google File System (GFS) for RS data management. Li et al. [69] used Cloud Computing technologies such as Bigtable and MapReduce along with spatial indexing to query high volume of FCD over the Clouds for effective monitoring. Schnase et al. [68] facilitated the support for metadata to identify the properties of stored object for easier management of the data archive maintained over the Clouds. Behzad et al. [23] used Geospatial middleware for effective data management in their work for ground water simulation.

2.4.4 Control Mechanism

Researchers have used different frameworks and techniques to enable various control functions within their systems to efficiently use the Cloud environment. Chavan et al. [145] proposed the technique of space-filing curves for load balancing across all the cores of GPU to enhance the performance of the system by utilizing all the processing units. The study done by Garg et al. [20] incorporated deadline-based execution, effective load balancing, on-demand execution, fault tolerance and scalability in the system so as to be able to handle multiple requests from the concurrent users. The Cloud-based simulator developed by Eriksson et al. [117] used Condor framework for job distribution and management of EC2 and local resources. Huang et al. [143] used Amazon SQS to handle the queue of the users in a reliable and scalable manner when user requests are traveling between computers. PipsCloud used xCAT to extend the capabilities of OpenStack for supporting resources provisioning. Vöckler et al. [136] constructed a virtual Condor pool to handle the resource provisioning in the system proposed for running the application on astronomical data over the Clouds. The system used Pegasus, DAGMan and Condor for failure recovery mechanisms.

2.5 Future Directions

Cloud Computing has revolutionized the way computing is carried out in many fields, with its unprecedented benefits in scalability, computational resources and vast potential storage. Based on this literature survey, NDM is an excellent candidate for deployment on Cloud systems, but there are still factors within this discipline that make implementation on the Cloud non-trivial. The work carried out so far has shown the possibilities and benefits of integrating Cloud technologies with Disaster Management, such as the ability to offer end services to agencies and authorities, and even general public during emergency and natural disasters. This work can be extended to to fully utilize the capabilities of Cloud Computing and address the various challenges in the field. Liang et al. [147] called for the development of Cloud Computing applications for disaster monitoring, forecasting and warning to mitigate potential losses caused by the disasters. Bessis et al. [133] proposed a roadmap highlighting the possible use of new and emerging

technologies to enable collective computational intelligence in managing disaster situations. These examples illustrate that the adoption of Cloud technologies in Disaster Management may significantly help in minimizing the impacts and losses from natural disasters.

As such, the following section discusses and analyzes potential research areas for the integration of Cloud Computing to NDM and highlights future directions where research can be focused for more effective disaster management.

2.5.1 Effective Handling of Ensemble Simulations

Geospatial processes and Natural Hazard models may require ensemble simulations to calculate probabilistic outputs based on uncertain input conditions. This involves running a set of simulations, where each simulation is usually based on a complex physical model. Computing a set of simulations requires a correspondingly larger computation time than a single simulation and, depending upon the complexity of the model and number of uncertain parameters, such ensembles may take anywhere from several hours to days to complete on servers or local workstations. Ensemble runs of the Geospatial model in different instances introduces further complications due to the necessity for ordering and synchronisation of results. Every output from every single run of simulation must be carefully collected, stored and processed for further reduction and statistical analysis steps.

A recent study carried out by De Luca et al. [139] used the Cloud Computing environment to perform unsupervised processing of large SAR data volumes on a large number of computing nodes in an Amazon Web Service environment. The study suggests Cloud Computing may be a possible alternative to HPC scheme for ensemble simulations of Geospatial processes. However, there is a current need to develop an optimized mechanism for distributed modes of operation for ensemble simulations in the Cloud. Neither this nor any other study have defined or considered any computing schema that considers time-sensitivity, resource utilization and user-defined requirements when it comes to implementation of an NDM model.

A future development pathway would be for an optimized mechanism for ensemble simulations ensuring maximum resource utilization within a user-defined cost or time envelope. This could involve the development of a robust and optimized mechanism for independent operation of simulations over a number of instances making sure any idle instance in the configuration could take over the other sets of simulations from other instances. Development of this mechanism would involve the use of different optimization techniques so as to ensure all the resources during the run are used in an effective and efficient way.

The concept of a centralized storage system would also be useful for ensemble simulations, but this must be a well-defined mechanism that pushes and pulls the results on demand from the storage system. Aggregation of results can be more challenging if these have to be filtered for an optimized visualization. The centralized storage system would have to deal with large amounts of data and may require an effective method to filter out any irrelevant results. A central storage system can also be important in caching any replicated ensemble runs of the model for the same set of parameters or inputs. Furthermore, the storage system could be coupled together with an effective checking algorithm as a pre-processing filter for unnecessary simulations in the ensemble, saving valuable computational time and resources.

2.5.2 Integrated Natural Hazard Models

As previously discussed, a natural hazard represents a significant risk to the environment, people and infrastructure. Any relevant information prior to and after the occurrence of the natural hazard can be crucial in minimizing the impact of the disaster. Such information can include historical information about the occurrence of a particular disaster in a specific area, prediction results for any disasters, information about the extent of impacts of the hazard for a particular location, disaster response management and damage assessment. Various studies have been carried out to provide such functionality in a Cloud environment but, so far, in an isolated manner. Furthermore, little has been advocated and addressed in the need for a complete disaster management system that is able to handle the spectrum of needs from preventive measures to post-disaster damage assessment.

Future Natural Hazard Management systems can leverage the capabilities offered by Cloud Computing. Such a system could use a data archive within the Cloud environment to provide instantaneous access to historical occurrences of disasters at a particular location of interest to better inform authorities or the general public for effective planning in case of an actual events. During an actual event, a disaster model can offer an end service to predict factors such as evacuation or impact times. This could allow individuals and agencies can take action within an available time window. Moreover, effective planning strategies could be developed using risk metrics based on ensemble runs of a natural hazard model. Furthermore, a disaster model could deliver a visualization platform on top of the Cloud environment to keep track of operational resources to effectively and optimally mobilize these during an actual event. Such a management system could also incorporate crowd-sourced real-time information of the disaster to form a clearer operational picture of the unfolding events.

Such a system could be based on an efficient group modules consisting of Geospatial processing and natural hazard modeling elements provided in a complete system. This complete system could be deployed over a Cloud environment and a generic framework for all natural hazards. Future development on such systems should be able to ensure seamless end services with clear and well-defined results to its users. Under this complete system, future research work could be directed to facilitate interoperability of different data storage techniques and more advanced capabilities enabled by evolving IoT applications. Some potential areas to focus on for such a system would be:

2.5.2.1 Handling the Challenge of Big Data within Cloud

For disaster management, data from a wide range of sources are to be considered. This includes the real-time spatiotemporal data from location services, social media, volunteer geographic information, satellites and UAVs [148]. The data from sensor web and IoT, airborne and terrestrial Light Detection and Ranging (LiDAR), simulation, spatial data, crowdsourcing and call data records are shown to be important for disaster management in [149]. For an effective hazard model, a number of diverse data sets may have to be repeatedly processed and analyzed by different modules to derive useful results. Therefore, the lack of interoperability between different data types can significantly hinder performance and the effectiveness of any system if not properly addressed [150].

Given the large amount and sources of data, manual analysis and interpretation of such integrated data are to be replaced by sophisticated and advanced automatic mechanisms to make the data analysis more efficient and effective [149]. Different machine learning techniques (text classification in [151][152], Neural Networks in [153],[154], [155]) have been used to derive more accurate results for disaster response and assessment for different disasters. Moreover, studies [90],[85], [156] have emphasized in setting up big data cyberinfrastructure for disasters that can help in efficient data collection, information extraction, distribution and visualization for effective disaster management. Future works should look into challenges created by a cyberinfrastructure in relation to efficient data management, more intuitive data visualization and low latency during data transfer. Various studies have made use of a wide range of data storage techniques, but there are no clearly defined mechanisms that explain how heterogeneity in data storage can be effectively dealt with. Due to this, future work should focus on effective mechanisms to support the interoperability between heterogeneous data sets within the Cloud environment. There have to be efficient analytical methods that can integrate the crowdsourced data with Geospatial data for better disaster situation awareness and prediction.

2.5.2.2 Handling Inaccessibility of Cloud Services during the Disasters

During the actual occurrence of the disasters, as Cloud services may be inaccessible due to communication and power outages, fog/edge computing can play a significant role in the optimal mobilization of the emergency response teams. As highlighted in the work [157], the rescue personnel engaged in search and rescue operation can be continuously tracked using end devices like phones and sensors. This tracking enabled by fog computing can be used to create a real-time density map of people in the affected region that can help and guide the response teams. In edge computing, a varying degree of computational powers is available to end devices like cell phones, tablets, cameras and sensors. Thus, less compute intensive processing can be directed towards these end devices rather than the traditional cloud infrastructure to significantly decrease the latency [158]. Similarly, in crowdsourced data analytics, more sensitive data can be processed closer to where they are generated while other data can be sent to the Cloud for further historical analysis and storage [87]. The end devices in fog/edge computing can make the people and the responders situationally aware for better-informed decisions during the emergencies [87]. Thus, there is no doubt that a Cloud-based solution can be used

in conjunction with end-devices of fog/edge computing and IoT networkby distributing the services based on time-sensitivity and compute-intensiveness. Use of IoT sensors could facilitate greater readiness and responsiveness but bring further challenges of *Big Data* within a Cloud-based system. Although there has been significant work carried out in addressing the challenges of handling massive data sets collected by extensive IoT sensor network in general [159],[160],[161], there are no clearly defined capabilities for scalably handling and processing such Geospatial data streams. Furthermore, there are not clearly defined architecture to integrate the evolving paradigms of edge and fog computing within a Cloud-based solution for effective diaster management. Future research could center around better integration of IoT sensor networks, Edge and Fog Computing in Disaster Management for more advanced real-time services.

2.5.3 Addressing the Need for Concurrent Access and Dynamic Configuration

This work has highlighted the need for intensive ensemble simulations for Geospatial processing and natural hazard models, ideally offered as an end service. Examples have been given of migration of some models to a Cloud infrastructure from the traditional use of a local HPC scheme with a set of static configuration. Ensemble simulations for such models are governed by a set of user inputs and a Cloud-based system must be able to update and change an entire set of Geospatial process if any of these inputs are changed without compromising performance. Moreover, it is not just a single user that may be using the model at a given instant of time, and any end service must be able to be concurrently accessed by a multiple users at different locations. There is a significant need for future systems that can effectively handle multiple sets of ensemble simulations with different configurations and concurrently provide results to a wide number of users. Future effective general-purpose systems must address these issues of concurrent access and dynamic configuration for ensemble simulations. This could result in improvements in existing resource pooling, scheduling, queuing and load balancing techniques for the advanced and sophisticated algorithms required to effectively deliver disaster management systems on the Cloud.

2.5.4 Overcoming the Bottleneck of Network Capabilities

Ensemble simulations for disaster management models within the Cloud environment are by nature intensive in concurrent-access, data processing and computation. The processing of huge data sets requires a large amount of data to be transferred between nodes in the Cloud. However, Cloud infrastructure may have limited network capacity that could potentially create bottlenecks in the development of Cloud based natural hazard modeling systems. Given the migration of disaster management models to the Cloud and the growing data intensiveness of such models there may be future performance and scalability issues owing to the large number of interacting services and networks. This requires effective network management services to ensure seamless integration and delivery of data-intensive Geospatial processes based on Cloud Infrastructure. Future work to address this could include automation of specific network functions in the Cloud environment to keep up with networking demands of the Geospatial processes. A possible separation of the control plane from the data forwarding plane in Software Defined Networks (SDN) [162] could be studied to find newer ways to accommodate and adapt to dynamic workload and find an optimized configuration for ensemble simulations. Moreover, future works could focus in exploring and discovering new ways of utilizing networking hardware to realize the full potential of massively distributed Cloud computation.

2.5.5 Risk Analysis for Operational Management

There are uncertainties associated with natural hazard models such as wildfire models and the model performance and the effectiveness of risk management achieved with such models are determined to a large degree by how well such uncertainties are understood and communicated [163]. Deriving accurate risk metrics from these models by quantifying the associated uncertainties can require a significantly high model runs under a wide range of possible scenarios. Running such computationally intensive analyses on a small pool of computers may take longer than the time window available for operational management. Cloud resources can support the computational requirements of such risk analyses but novel mechanisms have to be defined to integrate existing analysis methods into Cloud infrastructure. Sensitivity analysis has been widely studied as one of the most

popular approach to uncertainty quantification and risk analysis [15, 26, 29]. Future research works on enabling rapid risk analysis for operational disaster management could focus on determining ways to apply sensitivity analysis to operational disaster models in a convenient and time efficient manner.

2.6 Summary

In this chapter, we identified the commonality between different natural hazard modeling systems and proposed a generic framework for offering the functionalities of natural hazard models as a service for rapid risk estimation. Moreover, we also identified trends in research and identified future research areas which we believe will be important for this area for newer and more advanced capabilities of disaster management. Next generations of NDM systems should employ novel and intelligent mechanisms to quickly estimate and analyze the possible risks, and identify the high-risk areas of any disaster for operational management. The major problems associated with effectively handling ensemble predictions have been addressed in Chapters 3-5 to ensure rapid risk estimation, risk identification, and risk analysis for effective disaster management.

Chapter 3

An Efficient Framework for Ensemble of Natural Disaster Simulations as a Service

To achieve rapid risk estimation, in this chapter, we propose a system framework that offers ensemble predictions as a service in a convenient and time-efficient manner with optimized costs. The cost is minimized in two phases through efficient distribution of the simulations among the cost-efficient instances and intelligent choice of the instances based on pricing models. We validate the proposed framework using a real Cloud environment with real wildfire ensemble scenarios under different user requirements. Our findings give an edge to the proposed system over the bag-of-task type execution on the Clouds with less cost and better flexibility thereby demonstrating the ability of our Cloud-based framework to support ensemble predictions for rapid risk estimation.

This chapter is derived from the following published work.

KC, U., Garg, S., & Hilton, J. (2020). An efficient framework for ensemble of natural disaster simulations as a service. Geoscience Frontiers, 11(5), 1859-1873.

3.1 Introduction

Natural disasters are a worldwide hazard that causes a widespread loss of life and damage to infrastructure with associated economic losses. The advent of modern computational methods and hardware has allowed models to be developed to simulate and predict these complex phenomena. These models represent complex phenomena that are contributed by a large number of factors. Due to this, they usually have high computational requirements and are not feasible to run in an operational environment. Deriving accurate risk metrics from such models can require hundreds of thousands of possible scenarios, collectively referred to as an *ensemble* to be run. However, even a single simulation is a complex calculation based on interrelationships between different parameters, and must also deal with geographical information data sets. Running ensembles on a single computer or a small cluster can result in bottlenecks due to data access and processing constraints. Thus, it may take several hours to days to fully cover the required perimeter space. Furthermore, in a real-time operational environment where ensemble simulations are being run to predict real wildfires, resource constraints from a limited computing pool may delay predictions required for operational management with unwanted consequences for controlling fires effectively or timely evacuations from regions in danger.

Research carried out in recent years has put forward Cloud Computing frameworks as a possible solution to increase the efficiency of the prediction tools and make these services available to many users in a scalable way. Cloud Computing, which is based on principles of distributed computing, possesses the features of pooling, sharing, integrated computing technologies, and vast computer resources [164]. Cloud infrastructure itself does not decrease the computation time for individual simulation in an ensemble. But, it provides a means to reduce the overall time of the ensemble as it allows elastic on-demand access to almost unlimited storage, network, and computational processing. However, this access to the Cloud resources must be coupled with an effective control mechanism in the system design to manage the resources and support the prediction models in optimal manners.

It is desirable to offer the functionality of ensemble simulations of disaster models as end services. However, the inherent nature of ensemble simulations can invite several challenges regarding the resource utilization, user requirements and cost incurred. For

ease-of-use, there must also be an effective mechanism that can handle the ensemble simulations within the Cloud environment without requiring frequent user interventions. Kalabokidis et al. [71] initiated the use of Cloud Computing for fire simulation model while Garg et al. [20] provided a conceptual model to provide a scalable wildfire prediction over the Cloud environment. Garg et al. [20] proposed sparkCloud service - a web-based Cloud platform system to demonstrate the elastic and scalable Cloud solution for wildfire prediction model based on user requests and deadline requirements. KC et al. [165] proposed a conceptual solution framework to offer different disaster-related functionalities as a service over Cloud environment. However, no studies to date have clearly defined a mechanism for enabling the ensemble simulations of any natural disaster models as end services over the Cloud environment with optimized cost and resource utilization. Moreover, there are no specific studies that define how to enable ensemble simulations of natural disaster models over the Cloud foundation with minimal user interventions during the simulation run.

As such, this study puts forward a framework that helps in the realization of the ensemble of disaster simulations as end services over the Cloud environment. The proposed framework considers the user requirements and minimizes the cost of operation in two distinct phases. In the first phase, the possible incurred cost is minimized through efficient distribution of the simulations among cost-efficient workers while still complying to the user requirements. The second phase further minimizes the cost of operation by intelligently choosing the instances based on different pricing models - on-demand, reserved and spot. This study validates the working of the proposed system design by implementing the design with a wildfire prediction tool, Spark [3], in the Cloud environment. In the proposed system, end-users can ubiquitously access and use the ensemble services via a web interface using the internet with minimal cost.

3.2 Model and Challenges

In this section, we first discuss the ensemble of a general disaster model with different components and phases of simulating the dynamics of the phenomenon over time. We then explain in detail the challenges associated with offering such ensembles of disaster simulations as end services.

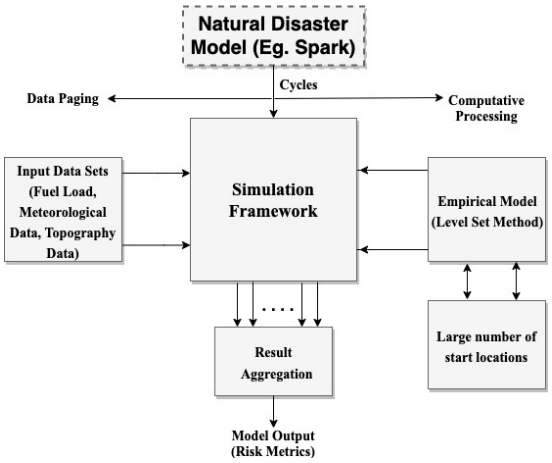


Figure 3.1: An ensemble of a General Disaster Model

3.2.1 Ensemble of Natural Disaster Model

For disasters such as wildfires, the parameter space of factors affecting the fire can be mapped to possible outcomes allowing the detailed risk metrics to be calculated. These input factors can include parameters such as the starting location for the fire, the wind conditions, and the air temperature. The possible outcomes can be the total area burned and whether the fire impacts any areas with homes or infrastructure. The number of required simulations can scale exponentially with the number of input parameters. Natural disaster models such as Spark, usually consist of two distinct cycles - data paging and computative processing, to simulate the behavior of the disasters. An overview of an ensemble of a general disaster model is shown in Figure 3.1. In Data paging cycle, all the required input data sets are collected and fed into the simulation framework. During computative processing, empirical models are used to predict the progression of the disaster phenomenon over time. The key feature of an ensemble of disaster simulation is the requirement of hundreds to thousands of simulations to derive more accurate risk metrics. For operational management, any predictions about the outspread of the disaster can be significant in saving lives and physical properties.

3.2.2 Challenges

Predicting accurate risks of natural disasters using an ensemble has a principle challenge of managing the execution of a large number of simulations in time and resource-efficient manner. As such, all the challenges associated with developing different mechanisms

to efficiently deploy the ensemble of disaster simulations as end services over a Cloud foundation, are described as follow.

3.2.2.1 Achieving Ensemble of simulations over multiple Cloud instances with minimal user intervention

While executing an ensemble of simulations over multiple Cloud instances, the scenarios for the ensemble have to be created through several simulations over a large number of start locations [20]. These simulations have to be distributed over multiple instances. Running the simulations in batch mode can save time as a single data paging would work for all the simulations in the batch, but, the same is not true for computative processing. It can be optimal to divide the ensemble scenario into several groups of simulation as subjobs. These subjobs have to be independently assigned to the instances within the system. Moreover, the methods how the multiple outputs from each simulation are collected and stored during Result Aggregation and processed are equally important and challenging for better interpretation of the results [165]. Achieving all these requirements effortlessly with minimal user intervention can be a big challenge.

3.2.2.2 Supporting computational complexity of ensemble simulations over the Cloud environments with optimal resource utilization

With the features of almost unlimited compute, network, and storage, Cloud Computing can support the computational complexities of ensemble simulations. But scaling out a pool of Cloud instances for every request received within the system is not a practical solution [166]. Such provision can waste the computing resources within the system environment as some resources may remain idle during the operation. A significantly large number of simulations needs to be run to offer the ensemble of disaster simulations as end services to multiple users. The computative processing for such a large number of simulations can be compute-intensive, and thus, the ensemble has to be broken into simpler groups of simulations, subjobs. Such fractions can independently run in multiple workers in batch mode. It can be a non-trivial task to define a mechanism that provides rational support to execute the computations required by the ensemble. Such a system should also consider all the related constraints and system scenarios at the given instant of the time. The decision to allocate new resources and delete the existing resources

from the available pool can be critical. It becomes more challenging when the system has to consider simultaneous user requests from multiple users. Advanced scheduling and optimization mechanisms may be required to ensure the maximum resource utilization while supporting the computational complexity of the ensemble of simulations.

3.2.2.3 Trade-off between user requirements and cost

The user requirements have to be considered while offering the ensembles as services to end-users. If required, the user requirements may have to be prioritized, and operations might have to be customized to meet the strict user requirements in terms of time and cost. Moreover, Cloud resources may be massively used as there may be a large number of concurrent users accessing the service. It can be a challenging task to ensure minimal operating cost while complying strictly with the user needs and requirements. The situations dealing with the trade-off between the operational cost and user requirements can be tricky to handle within the system. The diverse range of cost brought in by different pricing models can add more complexity to the trade-off between the requirements and the operating cost. To facilitate such capabilities, we have included Resource Handler in the proposed framework, that intelligently selects the cost-efficient resources entirely based on the user requirements.

3.3 Proposed Framework

In this section, we describe our proposed system design (as shown in Figure 3.2) that offers the ensemble as end-services by addressing the associated challenges. The system design consists of Users, Control Logic, and Cloud Infrastructure as major entities. Optimizer in the Control Logic takes the user input and requirements entered into the system through a web-interface into consideration to determine the best distribution of simulations for executing the ensemble. Resource Manager accepts the service request with corresponding worker configuration determined by Optimizer. It then selects the cost-efficient Cloud instances strictly based on their urgency level scores, calculated when the requests enter the block. Ensemble Distributor creates several variable-sized fractions of ensembles as subjobs in an orderly fashion before assigning them to the workers in the Cloud infrastructure. Multiple workers execute different runs of simulations to

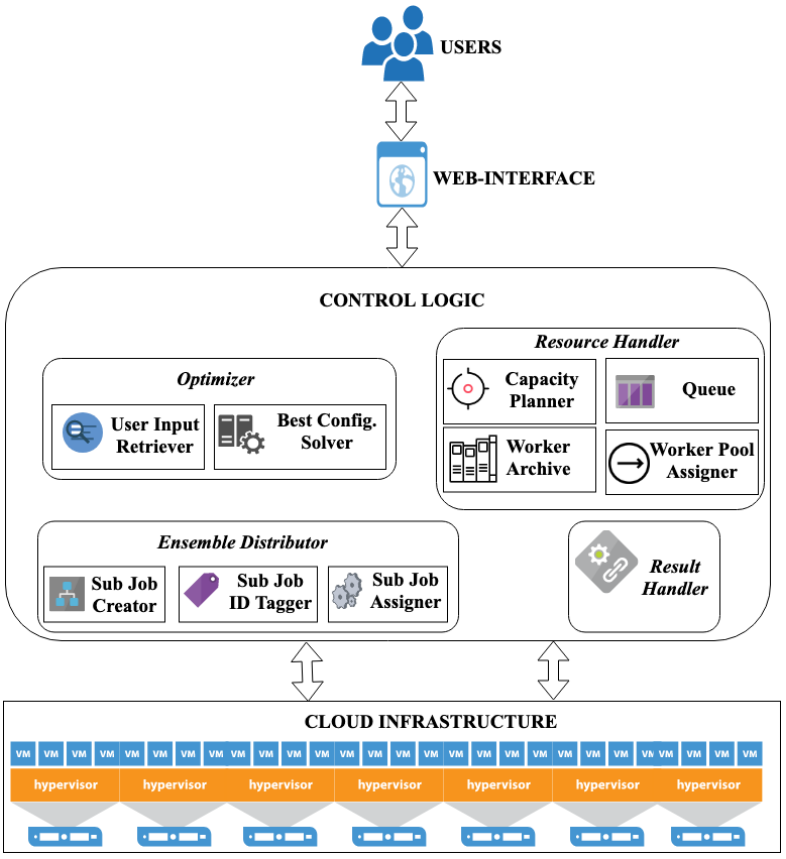


Figure 3.2: Component Overview of Proposed System Design

contribute to the ensemble simulations ultimately. The filtered results are collected by Result Collector, which can be accessed by the user through the same web-interface after all the workers have completed their subjobs. The overall sequence of the operations in the proposed system with the message exchange between the components is given in Figure 3.3. The system design is explained in detail with its components below:

3.3.1 Users

The users submit a service request along with input files and time and cost requirements through web-interface to initiate an ensemble simulation of the disaster model. The interface contains input fields for the time and cost requirements while the configurations of disaster simulations are defined in the input XML file. A sample of input XML file is shown in Figure 3.4. The XML file defines the location where the fire starts, the number of different fire start locations, simulation time and other information related to the input and output data sets. The input files contain the meteorological data and fuel information required for the fire simulation. The configuration defines the location, the

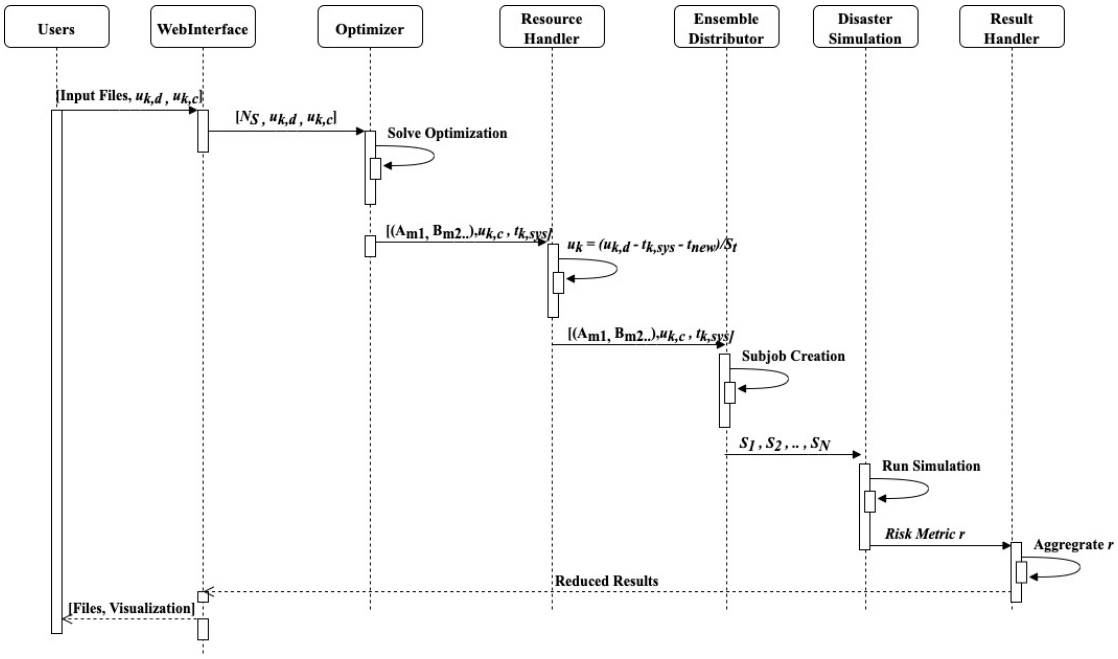


Figure 3.3: Sequential Overview of the Proposed System Design
**Note: The symbols and notations are listed in Appendix 1*

number of simulations in the ensemble and input data to be considered for calculation of the risk metrics from the simulation. Web-interface hides all the other steps that are carried out within the framework so as to serve a user request. The users get to download the result files through the same interface once the execution of the ensemble is completed.

3.3.2 Control Logic

Control Logic retrieves the user input and requirements and performs several operations through its components so that the ensemble of simulations are optimally distributed among multiple Cloud instances. The components of this entity are further discussed below with their functions.

3.3.2.1 Optimizer

It employs a user-based policy to manage the multiple user requests in an efficient manner that ensures the user requirements are met with maximum resource utilization. This block uses the retrieved user requirements in conjunction with benchmark records to give the best configuration for the job execution with minimal cost. The series of

```
<input globalname="Curing/Layer default">100</input>
<input globalname="Curing/Layer interpolation">1</input>
<input globalname="Elevation/Layer default">0</input>
<input globalname="Elevation/Layer projection WKT">EPSG:28355</input>
<input globalname="Elevation/Layer source file">../../../../input/Terrain_inputs/Spark_DEM.tif</input>
<input globalname="Elevation/Layer interpolation">1</input>
<input globalname="Fire history/Layer default">0</input>
<input globalname="Fire history/Layer projection WKT">EPSG:28355</input>
<input globalname="Fire history/Layer source file">../../../../input/Terrain_inputs/FireHistorySept2017.tif</input>
<input globalname="Fuel load/Layer interpolation">1</input>
<input globalname="Classification/Layer projection WKT">EPSG:28355</input>
<input globalname="Classification/Layer source file">../../../../input/Terrain_inputs/Fuel_TAS.tif</input>
<input globalname="Gridded/Source directory">../../../../input/Meteorological_inputs</input>
<input globalname="Gridded/Layer projection WKT">EPSG:4326</input>
<input globalname="Gridded/Time conversion coefficient">1</input>
<input globalname="Gridded/Wind/Layer direction source file"/>
<input globalname="Gridded/Wind/Layer direction source filter">Wind_2MC_2018_32bit_Wind_Dir_SFC.nc</input>
<input globalname="Gridded/Wind/Layer magnitude source file"/>
<input globalname="Gridded/Wind/Layer magnitude source filter">*_WindMagKmh_*.nc</input>
<input globalname="Gridded/Relative humidity/Layer source filter">*_RH_*.nc</input>
<input globalname="Initialisation Python input file">../../../../input/FuelTypes.xml</input>
<input globalname="Initialisation Python input file 2">../../../../input/Master_Spark_Lookup_v1_vesta.csv</input>
<input globalname="Initialisation Python script">
"Define how far out radius should search in this case use 6km"
xy_range = [-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]
range_len = len(xy_range)
centroid_easting = 538424.698422781
centroid_northing = 5246922.78512832
x_direction = int((seed - 1) / range_len)
y_direction = (seed - 1) - (range_len * x_direction)
nx_direction = int((seed - 1) / range_len)
ny_direction = (seed - 1) - (range_len * x_direction)
"Use a 1km grid of ignitions"
new_easting = centroid_easting + 1000 * xy_range[x_direction]
new_northing = centroid_northing + 1000 * xy_range[y_direction]
lat, long = utm.to_latlon(new_easting, new_northing, 55, 'G')
"******Spark inputs******"
lat = [lat]
long = [long]
radius = [200]
time = [0]</input>
<input globalname="Start time">2018-12-31T13:00:00+11:00</input>
<input globalname="End time"/>
<input globalname="Number of simulations">10</input>
<input globalname="Random seed">0</input>
```

Figure 3.4: A sample XML configuration file with key configuration parameters

operations in this block is algorithmically explained in Algorithm 1. Efficient resource utilization and cost is achieved through several sub-components, which are described below:

User Input Retriever This component retrieves the user inputs and requirements from the service request initiated by the end-users. It also defines the job complexity in terms of the number of simulations required for the ensemble. The configuration for the ensemble is also retrieved. These requirements are useful for determining the efficient resource for the service request.

Best Configuration Solver It deals with the efficient creation of variable fractions of the ensemble that ensures the user requirements are met with minimal cost. This

component undertakes the first of the two optimization tasks in the proposed system design by efficiently creating multiple fractions of the ensemble simulations as subjobs.

While deploying an ensemble of simulations over the Clouds, the ensemble has to be divided into several variable-sized fractions so that multiple workers can independently execute the simulations. The number and size of the fractions are the two most important factors in the deployment, which should be determined based on several constraints. The user requirements have to be considered as well during the deployment of the ensemble as end-services. The availability of different flavors of Cloud instances as workers with varying capabilities of computation is also a constraint in the problem formulation. As such, distribution of simulations in an ensemble to create several variable-sized fractions of the requests can be formulated as an optimization problem that minimizes the incurred cost of operation as explained below.

Let,
M_i be the worker of different flavors/types i,
p_{M_i} be the number of worker of type M_i in the best configuration,
C_{M_i} be the operating cost associated with the worker type M_i,
t_{j,M_i} be the time of operation for worker j of flavor M_i,
N_S be the total number of the simulations in the user request,
n_{s,j,M_i} be the number of simulation run by worker j of type M_i,
T_u be the user requirement of time,
C_u be the user requirement of cost,
N be the total number of different flavors of the workers,

The efficient distribution of an ensemble for a particular service request k can be formulated as:

$$\begin{aligned} min \qquad C &= \sum_{i=1}^{N}\sum_{j=1}^{p_{M_i}} C_{M_i} \times \ t_{j,M_i} \\ s.t. \quad \sum_{i=1}^{N}\sum_{j=1}^{p_{M_i}} n_{s,j,M_i} &= \ N_S \\ \sum_{i=1}^{N}\sum_{j=1}^{p_{M_i}} C_{M_i} \times \ t_{j,M_i} &\leq \ C_u \\ \forall j \in \{1,2,....,N_{M_i}\}, i \in \{1,2,...,N\}, \ \ 0 \leq t_{j,M_i} &\leq \ T_u \\ p_{M_1}, C_{M_i} &\geq 0 \end{aligned} \tag{3.1}$$

where,

t_{j,M_i} is the time for which the jth instance of flavor type M_i runs and n_{s,j,M_i} is the number of simulations in the fraction which the jth instance of flavor type M_i executes. The first constraint represents the number of simulations required in an ensemble while the second constraint is the related to the user-defined cost such that the feasible operating cost should always be less than or equal to the user-defined cost. The third constraint represents the user-defined time constraint while the last constraint defines the non-negativity of number and operating cost of the Cloud instances.

Algorithm 1 Algorithm for Operation of Optimizer

Input: u_k, $u_{k,d}$, $u_{k,c}$
Output: $[(A_{M_1}, B_{M_2}, ...)$, $u_{k,d}$, $t_{k,sys}]$
1: For every u_k
Retrieve $u_{k,d}$, $u_{k,c}$, N_S
2: Formulate as an optimization problem min C
3: Determine n_{s,j,M_i} using $G(T_u, M_i)$
4: Solve the optimization problem using Linear Optimization techniques
5: **return** $[(A_{M_1}, B_{M_2}, ...)$, $u_{k,d}$, $t_{k,sys}]$

The problem has to consider finding an efficient way of assigning the different numbers of simulations to each worker based on its type. This is a complex NP-Hard optimization which cannot be solved within polynomial time. For this thesis, a heuristic is considered that determines the variables n_{s,j,M_i} from the benchmark experiments using the function $G(T_u, M_i)$ defined as:

$$G(T_u, M_i) = \{n : n = Max\{M_i, n\} and n_{M_i} \leq T_u \ \}$$

The variable t_{j,M_i} is assigned a constant urgent value deduced after experimental studies. The NP-hard problem now becomes linear and can be solved using existing linear optimization techniques. The solution gives the efficient distribution of the ensemble concerning the best configuration of Cloud instances.

For any user service request u_k with associated requirements of cost $u_{k,c}$ and time $u_{k,d}$, this block gives out the efficient ensemble distribution in the form [(A_{M_1},B_{M_2}, ...), $u_{k,d}$, $t_{k,sys}$] where A, B, .. are the numbers of Cloud instances of flavor types M_1, M_2,... respectively required in the cluster to execute the request and $t_{k,sys}$ is the time for which the user request u_k has been in the system. This information is passed on to Resource Handler for the allocation of the resources. The working of Optimizer is algorithmically summarized in Algorithm 1.

3.3.2.2 Resource Handler

Resource Handler is the block in the proposed system design that undertakes the second phase of optimization by choosing the most cost-efficient instances based on different Cloud pricing models. The choice of Cloud instances based on pricing models can significantly minimize the cost of operation. The deployment of the ensemble runs on spot instances can incur comparatively lower cost when compared with on-demand instances, but the reliability of such spot instances is less. As such, we introduce three different categories for the user requests-high, medium and low, strictly based on their deadlines (similar to the concept explained in [167]). A predefined standard S_t obtained from benchmark studies is taken as a reference, and all the user requirements of the deadline ($u_{k,d}$) are compared against the standard to give a parameter, urgency level UL_k given as follow.

$$UL_k = \frac{(u_{k,d} - t_{k,sys})}{S_t} \tag{3.2}$$

where,

$t_{k,sys}$ is the time elapsed after the user request u_k is received within the system.

The urgent requests ($1 \leq UL_k < 2$) is directed towards the Capacity Planner, while for other user requests ($UL_k \geq 2$), the creation of new Cloud instances is considered by adding t_{new}, the average time required to create the new Cloud instance, in Equation

Table 3.1: Different urgency Levels of User Requests

Level	**Values of UL_k**
High	$1 \leq UL_k < 2$
Medium	$2 \leq UL_k < 3$
Low	$UL_k \geq 3$

3.2 and the urgency level UL_k is updated accordingly as follows.

$$UL_k = \frac{(u_{k,d} - t_{k,sys} - t_{new})}{S_t} \tag{3.3}$$

The updated parameter UL_k determines the position of the user request u_k in the queue and which types of instances are allocated to the request. The three defined categories for the values of UL_k are listed in Table 3.1.

Any service request with a value of UL_k less than one (1) is rejected as the request is not feasible. The requests under high urgency level can only be run once in the system and hence are serviced using highly reliable on-demand instances, handled by Capacity Planner. The requests under medium and low categories are served with spot instances with relatively low reliability. If unsuccessful, the requests are rerun with altered urgency level values with more reliable instances. The proposed system does not consider fault tolerance and checkpointing for recovery in spot instances. The working of Resource Handler is algorithmically discussed in Algorithm 2. The components of Resource Handler are discussed further below.

Capacity Planner This block is included in the proposed system to save time for creating new instances for the user requests with high urgency levels. It keeps track of the rate of the urgent user service requests that are received at Resource Handler in a queue CP_q. In the proposed system, especially for the user requests with urgent deadlines, there must be workers readily available as the time required for the creation of new workers can significantly compromise the urgency of the requests. To overcome this issue, Capacity Planner makes sure that there is at least a minimum number of different workers always available in the system. Capacity Planner can increase the number of already available worker based on the emergency situation and the demand of user requests with urgent deadlines. The additional cost of keeping the cloud instances alive even without any operation can be distributed over the users who initiate such requests.

Algorithm 2 Algorithm for Operation of Resource Handler

Input: $[(A_{M_1}, B_{M_2}, ...)$, $u_{k,d}$, $t_{k,sys}]$
Output: $[u_k$, $(A_{M_1}, B_{M_2}, ...)$, D_{type}, $bid_{price}]$

1: For every u_k, Calculate $UL_k =$
$$\frac{(u_{k,d} - t_{k,sys})}{S_t}$$
2: **if** UL_k <1 **then**
3: reject u_k
4: **else if** $1 \leq UL_k$ <2 **then**
5: $D_{type} = on - demand$
6: bid_{price} =0
7: Send u_k to Capacity Planner Queue CP_q
8: Push u_k to **R**
9: **else**
10: Update UL_k as follow: $UL_k =$
$$\frac{(u_{k,d} - t_{k,sys} - t_{new})}{S_t}$$
11: **if** UL_k <2 **then**
12: go to Step 4
13: **else**
14: Push the request u_k into the queue **Q** and sort **Q** based on the values of UL_k
15: **end if**
16: **end if**
17: **while** u_k on the top of the queue **Q** **do**
18: **if** $2 \leq UL_k$ <3 **then**
19: $D_{type} = spot$
20: $bid_{price} = bid_{medium}$
21: **else**
22: $D_{type} = spot$
23: $bid_{price} = bid_{low}$
24: **end if**
25: Retrieve the number of free and available workers $n_{M,i}$
26: **for** every M_i **do**
27: Calculate $\Delta n_{M,i} = n_{M,i} - X_i, X = A, B, ..$
28: **if** $\Delta n_{M,i} < 0$ **then**
29: Create $\Delta n_{M,i}$ new Cloud instances of flavor type M_i
30: Update information in Worker Archive
31: **end if**
32: **end for**
33: Forward u_k to Ensemble Distributor
34: Remove u_k from **Q** & Push u_k to **R**
35: **if** u_k == **completed then**
36: Remove u_k from **R**
37: Update workers' status in Worker Archive
38: return
39: **else if** u_k == **failed then**
40: go to step 1
41: **else**
42: wait
43: **end if**
44: **end while**

Capacity Planner can use M/M/c [168] queuing model to estimate the number of on-demand instances to be created in advance. For the model, λ is the arrival rate of urgent user requests, μ is the service rate, and c is the number of clusters. For the arrival rate of requests and service rate of the system assumed to follow Poisson distribution, the minimum number of workers of each flavor type M_i required can be determined using Erlang B formula [169] (Equation 3.4) with very small (nearly zero) value of blocking probability.

$$N_{M_i} = min\{y : B(X, y) \leq T\}, y\epsilon\mathbb{N} \tag{3.4}$$

where,

$X = \frac{\lambda}{\mu}$ is the traffic offered in Erlang, T is the desired blocking probability (very small) for Capacity Planner, and $B(X, y)$ is the blocking probability expressed as follow.

$$B(X, y) = \frac{\frac{X^y}{m!}}{\sum_{i=0}^{y} \frac{X^i}{i!}} \tag{3.5}$$

For any instant of time, the number of Cloud instances to be created in advance can be calculated using the historical data (after determining the values of μ and λ). Capacity Planner determines the number of minimum workers required for an almost zero blocking probability in a fixed interval of time (average time for the creation of the new instances). The operation of Capacity Planner for urgent user requests is algorithmically presented in Algorithm 3.

In addition to the urgent user requests, this component is useful in deciding when to reserve the Cloud instances to further minimize the cost based on the historical records. For example, for wildfire ensembles, based on the historical information about the arrival rate of user requests, Capacity Planner can reserve a pool of instances during the summer.

Queue It keeps the record of all the user requests with the corresponding efficient ensemble distribution scheme given by Optimizer. For each user request u_k, urgency level UL_k is calculated. The queue stores all the user requests in a sorted manner such that UL_k with lower values are placed on the top. The required Cloud resources are allocated to the requests on a one-at-a-time basis. There is an additional queue R which keeps the record of all the running user requests with corresponding $t_{k,sys}$.

Algorithm 3 Algorithm for Capacity Planner of Urgent user requests

Input: $[\lambda, \mu]$
Output: $[N_c]$

1: Forward all user requests u_k in CP_q to Ensemble Distributor
2: For every time interval t (average time required for instance creation), Retrieve updated λ_t, and μ_t
3: Retrieve the number of free and available on-demand workers $nd_{M,i}$
4: **for** every M_i **do**
5: Calculate $Nd_{M,i}$,
$Nd_{M,i} = min\{y : B(X, y) \leq T, y \epsilon \mathbb{N}$
6: Calculate $\Delta nd_{M,i} = nd_{M,i} - Nd_{M,i}$
7: **if** $\Delta nd_{M,i} < 0$ **then**
8: Create $|\Delta nd_{M,i}|$ new on-demand instances of flavor type M_i
9: **else**
10: Delete $\Delta nd_{M,i}$ on-demand instances
11: **end if**
12: Update information in Worker Archive
13: **end for**
14: Wait until the end of time interval t
15: Go to Step 1

Worker Archive It keeps a record of all the workers within the proposed system. The information about the flavor, pricing model and availability of the worker is essential for effective resource allocation. For any cluster size requested by the service request, this component provides the information about the availability of the workers running in the system to prevent the creation of new instances if not required.

Worker Pool Assigner It decides to deploy the cluster of Cloud instances based on different pricing models strictly based on the category defined by the values of UL_k. This component handles the trade-off between the urgency level and cost by altering the reliability of the instances accordingly. If the user request has an urgent deadline, Worker Pool Assigner opts the on-demand instances with higher reliability. Worker Pool Assigner bids for spot Cloud instances for the job in medium and low categories with bid prices $bid_{medium} > bid_{low}$ established based on historical information. If the requests in medium and low categories are not completed due to the unavailability of the spot instances, the requests are pushed into the queue Q with an altered value of $t_{k,sys}$.

3.3.2.3 Ensemble Distributor

Worker Distributor handles the creation and distribution of the variable-sized fractions of the ensemble initiated by the user service request following the cluster size and type defined by Optimizer and Resource Handler. For a worker W_{M_i} of flavor type M_i, Ensemble Distributor retrieves the number of simulations in a process n_{sp} and the number of simultaneous processes of the disaster model x_p and assigns the corresponding fractions to the workers. Depending on the computational capability of the instances, the worker nodes may or may not implement multiple processes of prediction software tool simultaneously. The functionalities of this block are algorithmically represented in Algorithm 4. Worker Distributor in turns consists of the following components:

Subjob Creator It creates several subjobs with variable sizes based on the configuration given by Optimizer. All the subjobs possess the characteristics of the main job and can be run in an independent mode. The last subjob created by the Subjob Creator compensates for any additional number of simulations in the best configuration by assigning a lesser number of simulations to the worker under that particular subjob.

Subjob ID Tagger It adds identification tags to all the created subjobs before assigning them to the workers. The information tags are received and decoded for customizing the simulation runs for contributing to the specified fraction of the entire ensemble run.

Subjob Assigner After addition of the identification tags, Subjob Assigner assigns respective subjobs to the corresponding workers in the cluster. The last subjob that compensates the over-estimation of the best configuration is chosen such that the operation cost is reduced for the service request. All the necessary files required for the execution of the prediction software tool are downloaded in the worker nodes from the master controller within the system environment.

3.3.2.4 Result Handler

During the execution of simulations in the workers, multiple output files are created at the end of each simulation run in different formats after processing a more significant

Algorithm 4 Algorithm for Work Division and Distribution

Input: Cluster Size, N_c
Output: Intermittent Result Files
1: Retrieve n_{sp} and x_p for each W_{M_i}
2: Create S_N subjobs where subjob S_N acts as compensating subjob with possibly less number of simulations
3: Add subjob identification tags # to subjobs
4: Assign x_p subjobs as different fractions to corresponding worker W_{M_i}
5: Assign compensating subjob S_N to the least costly worker W_N in accordance to the configuration given by Optimizer
6: Wait until all subjobs are completed
7: Reduce the result files
8: **return** *Reduced Result Files*

amount of relevant data. The transfer of the entire simulation results back and forth between the worker nodes and the master node can create a network bottleneck, thereby compromising the performance of the system. As such, Result Handler makes sure only the significantly important information is extracted out from the outputs generated after every run of the simulation. The reduced but important output information is gathered in a centralized fashion under a single folder that references to the subjob identification tag. Upon completion of the execution of the subjobs, only the critical information set with relatively small data size is sent back to the master node. The master node stores the files in a centralized fashion. After successful uploading of the data to the master instance, the worker nodes delete all the files related to the completed job and make themselves available to take new subjobs. When the master node receives all the relevant output files from the worker nodes under a single folder referencing to the main job, the job is deemed to be complete. Upon completion of the main job, the users can see the status reflected in the web interface and download all the output files for further interpretation and visualization.

3.3.3 Cloud Infrastructure

Cloud Infrastructure uses public Clouds to provide required hardware foundation in terms of virtual machines of different flavor types to support the computational needs of ensemble simulations. All the workers have Spark tool pre-installed on them that run different processes with different start points to contribute to the ensemble simulation as initiated by the service request.

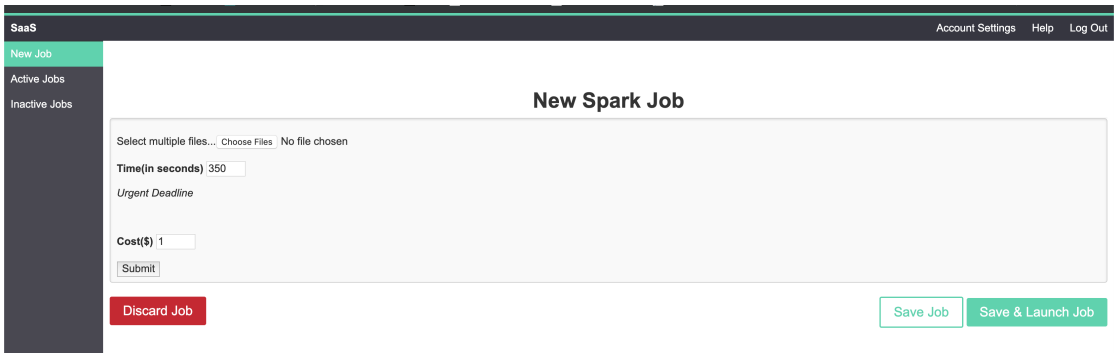


Figure 3.5: Web-Interface to initiate request in Proposed System

3.4 Evaluation

The working of the proposed system design is validated through a real prototype which utilizes Spark, a wildfire simulation tool that predicts the progression of a wildfire. Spark offers a modular framework for wildfire spread prediction where several packages and models can easily be plugged in. These packages and models include generation of wind fields and their topographic correction, ignition models, fire-line interactions, road and transmission models and firebrand transport [3]. All the calculations required for a fire simulation in Spark are parallelized on Graphical Processing Unit (GPU) architecture such that the simulations can run faster than in real-time. This is true for all the simulations that aggregate in an ensemble to give more accurate risk metrics of a fire.

All the steps explained in the proposed foundation system are closely followed during the evaluation. The proposed solution provides modular system design, thereby offering flexibility in changing the components (e.g. wildfire simulator) with other disaster simulation tool. Java is the main programming language used to enable different mechanisms within the system. A web-based user interface is developed to facilitate the users to access the system and initiate the request to use ensemble simulations as end services. The web-interface to upload the files and enter the user requirements of time and cost is shown in Figure 3.5.

In the following section, we first give the details of the use case scenario and Cloud infrastructure that are utilized for validating the proposed system. Then, the results for benchmarking of Spark in the Cloud environment are presented and discussed. The critical parameters required for subsequent operations in different blocks are determined through the benchmark studies. Based on these results, *Optimizer* decides how to create

multiple fractions with variable size to contribute to the ensemble required by the users. The influence of user-defined deadlines on the choice of instances based on different pricing models and subsequently on the total cost of operation is also studied. Finally, we evaluate the overall performance of the proposed system against the comparable on-premise system and bag-of-task type execution over the Clouds.

3.4.1 Ensemble Use Case Scenario

For evaluation, a real ensemble scenario using data kindly provided by the Tasmania Fire Service (TFS) is used. The scenario consists of a total of 169 simulations starting at equally spaced locations 1 km apart on a 13 km $\times$ 13 km grid around a central point. Each simulation is configured to run for nine hours after the fire has started at a point. The model is configured for various fuel types in Tasmania and also takes into account impact with any urban areas by counting the number of urban cells burnt for a particular wildfire. This prediction model falls into a risk modeling category of ensemble simulations analyzing the risks of a wildfire starting at an unknown location under a particular set of weather conditions. This fundamental design can be further extended to work for operational modeling that deals with direct suppression and evacuation efforts once the fire has been reported to start.

3.4.2 Setting Up the Cloud Environment

In this experiment, Nectar Cloud [170], an OpenStack-based community Cloud infrastructure, is used as an emulated Amazon Cloud environment for conducting different experiments. It is clear from the benchmark studies that the number of cores in the Cloud instances is the key factor in determining the time taken to run a fixed number of simulations. All the available cheapest instances with their hardware specifications along with their unit cost are listed below in Table 3.2. The cost of operating the instances is set according to the Amazon Web Services (AWS) [171] pricing model. The data transferred into Amazon Cloud and data transfer between the instances in the same availability zones are free. Thus, data transfer cost is not taken into consideration in this thesis. But, the time taken for the data transfer is considered for total operation time, and hence, the time taken for data transfer contributes to the total operating cost

Table 3.2: Different Flavors in Nectar Cloud

VCPUs	Flavor	RAM (GB)	On-Demand Cost($/hr)	Spot Cost ($/hr)	Reserved Cost($/hr)
1	m2.xsmall	2	0.0146	0.0035	0.01
2	t3.small	2	0.0209	0.0051	0.0142
4	t3.medium	4	0.0418	0.01	0.0284

and time. As for the spot instances, resources are abruptly taken out from the system design during operation with the probabilities calculated using existing works.

3.4.3 Benchmarking of Spark over Cloud Environment

For the natural disaster simulations like fire simulations, the time taken for each simulation is dependent on several factors and has not been previously studied. Creating several batches without a general understanding of the fire dynamics can contribute to inefficient operation in the proposed system design. Given the parallelization of the simulations in Spark, independently accommodating simultaneous Spark process can enhance the resource utilization. As such, we conducted a set of different experiments under the benchmark study to determine the efficient distribution of the simulations in the ensemble based on the processing capacities of the workers. For all the different flavors of instances available in the Cloud environment, we analyze the implications of the processing capabilities and cores in the total execution time. The benchmark tests were carried out in two distinct phases - first with the different number of simulations in each instance and later with several simultaneous processes of disaster model in the instance. Moreover, the key parameters S_t and n_{s,j,M_i} are also determined after the experimental analyses.

3.4.3.1 Number of Simulations

For each of the different instance flavors, a series of experiments was carried with different fire start points with a batch of variable size of the simulations in each worker. For the TFS sample, there are 169 different geographical start points for the fire. The fires are started on a regularly spaced grid at 1 km intervals irrespective of the land classification. It should be noted that fire simulations starting in areas of water take significantly less time (as they terminate immediately) compared to the simulation on land. For every

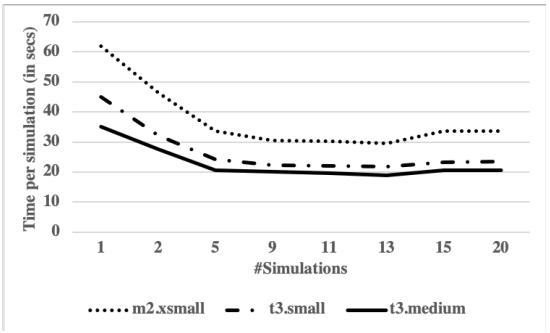


Figure 3.6: Unit Simulation Execution Time for different worker flavors

sample file, the experiments are carried out in six distinct sets of 1, 2, 5, 9, 11, 13, 15 and 20 simulations in each unit of workers. The findings for all the instance flavors are figuratively presented in Figure 3.6. Moreover, the average times taken by the worker instances to complete the different sets of simulations with a single process of Spark running are depicted in Figure 3.7.

Due to the more significant computation resources in t3.small and t3.medium workers, as compared to the m2.xsmall worker, the average unit execution time for a Spark simulation is lower in t3.small and t3.medium. Knowledge of this difference is useful while choosing the cluster of workers for a user service request with different sets of inputs. The execution time per simulation decreases when executed in batches until the saturation point (different for different instance flavors). The average time per simulation keeps improving until the set of 13 simulations for all instances and saturated after that with a slight increase. This improvement is due to a common data fetch cycle for all the simulations which can be done once when executed in a batch compared to multiple times when executed as independent units. The findings of the benchmark study show that the time performance of the system improves when simulations are executed as a batch (variable) rather than when executed independently in different machines. For a single Spark process, the execution time of the simulation increases with the increased number of simulations in the batch. The time performance of the instances beyond the saturation points is out of the scope of this thesis. Moreover, the improvement in the time performance of the instances is not linear with the increase in RAM size, as shown in Figure 3.7.

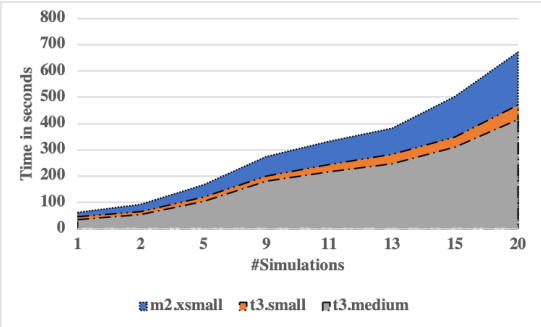


Figure 3.7: Total Simulation Execution Time for different worker flavors

3.4.3.2 Simultaneous operation of Spark processes

The disaster model, Spark, consists of two cycles during the execution of fire simulations-data paging and computative processing. Since there are large data sets involved in the process, there is a possibility of the process sitting idle while the large data sets are paged from storage into memory. Due to this, we evaluated the feasibility for running different batches in a single worker to ensure maximum resource utilization within the system environment. For all the instance flavors, tests were carried out in a way such that multiple subjobs are assigned to a single worker for simultaneous operation. Under such an operation, the worker has to execute the different Spark processes with different start points contributing to the ensemble. In a trial and error fashion, we related the total number of simulations a worker can support, for a given deadline, with the varied number of VCPUs available in the workers. Moreover, the effects of using more processor memory in the execution are compared against the performance gain achieved by accommodating multiple model processes in an instance with multiple VCPUs. The multiple subjobs run on a single machine in an independent under different configurations and the time performance of the instances were recorded for further analysis.

The time performance of different instances for multiple processes of Spark is depicted in Figure 3.8. In the figure, n_1 is the number of simulation in a single process, x is the number of Spark processes, x_n is the number of simulation in each Spark process, and $N_{S,x}$ is the total number of simulation for x Spark processes in a single machine. A single Spark process consists of two main sub-processes that are CPU-dependent computations and disk/network-dependent data operation. The efficiency of the worker nodes can increase significantly if the computation sub-process can be overlapped with the data operation of another simulation. The presence of a single processor is unable to

complement the data fetch and computation cycles. It is thus, clear that the instances with a single VCPU are not able to support the multiple processes of Spark. The time performance keeps improving until N processes are accommodated in the instances with N VCPUs, which facilitates the system to accommodate more simulations in a fraction of the ensemble. For a deadline of 300 seconds, the worker of t3.medium type with 4 VCPUs can run four simultaneous processes of Spark with a total of 32 different simulations compared to the run of three simultaneous processes with a total of 30 simulations and single process with a total of 15 simulations. When five simultaneous processes are run on the instance, there is no improvement in the total number of simulations that can be run. Based on the findings, we establish a fact that N simultaneous processes of Spark can be run a Cloud instance with N processors for optimal performance.

The performance gain, due to the increasing the number of VCPUs, out shades the same due to increased RAM sizes in the instances. Moreover, for a constant number of VCPUs in the instances, the increase in the RAM sizes does not significantly increase the total number of simulations. Thus, we focus on the most cost-effective instances with a varied number of VCPUs without any regard to the RAM sizes.

3.4.3.3 Determination of S_t and n_{s,j,M_i}

The set of 169 simulations in the ensemble was run with different locations sequentially over instances with different flavor types. The value S_t was fixed at 300 seconds by considering the fact that the average time to run the ensemble over the most powerful machine is 3912 seconds. The value S_t can easily be adjusted to make the system more responsive to the user requests. The corresponding values of n_{s,j,M_i} were then obtained from the experiments conducted in the first two phases of benchmark studies. For example, the value of n_{s,j,M_i} is 32 for T_u = 300s and M_i = t3.medium as retrieved by using the function $G(T_u, M_i)$.

3.4.3.4 Approximation of minimal time and number of simulations for an instance

Given a set of options for the Cloud instances available, it is always a non-trivial task to accurately estimate the time taken to execute a particular number of simulations and

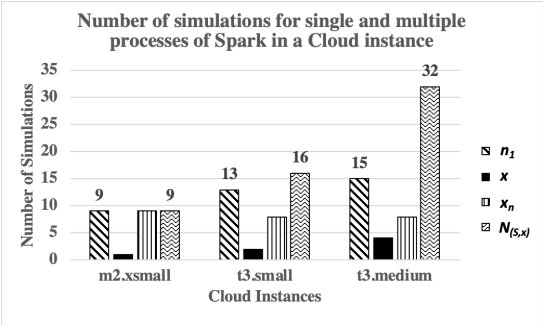


Figure 3.8: Number of Simulations for multiple processes of Spark running in the instances (Time: 300 seconds)

vice-versa. The number of simulations that can be executed by the Cloud instances increases with the increase in the size of RAM when a single process of disaster model is run. The size of RAM does not have a significant impact when multiple processes of the models are run in the instances. As such, we use linear regression to define a relationship between the number of cores, time and number of simulations to provide an approximation of time-based on the configuration of the instance. It should be noted that the accuracy of the approximation is not the primary focus of this thesis, but the cost optimization based on the results obtained from the approximation is. As such, different advanced methods can substitute the linear regression module to improve the accuracy of the approximation.

3.4.4 Experimental Setup for Evaluation of Proposed System

As previously discussed, the objective of the proposed system is to enable the ensemble of natural disaster scenarios as end services with minimal cost achieved through two phases of optimization. To evaluate the performance of the proposed system, we compare the incurred operating cost and time against the ones incurred in an on-premise system and bag-of-task type executions. For an on-premise system, we consider a single machine with the same hardware configurations as the Cloud instances have. Consequently, we have three different on-premise systems with Spark pre-installed on them. We then consider a conceptual idea of bag-of-tasks (BoT) in a distributed environment where each simulation in the ensemble requested by the user is considered a task and executed in as many machines. To compare the resource and cost optimization achieved by the proposed system, we further consider an adaptation of tasks clustering mechanism, as explained in [172] for a distributed environment. In what we call the adaptation as

modified bag-of-tasks (mBoT) execution, equal-sized clusters are formed based on the grid size of the configuration and the job complexities. For example, for a grid size of $13km \times 13km$ which yields 169 simulations, 13 clusters with 13 simulations are created. The cost and time performance of the proposed system are compared accordingly against that of the $mBoT$ execution.

3.4.4.1 Evaluation Metrics

Operation Cost The total operation cost in the proposed system design is the cost incurred to run the ensemble simulation over the Cloud environment. The cost is referred to as *Ensemble Service Cost*, which is the cost calculated taking the actual duration for which the workers are in operation while serving the user service request. The cost is calculated on a "per second" basis based on the AWS pricing model as listed in Table 3.2 using a basic unitary method.

Operation Time The operation time for a user request is the total time elapsed after the user submits the request to the system until the user gets the result files back. The operation time takes the time taken to upload the required files for the ensemble to the Cloud environment into consideration and is reflected accordingly in the total operation cost. The operating times for multiple workers allocated for a single user request can be different. The operation time for the user request is the maximum of the operating times for each worker allocated for that request.

3.4.4.2 Experimental Scenario

Different levels of user-defined deadlines For the user requirements of time, we consider three different levels of the deadline, namely High, Medium and Low are considered, as shown in Table 3.1. The experiments are repeated for five random values in each range to study the influence of urgency level on the total operating cost, and average values are presented.

Complexity of the user request The TFS samples for wildfire propagation simulation consists of a grid of $13km \times 13km$ spaced at $1km$, comprising of 169 simulations for

Table 3.3: Complexity of User Requests

Label	Grid Size (km × km)	#Simulations	Batch Size (mBoT)
small	5×5	25	5
medium	9×9	81	9
large	11×11	121	11
TFS	13×13	169	13
2 × TFS	26×26	676	26
3 × TFS	39×39	1521	39

**Note: The batch size is 1 for BoT execution, while the batch size is variable for the proposed system.*

the ensemble. To validate the effectiveness of the proposed system design, we conduct various experiments considering other sizes of the grid ($5 \times 5[small], 9 \times 9[medium]$, and $11 \times 11[large]$) in the sample files for the comparison against the on-premise system. For comparison against the bag-of-tasks type execution, the sizes (2× TFS configuration [26×26] and 3× TFS configuration [39×39]) are considered, which are listed in Table 3.3.

3.5 Results and Discussions

In this section, we discuss the results obtained while validating the proposed framework under different experimental scenarios of user requirements of time and complexities. We also present the comparative analysis of the performance of the proposed system with an on-premise system and the existing state of the art concepts of bag-of-tasks executions and job clustering. Besides, we also present a brief performance analysis of the proposed system under multiple simultaneous users with urgent deadlines.

3.5.1 Proposed System Vs On-Premise Setup

For an on-premise setup, the ensemble was run on the instances of each flavor type in a sequential manner. The same sets of the ensembles were run on the proposed system with a high level of urgency. Figure 3.9 represents the comparison of the cost incurred when the ensemble is executed using the proposed system and on-premise setup. The on-premise system is painstakingly time-consuming as the efficiency achieved by running the simulations in batches ceases after the saturation point. The comparison of operation

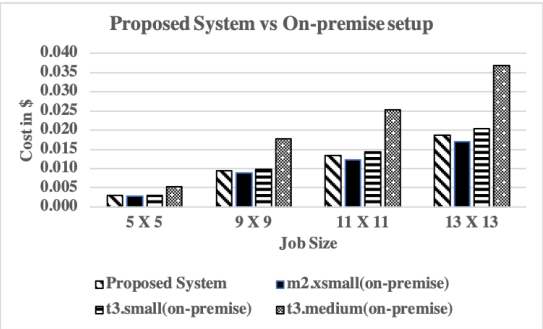


Figure 3.9: Cost Comparison between proposed and on-premise system

time is shown in Figure 3.10. On the other hand, the proposed system distributes the ensemble to multiple workers that operate within the optimal performance configurations and produces the desired output in a time-efficient manner. The cost incurred by the on-premise system with the cheapest machine (with the configuration of the cheapest Cloud instance), is the minimum of all. The proposed system operates the workers in their optimal performance region and hence achieves the operating cost closer to the on-premise cost. As such, the proposed system can offer the required services with the cost comparable to on-premise cost but with much improved time efficiency. There is no further cost minimization when the ensembles are run on the on-premise machine with better configuration as the resources are under-utilized. The operation cost is up to 98% more than the proposed system, as shown in Figure 3.9. The proposed solution ensures the resources are used optimally to avoid such under-utilization. Besides, the end-users get added benefits from the proposed system as the system setup, configuration, and dependencies are well-handled. The same could be a cumbersome task in the on-premise system.

3.5.2 Proposed System Vs Bag-of-Task Execution

The simulations in an ensemble are independent units of work without any dependencies among themselves. Thus, for bag-of-tasks type of execution, we consider each simulation as a task and run them independently in a single machine. Figure 3.11(a) shows the comparison between the cost incurred within the proposed system and bag-of-task type execution. The bag-of-tasks execution runs the simulation for a lesser time, but on the other hand, the execution incurs significantly high cost (131-316%) more than that of the proposed system). The large data sets required for each unit of simulation have to be fetched into the workers. Consequently, the improvement in time performance and

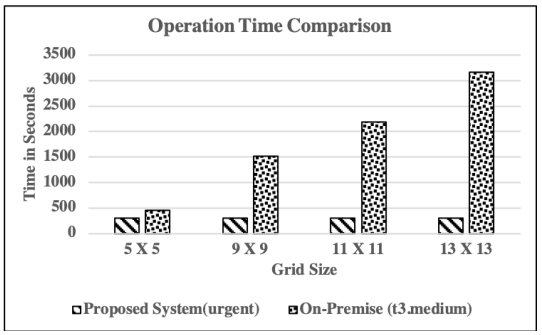
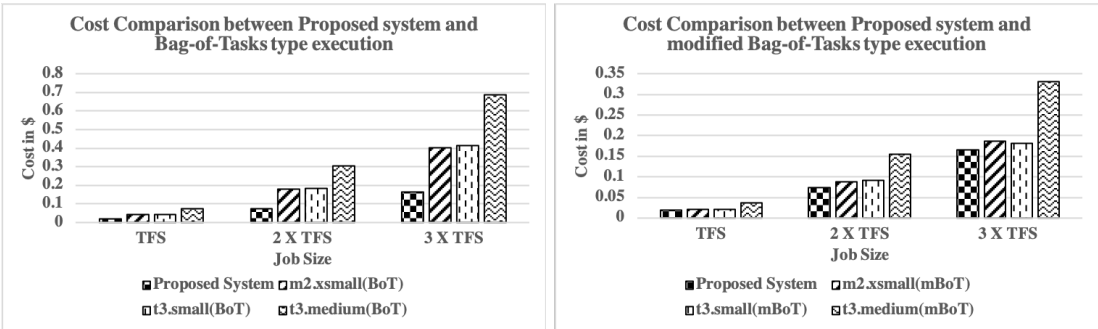


Figure 3.10: Comparison of Operation time between proposed and on-premise system



(a) (b)

Figure 3.11: Cost Comparison between the Proposed System and (a) Bag-of-Tasks Execution and (b) Modified Bag-of-Tasks Execution

resource utilization brought by the running the simulations in batches is non-existent when each simulation is run independently in separate machines.

Moreover, we divide the total number of simulations in an ensemble into several subjobs with an equal number of simulations. Each subjob is considered to be a unit of work and run in as many workers in a modified bag-of-task execution. Instances of all three flavor types are considered for the modified bag-of-task execution. Figure 3.11 shows the comparison of the cost incurred in the modified execution and the proposed system. The cost incurred in the modified execution is 9-108% more than that of the proposed system. The finding reflects the fact that the execution of an equal number of simulations in different fractions in an ensemble is not the optimal way of running the ensemble. The cost-efficiency of the proposed system over the modified bag-of-tasks type execution increases significantly with the increase in the total number of simulations in the ensemble. The execution of the simulations in variable-sized fractions utilizes the versatility of the available workers. Hence, it is possible to further optimize the operating cost by choosing cost-efficient workers in terms of the simulations.

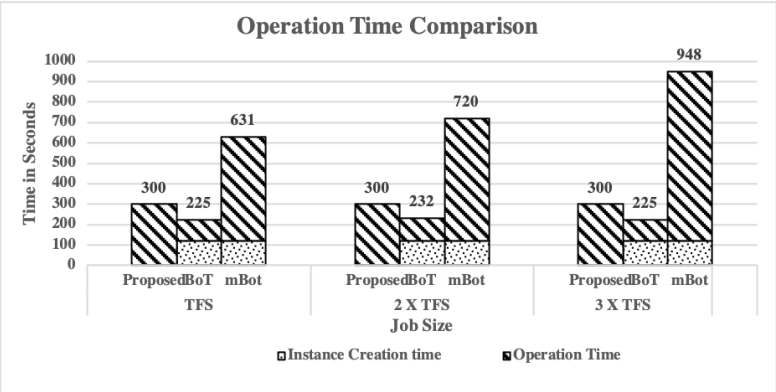


Figure 3.12: Operation time for different execution methods

Figure 3.12 shows the comparison of time performance between the two systems along with conventional bag-of-tasks execution. The simulations in the ensemble, when considered independent and run over as many workers as the number of simulations, produce the outputs in less time but incurs high cost. The operation time is variable in the modified bag-of-tasks execution, which assumes the equal size of the batch while the operation time in the proposed system is dependent upon the user requirements. When the size of the job increases, the modified bag-of-tasks type execution takes more time as shown in Figure 3.12 (For $3 \times TFS$ job, the operation time is about 176% more than the proposed system). For the urgent user requests, the system does not have to consider the additional time for the creation of the new instances. In contrast, for similar bag-of-task executions, there is always the time of creating new instances added in the total operation time. Consequently, as shown in Figure 3.12, the total operation time of urgent requests in the proposed system is always less than that of the modified bag-of-tasks type executions. Moreover, for other urgency levels of the requests (medium and low), the proposed system solves the trade-off between the time and cost by minimizing the cost to the maximum possible extent.

3.5.3 Cost Reduction using Resource Handler

Resource Handler in the proposed system minimizes the operation cost by intelligently choosing the cost-effective instances based on the urgency levels calculated for each user request. The on-demand instances offer higher reliability as these instances are dedicated to the user request once allocated, until the completion of the subjob. The spot instances provide cheaper options for execution, but the reliability offered by these

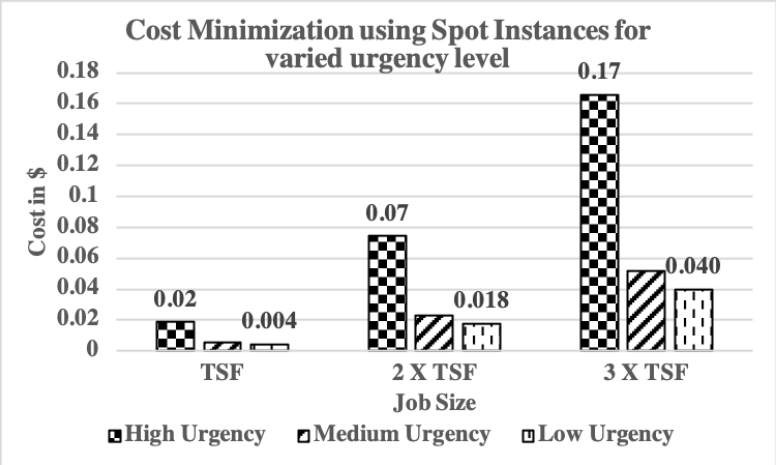


Figure 3.13: Cost Minimization using Spot instances

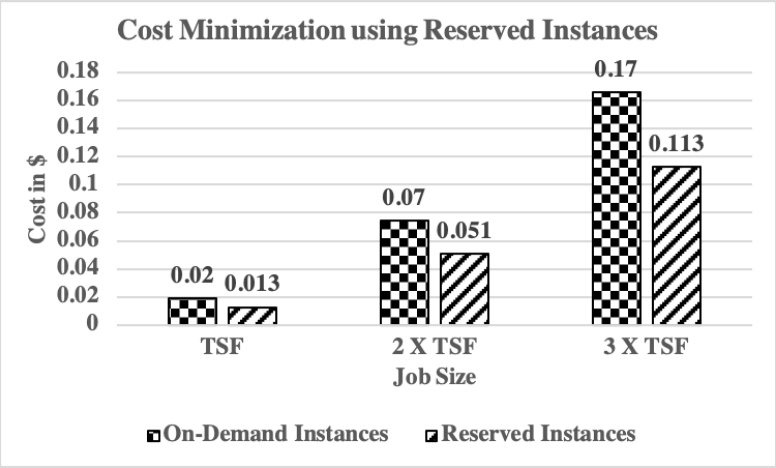


Figure 3.14: Cost Minimization using Reserved instances

instances is lower. The spot instances are offered to the other users with a higher bid in the Cloud environment, even if the current execution of the subjob is not complete. In this thesis, the bid prices for the medium and low urgency levels are derived from the historical information issued by different Cloud providers. It is to be noted that the calculation of bid amounts to ensure high reliability is not the aim of this thesis.

Figure 3.13 reflects the possible minimization of the operating cost by deploying the ensembles on spot instances rather than on on-demand instances whenever possible. The user requests with high urgency level were executed on on-demand instances, while those with medium and low urgency levels are executed on spot instances with different bid amounts. For the user requests with low urgency level, the users can minimize the cost up to 73% compared to the requests with the high urgency level. For the requests with the medium urgency level, the cost minimization is up to 76%. This cost

minimization is possible due to the trade-off between the reliability and operation cost of the instances. If the proposed system has to abandon the spot instances to other users in the Cloud environment because of higher bids, the system adds those user requests into the queue with altered urgency levels. The recovery and fault-tolerance techniques can ensure the execution of the subjobs getting resumed from the point where they were interrupted, but these techniques are beyond the scope of this work. If medium and low urgent labeled requests fail in the first round, the incurred operation cost is likely to increase. To overcome this cost discrepancy, a cost model that calculates the operating cost based on the request complexity and user deadline can be introduced.

Moreover, Capacity Planner in Resource Handler keeps track of the urgent user requests received at the system based on the time. The tracker assigns "peak" label to the duration based on the historical records. The proposed system reserves the Cloud instances in advance for the peak duration, which can further minimize the cost for requests with high urgency levels. The experimental results show that the cost for the user requests with high urgency levels were minimized by about 32% as depicted in Figure 3.14.

3.5.4 Cost Vs Levels of Deadline

The Ensemble Service cost generally increases with an increased level of urgency in the user requests. The total cost of operation is calculated based on the actual time for which the workers were in operation. The cost incurred in the proposed system for different levels of user-defined deadlines is less than the cost incurred in the bag-of-tasks system and close to the cost incurred by an on-premise system with the cheapest machines. The increase in urgency level incurs higher operation cost (see Figure 3.13). The urgent requests have higher cost and higher reliability as the reliability is traded against the cost. When compared to urgent request, the medium and low urgency incur up to 69% and 73% lesser operation cost. Moreover, the medium urgency incurs about 28% more operation cost compared to low urgency based on different user request complexities. For the requests with medium and low urgent level (based on the values of UL_k), the Ensemble Service cost increases if the service of the spot instances allocated for them is abruptly interrupted by the Cloud provider because of higher bids from other users (not in the proposed system). In the worst case, the user request with a low urgency level can incur the same cost as the request with a high urgency level. This cost discrepancy

can be solved by developing a cost model that charges the requests based on the urgency level and the job complexities.

3.5.5 Cost Vs Complexity of User Requests

The total number of simulations in the user requests can be altered by changing the size of the grid for the wildfire simulation in TFS samples. The operating cost for user request increases with increase in the grid size in the ensemble configuration, which ultimately increases the total number of simulations in the user request. Even for the varied number of total simulations, the proposed system design yielded minimal operating cost which are always less than the cost incurred by the bag-of-tasks execution. The operation cost is close to the cost incurred by the on-premise system with the cheapest machines.

3.5.6 Analysis of Time Performance under Multiple Urgent User Requests

To validate the support of multiple simultaneous users, we considered several simultaneous users submitting the requests (TFS configuration) roughly at the same time. The experiment was conducted with a maximum number of 150 VCPUs in the Cloud environment. Consequently, when more than seven urgent requests are received in the system, the variable batches of three service requests have to wait in the queue. In theory, Cloud infrastructure with a large number of computing nodes would not have any limitation. In the proposed system, when the urgent requests have to wait, the value of UL_k for each request becomes less than one and would otherwise be rejected as infeasible request failing to meet the deadline. For this analysis, we consider waiting time for the requests unable to find free resources. The waiting time contributes to the total time required for serving the requests. For 10 and 15 simultaneous urgent requests, the maximum time taken for serving the requests were 592 and 878 seconds, respectively (as shown in Figure 3.15, including the waiting time in the queue. Nevertheless, the actual time for which the simulations were run is comparable to the time taken to serve at most seven simultaneous service requests. The total cost calculation does not consider the waiting time. Consequently, the user requests with same complexity with similar deadlines have comparable operation cost. This limitation which requires the requests to wait in the

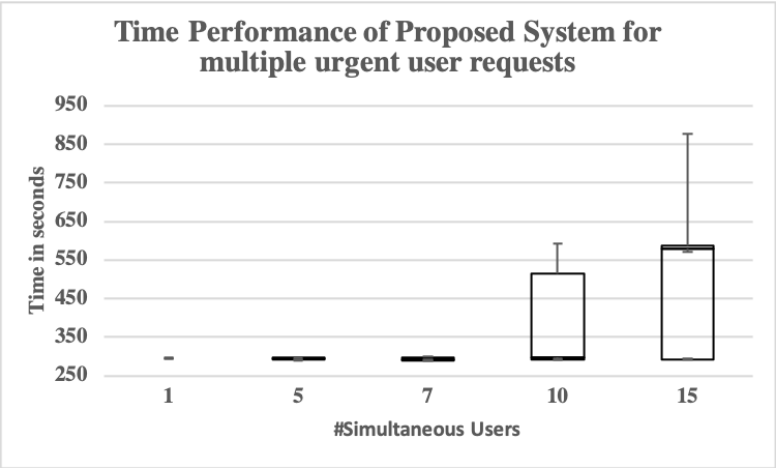


Figure 3.15: Time Performance Analysis under multiple simultaneous users with urgent deadlines (TFS Configuration with 169 simulations)

queue, in the proposed system, can be overcome by adding more computing nodes in the Cloud environment during the peak disaster season.

3.6 Related Works

Several studies have implemented geospatial models over the Cloud for different disaster management scenarios. Eriksson et al. [117] developed a simulator in Amazon EC2 Clouds to understand the outbreak of pandemic influenza over a particular place. Wan et al. [85] used Cloud infrastructure to classify the different occurrences of the flood into different levels based on severity and fatalities. The work done by Montgomery and Mundt [70] processed different geospatial data sets using a Cloud environment to predict the changes of the natural resources. The climate engine Huntington et al. [89] was developed using Cloud infrastructure to forecast the weather through climatological calculations and related statistical analyses. Pajorova and Hluchý [134] carried out complex Earth and astrophysics simulations using a Cloud environment. For wildfires, Kalabokidis et al. [173] highlighted the need for quantitative indices of wildfire behavior and effects with spatial layers of meteorological, vegetative, topographic and socioeconomic information for a holistic fire risk assessment of hazards and vulnerability. Kalabokidis et al. [71] proposed a web-based GIS platform called Virtual Fire using FARSITE [174] over the Cloud that offers various fire management related services. The study accommodated the fire propagation simulation in Virtual Fire, but the end-users

could not initiate fire behavior simulations for various technical and operational reasons. Kalabokidis et al. [175] explained how wildfire risk and spread simulation services could be offered as Software as a Service (SaaS) over the Cloud environment with more flexibility. Garg et al. [20] developed sparkCloud using Spark for wildfire prediction to demonstrate the capability of Cloud Computing to support different natural disaster models. However, the study focused on providing scalable solutions for running a wildfire propagation simulation within a Cloud environment based on user requirements without considering the ensemble with a large number of simulations.

Huang et al. [21] verified the capability of Cloud Computing to support ensemble simulations by deploying a complex dust forecasting model on an Amazon EC2 foundation with reduced cost when compared to using local resources. Li et al. [22] described a Model as a Service (MaaS) framework to support ensemble simulations of different Geoscience models over the Cloud infrastructure. Moreover, a cyberinfrastructure based system developed by Behzad et al. [23] detailed the implementation of ensemble simulation of groundwater system modeling over the Cloud environment provided by Microsoft Windows Azure Cloud Platform. These works have validated the readiness of Cloud infrastructure to support the complex ensemble simulations of different Geoscience models. However, fewer developments have been made to offer these models as end services to the users. Cost and resource optimization for ensemble simulations of natural disasters models over the Cloud environment have not, to our knowledge, been previously considered. Moreover, there are not any well-defined mechanisms to initiate and automate the multiple runs of simulations with minimal user interventions (a single user request) for an ensemble of disaster simulations.

The execution of simulations in an ensemble is conceptually similar to the execution of tasks in a bag-of-tasks application. These well-studied applications deal with a large number of independent tasks which can be executed in any order on any computational resource. However, for disaster models executing the simulations in variable batches, rather than as independent units, can significantly enhance the overall performance due to the large sizes of the input data sets, the sharing of intermediate data sets between different simulations and the specific geospatial requirements of the models. As highlighted in work [176], Cloud Computing has been widely adopted for bag-of-task applications due to flexibility in resource provisioning and on-demand pricing models. The optimization of the cost and the resource usage is focused on different perspectives of data centers

and the users [177]. There are different frameworks proposed in different works [178], [179], [180] where user-defined requirements, bandwidth and storage constraints and monetary cost are considered while executing the bag-of-task applications. These existing frameworks and mechanisms may not ensure reduced operational cost for ensembles of simulations as end services, and this is where the extension of the existing optimization schemes is required. Moreover, so far, the task clustering (creation of batches) has been done based on user requirements (time and budget) [172] [181], bandwidth [182] [183] and resource constraints [184]. For the ensembles of disaster simulations, each simulation is both compute and data-intensive. Thus, the creation of batches of simulations based on the most effective operation regions of the machines for user requirements can be more efficient. The estimation of resources required to execute the requests can also be helpful. As such, this study considers the unique features of disaster models and simulations to schedule the simulations in an ensemble to offer such functionalities as end services with minimal cost and resources. This study also considers the capacity planning and different pricing models of Cloud instances.

3.7 Summary

This chapter investigated the implementation of a cloud-based framework to offer the ensembles of disaster simulations as end services for rapid risk estimation. The proposed framework with the help of scalable Cloud resources was able to support compute-intensive ensembles in a convenient and time-efficient manner, which may take several hours to days in a conventional on-premise system with a small pool of computers. Additionally, the validation results are quite promising with an operating cost comparable to conventional and cheapest on-premise setup and up to 300% when compared to bag-of-tasks type execution. The next chapter discusses the extension of the Cloud-based framework to support the sensitivity analysis of operational fire simulations for rapid risk analysis.

Chapter 4

Sensitivity Analysis of Natural Hazard Models for Rapid Risk Analysis

In this chapter, to achieve rapid risk analysis, we first demonstrate how risk analysis is possible with sensitivity analysis (SA) and present comparative analysis of different existing SA methods and then propose a Cloud-based framework for sensitivity analysis of wildfire simulations for rapid risk analysis. In the first part, we apply two SA methods to empirical fire spread models recommended for operational use in Australian vegetation (AFDRS) to measure the sensitivity of fire spread rate to input conditions for risk analysis. Additionally, we present an analytical comparison of four different popular sensitivity analysis methods applied to two empirical fire models (Dry Eucalypt and Rothermel models). The parameters and their ranges chosen for the analysis have been adapted either to draw closer comparisons between fire models from different regions (Dry Eucalypt and Rothermel models) or to cover all operational fire conditions (Australian wildfire models). In the second part, we explain the extension of our generic Cloud-framework to support the computationally intensive task of performing sensitivity analysis of operational fire models in a convenient and time-efficient manner for rapid risk analysis. The efficacy of the framework is demonstrated by the scalability achieved while running large-scale wildfire simulations. We present a comprehensive sensitivity analysis of the input parameters used in the fire simulations for rapid risk analysis in

an operational environment. The ability to efficiently perform sensitivity analysis using the framework could allow such analysis to be performed as an on-demand service for operational disaster management.

This chapter is derived from the following works.
KC, U., Sullivan A., Hilton, J., Plucinski M., Garg, S., & Aryal, J. (2021). Assessing the sensitivity of Australian operational wildfire spread models. International Journal of Wildland Fire, (Under Review).
KC, U., Aryal, J., Garg, S., & Hilton, J. (2021). Global sensitivity analysis for uncertainty quantification in fire spread models. Environmental Modelling & Software, 143, 105110
KC, U., Garg, S., Hilton, J., & Aryal, J. (2020). A cloud-based framework for sensitivity analysis of natural hazard models. Environmental Modelling & Software, 134, 104800.

4.1 Risk Analysis and Sensitivity Analysis

In this section, we demonstrate how risk analysis can be achieved through sensitivity analysis of wildfire models and present how different existing SA methods should be chosen through a comparative analysis of the methods based on several factors.

4.1.1 Risk Analysis

Understanding wildfire behaviour, especially the rate-of-spread of the fire, is crucial for operational management during an ongoing wildfire, and risk mitigation and planning as such information can be used for prescribed burning, wildfire suppression, and issuing public warnings [165, 185–187]. Fire behaviour varies significantly between different vegetation types and many studies have examined the behaviour of wildfires in Australian fuels [188–191]. Recently, Mathews et al. [192] identified and described seven different empirical fire spread models for national use in the Australian Fire Danger Rating System (AFDRS). These models are the CSIRO Grassland [193], Dry Eucalypt [188], Buttongrass [194], Temperate Shrubland [195], Spinifex [196], Semi-arid Mallee-Heath [197], and Pine Plantation models [198]. Each of these models uses input parameters to represent weather and fuel conditions and provides a resulting rate-of-spread. However,

there are uncertainties associated with each of these input parameters that can influence the resulting rate-of-spread. As highlighted in the same report, quantifying the sensitivity of these input parameters to the resulting output and treating the associated uncertainties is useful for worst-case scenario analysis when applying these models in operational, or risk management scenarios such as fire danger rating system.

For Australian vegetation types, many specific models have been developed for a long period of time [188, 193–196, 199–201]. Cruz et al. [202] presented a comprehensive review of 22 fire spread models along with their applicability for prescribed burning and wildfire management. The model form and behaviour were discussed as well as the mathematical equations, model evaluation and the main input variables along with their influence on the fire spread rate. Additionally, the report identified and recommended the models that represent best practice for the operational and scientific prediction of fire spread in major vegetation types. These were the models defined by Cheney et al. [193], and Burrows et al. [196] for continuous Grasslands, Cheney et al. [188], Marsden-Smedley and Catchpole [194], Anderson et al. [195], Cruz [203], and Cruz et al. [197] for Shrublands, McArthur McArthur [199], Sneeuwjagt and Peet [204], Cheney et al. [205], and Cheney et al. [188] for Eucalypt forests, Byrne [206], Hunt and Crock [207], and Cruz et al. [198] for pine plantations. In the AFDRS research prototype report Mathews et al. [192] conducted a brief sensitivity analysis (based on 1000 runs with randomly selected conditions) of the fire spread models used in the rating system using relative sensitivity score [208] quantifying the proportional response of the model to changes in a perturbed input parameter. Such analyses explain the relative change in the output caused by the perturbation in the input, but cannot explain the changes caused by non-linear interactions between two or more input parameters. Despite being helpful for identifying the parameters with the greatest influence in the models, such analyses cannot explain all the uncertainties associated with the model parameters. Nonetheless, parametric uncertainty quantification would offer better understanding of the fire behaviour and provide pivotal information for effective prescribed burning, fire suppression, and operational management. As such, to fill the gap by precisely defining the extent of the contribution of different input parameters in the uncertainty of fire spread rate, we apply two different widely popular GSA methods (Morris and Sobol) to operational fire spread models.

4.1.1.1 Fire Models

We consider six different fire spread models that are recommended for the prediction of wildfire spread in major Australian vegetation types [190]. The models are: the CSIRO Grassland model [193], the Dry Eucalypt model [188], the Buttongrass model [194], the Spinifex model [196], the Semi-arid Mallee Heath model [197], and the Temperate Shrubland model [195]. The Pine Plantation Pyrometrics (PPPY) model is not included in this study as it is a complex iterative model involving several other models at different iterations with more than 100 sets of equations [202]. These fire spread models predict the rate of spread (ROS) of the headfire using different input parameters that represent weather conditions and fuel loads. We consider the three pasture conditions (undisturbed natural (uncut), cut and grazed (cut) and eaten out (grazed)) for the Grassland model, while surface fire spread rate and crown fire spread rate are considered as model output for the Semi-arid Mallee Heath model. Detailed information on the mathematical relationships in the models can be found in [190].

Based on the details of the fire models, we consider different relevant parameters that can be measured directly. The unit and ranges used to generate samples assigned to the parameters are taken in reference to the ranges used in [190] and listed in Table 4.1. The ranges are considered to represent the realistic environmental condition over which the model should be used. We use the uniform distribution to generate the samples for all the input parameters.

4.1.1.2 Sensitivity Indices Estimation

We used 10,000 samples of each input parameter within the given range be generated using sampling. Such a large number of samples ensures that the estimated sensitivity indices have converged [209] and the subsequent interpretation is a true quantification of the output uncertainties. The SALib [210] Python library was used to estimate the sensitivity indices for each of the methods.

4.1.1.3 Results

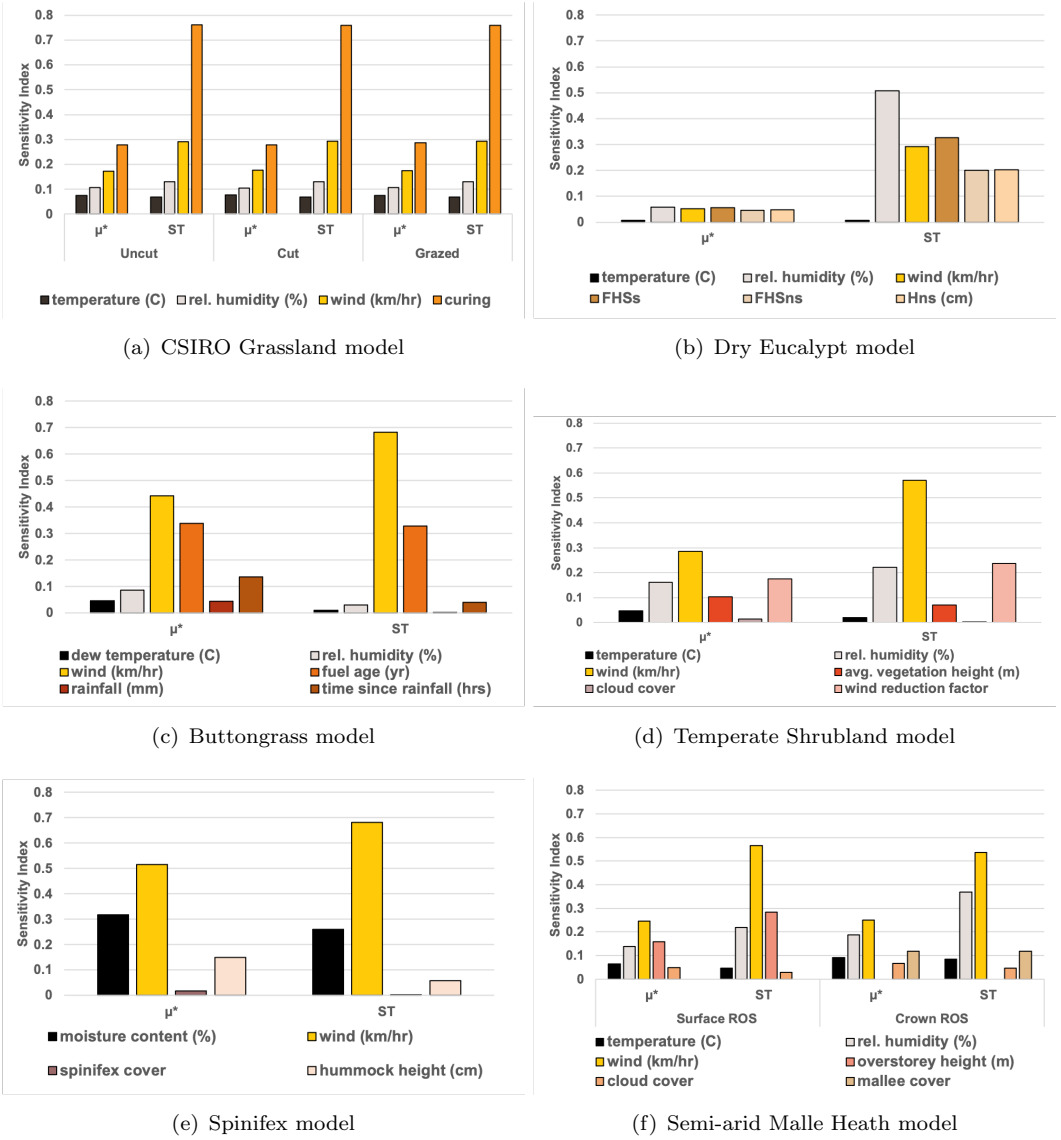
Figure 4.1 represents the values of sensitivity indices estimated for different fire spread models using Morris and Sobol methods while Figure 4.2 shows the extent of the pairwise

Table 4.1: Input parameters in different fire spread models. The acronyms are defined as adapted in the AFDRS) to be valid for the section only.

Parameters	**Unit**	**Range**
Common Parameters		
Temperature	^oC	[10 - 45]
Relative Humidity	%	[5 - 90]
Wind Speed	kmh^{-1}	[5 - 70]
Grassland Model		
Curing	%	[10,100]
Dry Eucalypt Model		
Surface Fuel Hazard Score (FHSs)	-	[0,4]
Near-Surface Fuel Hazard Score (FHSns)	-	[0,4]
Near-Surface Fuel Height (Hns)	cm	[0 - 50]
Buttongrass Model		
Dew Point Temperature	^oC	[10 - 45]
Rainfall	mm	[0 - 100]
Time since rainfall	hr	[0 - 480]
Fuel Age	$years$	[0 - 35]
Temperate Shrubland Model		
Wind Reduction factor	-	[0.3,1]
Average Vegetation Height	m	[0.25 - 5]
Cloud Cover	-	[0 - 1]
Hummock-Spinifex Model		
Moisture Content	%	[5 - 30]
Spinifex Cover	%	[30 - 70]
Hummock height	cm	[10 - 100]
Semi-arid Mallee Heath Model		
Overstorey Height	m	[1 - 5]
Overstorey Mallee cover	%	[5 - 80]
Cloud Cover	-	[0 - 1]

interactions between the parameters in fire spread models.

For the CSIRO Grassland model, curing was found to have the highest influence on the fire spread rate for all the pasture conditions and temperature was found to have the least influence (Figure 4.1(a)). Curing accounted for about 60% of the variation in fire spread rate, while temperature accounted only 5% of the variation. The overall contributions of wind speed and relative humidity stood at 23% and 10% respectively as calculated using Sobol method. The order of the parameters in terms of the influence on the spread rate is the same in Morris method with similar trend in the non-linearities of the parameters. As can be seen in Figure 4.2(a), the combination of curing with wind and relative humidity has the highest influence on the spread rate for uncut pasture condition. Consequently, the fire grows rapidly under environmental conditions with



(a) CSIRO Grassland model

(b) Dry Eucalypt model

(c) Buttongrass model

(d) Temperate Shrubland model

(e) Spinifex model

(f) Semi-arid Malle Heath model

Figure 4.1: Estimated sensitivity indices for various fire spread models, μ^* is the mean elementary effect, while ST is the total sensitivity index. Higher values of these indices for a parameter represent a greater influence of the parameter in the fire spread rate.

high curing and high wind speed or high curing and low relative humidity irrespective of the pasture condition.

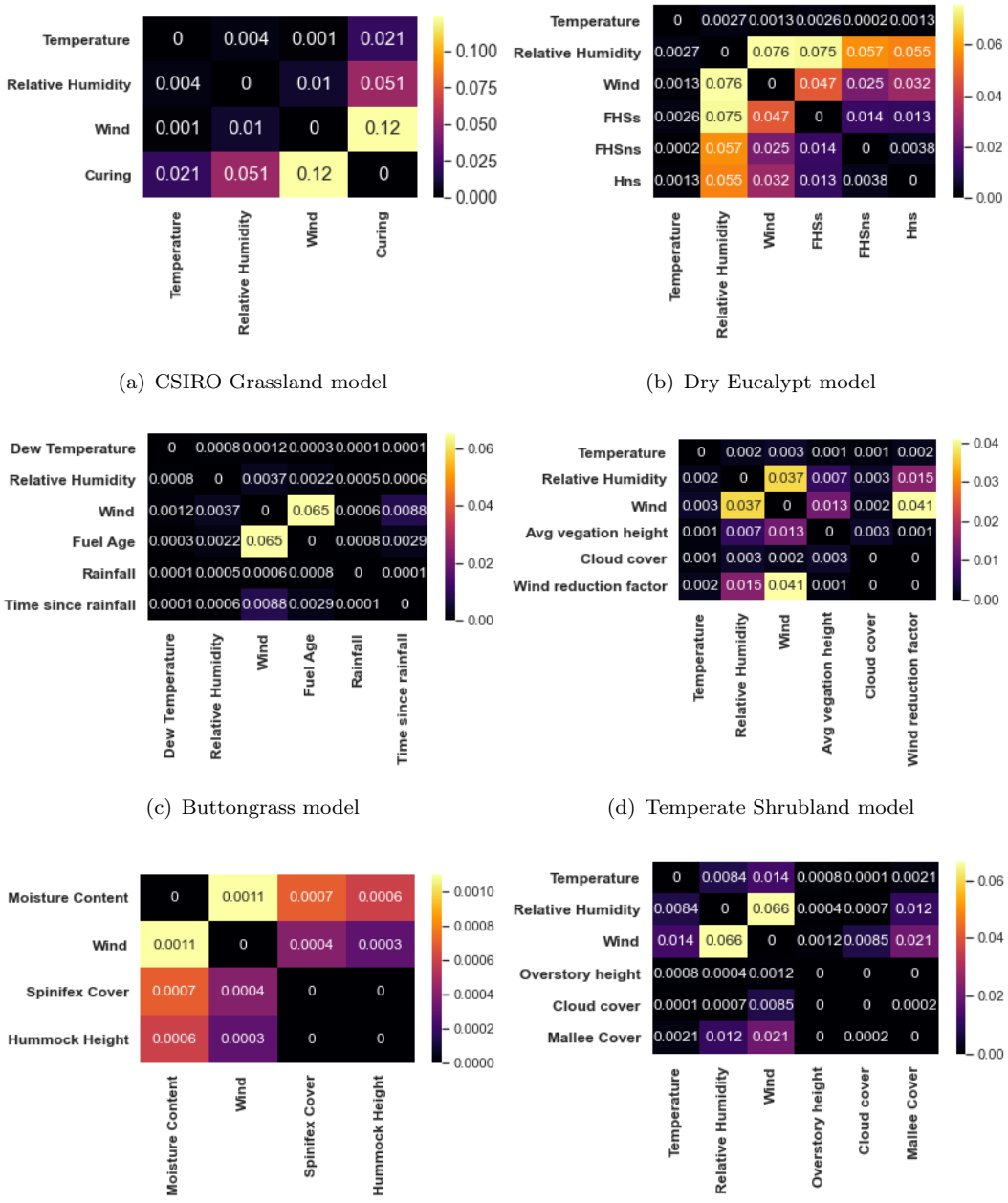
For the Dry Eucalypt model, the relative humidity had the highest influence on the model output variability, followed by surface Fuel Hazard Score (FHSs) and wind speed with temperature having the weakest influence (Figure 4.1(b)). Relative humidity had an overall contribution of 33%, while temperature had an overall contribution of less than 1%. The overall contribution of FHSs and wind speed stood at 22% and 19% respectively, while near-surface FHS (FHSns) and near-surface fuel height (Hns) also

have significant contribution of about 13% each. As can be seen in Figure 4.2(b), the second order interaction between relative humidity with wind is the pairwise parameter interaction in the model that highly influences the fire spread rate. Other notable pairwise interactions are between relative humidity with near-surface fuel hazard score and near-surface fuel height.

Wind speed was the most influential parameter for the Buttongrass model with an overall contribution of 63%, followed by fuel age at second rank with a contribution of 30% (Figure 4.1(c)). Dew point temperature and rainfall were found to have negligible influence with a combined contribution of less than 1%. Time since last rainfall had an overall contribution of 4% and Relative humidity had only a 3% contribution. As a general trend, the parameter with highest overall contribution is expected to have more non-linear interactions with other parameters. But, as estimated by the Morris method, fuel age had more non-linear interactions with other parameters when compared to that of the wind speed. The only notable pairwise interaction in the model is the interaction between wind and fuel age (see Figure 4.2(c)) thereby indicating the fact that the fire propagates quickly under stronger winds with older fuel age.

For the Temperate Shrubland model, wind speed had the highest influence on the fire spread rate, while cloud cover, which is only used in the selection of a fuel moisture model, had the least influence (Figure 4.1(d)). The wind reduction factor and relative humidity followed wind speed as the second and third ranked parameters based on the overall contribution. Wind speed, wind reduction factor and relative humidity were found to be the most significant parameters to the model with an overall combined contribution of 90% to the model output. Average vegetation height had a modest influence on the variation of fire spread rate at around 7%. Temperature and cloud cover accounted for less than 2% contribution to the output variation in the model. The interaction between wind and wind reduction factor in the model is the second-order interaction in the model with the highest influence in the fire spread rate hinting to the stronger influence of the wind. Other notable second-order interactions are the ones between wind and relative humidity, and wind and average vegetation height (see Figure 4.2(d)).

For the Spinifex model, wind speed and moisture content were found to influence almost all the variations in fire spread rate (Figure 4.1(e)). As the moisture content was



(a) CSIRO Grassland model

(b) Dry Eucalypt model

(c) Buttongrass model

(d) Temperate Shrubland model

(e) Spinifex model

(f) Semi-arid Malle Heath model

Figure 4.2: Pairwise parameter interaction influencing the fire spread rate (Uncut pasture condition for the Grassland model and surface ROS for the Semi-arid Mallee Heath model). Lighter shades for a parameter-pair in the color map represent the favourable combinations for extreme fires while the darker shades represent the combinations for low risk fires.

considered as a parameter which had to be measured directly as explained in the literature [196], we considered moisture content as a parameter for the model, which would otherwise be defined by the relationship between temperature and relative humidity. Moisture content accounted for about 50% of the variation in fire spread rate, while wind speed accounted for around 30% of the variation. Spinifex cover had about 20% contribution on the model output. Analyzing the second-order interactions within the model, the interaction between wind and moisture content is the most significant one with the highest influence in the fire spread rate. The pairwise interactions of moisture content with spinifex cover and hummock height also have noteworthy influence in the fire spread rate (see Figure 4.2(e)).

Wind speed was the most influential parameter in Semi-arid Mallee Heath model accounting for around 50% of the variation in surface fire spread rate (Figure 4.1(f)). For the same spread rate, overstorey height followed closely at second rank with an overall contribution of 25%. Relative humidity had an influence of 20% on the variation of surface fire spread rate, while the combined influence of temperature and cloud cover was less than 7%. As expected, mallee-cover did not have any influence in the surface fire spread rate. For crown fire spread rate, the contribution of all the parameters was similar. Wind speed contributed to 47% of the output variation followed by relative humidity and mallee-cover at 32% and 10% respectively. As expected, overstorey height had no influence in the crown fire spread rate. As can be seen in Figure 4.2(f), the pairwise interaction between wind and relative humidity has the highest influence on the surface fire spread rate as the fire can grow aggressively under stronger winds and low humidity. The interactions between wind and mallee cover and wind and temperature also have significant influence in the surface fire spread rate in the model.

In a nutshell, the temperature was found to be the parameters with the least influence on the fire spread rate, while fuel moisture, wind speed, and fuel characteristics were found to have a significant influence. The input factor combinations for which the fire risks are extreme are also clearly presented. The findings in our analysis qualitatively agree with previous experimental studies. Such findings, along with the quantitative sensitivity values, may contribute to the application of these models in risk management and operational contexts for risk analysis.

4.1.2 Comparative Analysis of SA Methods

Sensitivity Analysis (SA) is the study of the uncertainties in model output caused by the variation in the model inputs. In local SA, the impact of the parameters is studied around a local point [26]. In global sensitivity analysis (GSA), the entire range of the input parameters is taken into consideration while analyzing the model outputs. GSA methods are one of the most significant quantitative techniques in risk modeling and analysis [211–214]. GSA of natural hazard models can help identify the factor(s) or scenarios that pose a significant risk in an event of the outbreak of the disaster. Such identification can prioritize strategic plans for effective risk management [215]. For example, GSA of fire spread models can help the authorities identify adverse weather scenarios or conditions that may contribute to dangerous wildfires.

GSA has recently gained attention in environmental modeling in areas such as wildfire, hydrology, decomposition, and crops [27–30]. GSA helps to identify influential and non-influential factors in the model and fixing the non-influential factors to a known value, and the treatment of uncertainties that contribute to better understanding and interpretation of the model [25, 26]. There are several GSA methods in the literature. In the Morris method [216], the influence of a parameter is estimated by assessing the variation in the model output caused by varying values of the parameter within its entire range when other parameters are kept constant. Several works [15, 217–219] have conducted sensitivity analyses of different environmental models using the Morris method as it provides a good trade-off between the efficiency and accuracy for compute-intensive model [220]. The Morris method cannot explain the pair-wise interactions between the input factors for models with non-linear input-output relationships and cannot be used for non-orthogonal input factors (i.e. any correlated factors, as the correlation cannot be induced) [25, 221]. The Sobol method [222] and the FAST method (as proposed in [223]) are two widely used variance-based SA methods in environmental models [15, 25, 26, 224]. In the variance-based approach, the sensitivity of an uncertain input factor is estimated by investigating the factor's contribution to the model output. Variance-based methods give a good measure of the contribution made by the input factor and its interaction with other factors. For robust results, these methods require many runs of the model, which can be computationally costly if the number of model runs is

significantly high [225]. Furthermore, variance is not a sensible measure of model output uncertainty when the model has multi-modal or highly skewed output distribution [226]. The PAWN method [29] is a density-based method that estimates the sensitivity indices based on the density function of the model output. Applications of density-based approach include HydMod model [29], fire spread modeling [227], probabilistic risk assessment model [226] and engineering design system [228]. The density-based approach can overcome the limitations of other approaches but, its adaptation has been fairly limited as it is difficult to implement as one requires the knowledge of the conditional PDFs of the input factors [29].

The existing literature details several GSA methods and instances where they are applied to fire spread models. An uncertainty analysis study was carried out on wildfire models of boreal forests using Morris, Monte Carlo, and first-order analysis methods in [15]. The SPITFIRE fire model [229] was studied in a similar study where the Morris and the Sobol methods were used by Gomez-Dans [230] to study different forests (Boreal, Savanna, Temperate, and Tropical). A global sensitivity analysis of the Rothermel model over the Mediterranean region was done by Salvador et al. [231] which established low heat content, particle density, and mineral content as the parameters with negligible influence on fire spread rate. Moreover, Liu et al. [16] used variance-based methods to reduce the number of parameters in the Rothermel model (Chaparral fuel model) and used quasi-Monte Carlo methods for parametric uncertainties quantification in the reduced model.

The choice of the approach used in these studies was dependent on the compromise between accuracy, computational cost, and objectives. There are a few studies [232–234] that have presented comparative analyses of different GSA methods. In this part, we expand the scope of such analyses to the wildfire domain by presenting a comprehensive comparative analysis of various GSA methods applied to fire spread models. Given the inherent dangers, wildfires pose to lives and infrastructure, determining the intrinsic uncertainties of wildfire models is crucial for their use in operational wildfire management. As such, we draw a clear picture of four different GSA methods (Morris, Sobol, FAST, and PAWN) applied to two different fire models: the Dry Eucalypt model [188] (used mainly in Australia) and Rothermel [235] (used widely in North America). The choice of fire spread models has been made based on the number of the model parameters to draw a clearer picture of the comparative analysis between different GSA methods applied to

models with different numbers of parameters. We further discuss the implications of the findings of the analysis on the model uses and optimization through factor fixing, prioritization, and uncertainties treatment. We also present an investigative analysis of all four methods applied to fire models for an ability to guide the use of the model and treat different kinds of uncertainties in such models.

4.1.2.1 Wildfire Models

We consider the Dry Eucalypt model and the Rothermel model for this work. The Dry Eucalypt model is widely used for Australian eucalypt forests. The model was developed from a sequence of experiments called 'project Vesta' [188] carried out in south-western Australia, aimed at updating an older model for the fuel type [236].

The Rothermel wildfire model, which is widely used in North America, was developed by Rothermel in 1972 based on the principle of conservation of energy and experimental tests carried out with different fuel models in the US [235]. The model describes the fire behavior in terms of the rate of spread, flame length, and intensity. The fuel models are used to define the fuel input parameters, while dynamic fuel models and other models are used to define live fuel curing and the effects of cross-slope wind in fire spread. In this part, we adapt the model as described in [237] and used the mathematical equations, input parameters, and their distributions accordingly for the analysis.

4.1.2.2 Parameter Selection

Based on the working of the Dry Eucalypt fire model as described in the work of [188], temperature, relative humidity, fuel age, and wind, as listed in Table 4.2 are selected as input parameters for the sensitivity analysis. These parameters are considered to be the major input parameters of the fire model by wildfire communities as well. Fuel age was selected as the fuel descriptor parameter as it closely resembles the fuel parameters used in the Rothermel model widely used in North America. For the Rothermel fire spread model, the parameters and their distributions, as listed in Table 4.3 were chosen based on mathematical equations and setup of the work of [237].

4.1.2.3 Determination of Input Parameter Distribution Function

To define the range for each input parameter in the Dry eucalypt model, we considered the ranges detailed in [238]. For example, the experimental fires were carried under the temperature range $21°C - 32.5°C$, but the overall applicable temperature range for the model is $10 - 40°C$, and this latter range is used in the part . For this analysis, we assigned uniform distributions to all the parameters to account for the variation within the range as shown in Table 4.2. The range and the distribution assigned to each input parameter could easily be changed during the analyses if required. For the Rothermel fire spread model, we chose the ranges and distributions (Table 4.3) as defined in [237] for testing the estimated Shapley effects. For the parameter *slope*, the tangent of the angle of steepness (in degrees (^{o})) is considered as the input as per the experimental design in the same work and the input includes the values of the angle of slope up to about 39°.

Table 4.2: Probability distribution functions (PDFs) of Input Parameters for Dry Eucalypt Fire spread models. The parameters for the uniform distribution are minimum and maximum values respectively. The acronyms here are adapted to have same symbols for similar parameters in two different models and are valid for the section only.

Symbol	Parameters	Units	Distribution	Remarks
T	Temperature	^oC	uniform(10,40)	[10, 40]
RH	Relative Humidity	%	uniform(10,90)	[10,90]
U	Wind Speed	km/hr	uniform(10,60)	[10,60]
FA	Fuel Age	yr	uniform(0,35)	[0, 35]

Table 4.3: Probability distribution functions (PDFs) of Input Parameters for Rothermel Fire spread model. The parameters for lognormal and normal distribution are mean and standard deviation respectively. The acronyms here are adapted to have same symbols for similar parameters in two different models and are valid for the section only.

Symbol	Parameters	Units	Distribution	Remarks
fd	Fuel depth	ft	lognormal(2.19,0.517)	
$a2v$	Fuel particle area to volume ratio	ft^{-1}	lognormal(3.31, 0.294)	>5
h	Fuel particle low heat content	btu/lb	lognormal(8.48,0.063)	
od	Oven-dry particle density	lb/ft^3	lognormal(-0.592,0.219)	
ml	Moisture content of live fuel		normal(1.18,0.377)	>0
md	Moisture content of dead fuel		normal(0.19,0.047)	
mc	Fuel particle total mineral content		normal(0.049, 0.011)	>0
U	Wind speed at midflame height	ft/min	lognormal(2.9534,0.5569)	
tp	Slope		normal(0.38,0.186)	>0
P	Dead fuel loading to total fuel loading		lognormal(-2.19,0.66)	<1

4.1.2.4 Calculation of Sensitivity Indices

The sample generation and model evaluations of empirical fire models for Morris, Sobol, and FAST methods are carried out in Python using SALib [210]. SALib is a library in Python that supports different sensitivity analysis methods. The sensitivity indices are also calculated using the library for different values of sample size, which give a different number of total samples for different SA methods. For the PAWN method, a Matlab-based SAFE toolbox [239] was used for sample generation, CDF generation, model runs, and calculation of the indices through statistical measures. For this, we consider "maximum" as the statistic to calculate the PAWN indices for the input parameters in the fire models as used in [29].

4.1.2.5 Results and Discussions

In this section, we present the results obtained from the sensitivity analyses of the fire models and discuss the implications of the values of sensitivity indices on the fire models.

Convergence of SA indices In our convergence analysis of SA indices, we tested the convergence more intuitively based on the ranking, values, and screening of μ for Morris, Total Effect for Sobol (ST), and FAST ($Total$), and PAWN indices for PAWN).

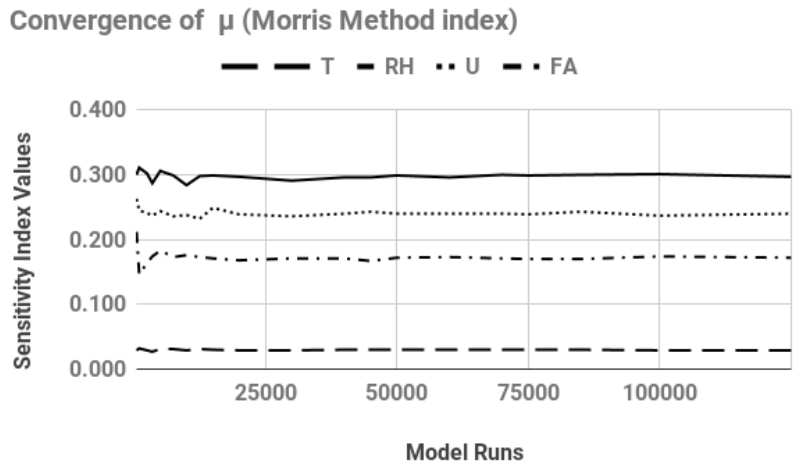
Figures 4.3-4.6 represent the values of SA indices calculated using different SA methods for varying the number of model runs. The rank (order of the parameters with the highest to the lowest impact on fire spread rate) of the input parameters in terms of their effects on the model outputs is consistent for all the methods in the Dry Eucalypt model. For the condition of convergence based on the values of the indices, we follow the maximum difference between the indices calculated in successive model runs, which should be less than the threshold of 0.05 as defined by Sarrazin et al. [209]. In our analysis, the Morris method took at least 44000 model runs to converge (25,000 for Dry Eucalypt and 44,000 for Rothermel), as shown in Figure 4.3, while the Sobol method took 110,000 for convergence (50,000 for Dry Eucalypt and 110,000 for Rothermel), as shown in Figure 4.4. The FAST method took at least 220,000 model runs (100,000 for Dry eucalypt and 220,000 for the Rothermel) as shown in Figure 4.5) to converge. The PAWn method took at least 33,000 (22,000 for Dry eucalypt and 33,000 for Rothermel)

to converge. For lesser model runs, the rank and value of the indices for the input parameters kept changing beyond the limit of the threshold required for the convergence.

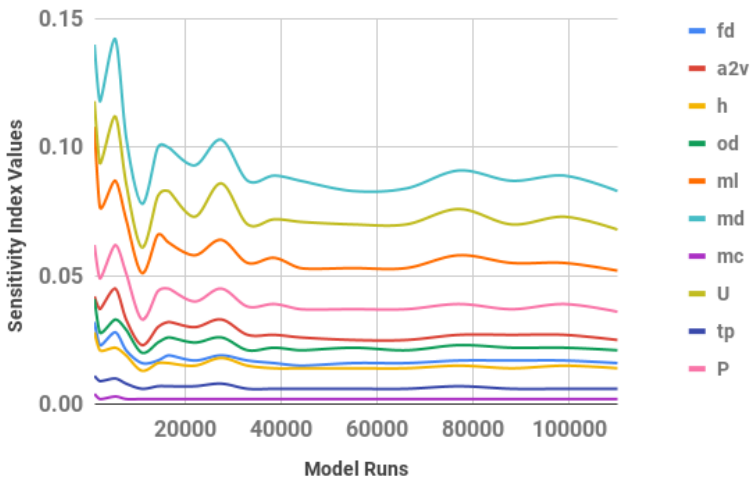
The gap between the most sensitive and least sensitive parameter has remained consistent for the Morris and Sobol methods beyond 44,000 and 110,000 model runs respectively, while the same for the FAST method is beyond 220,000 model runs. For the PAWN method, the indices converged at relatively lesser model runs. After 22000 model runs, the value of PAWN indices for all the parameters in the Dry Eucalypt model. The rank of the parameters based on their influence only marginally changed before 6600 model runs, after which the rank remained constant throughout the analysis, as shown in Figure 4.6(a). For the Rothermel model, the rank of least significant parameters changed until 33000 model runs, after which the rank started staying consistent. For the least influential parameters, the values of PAWN indexes seem to decrease with an increase in the model runs, but the difference is well within the threshold of 0.05.

The distance between the values of indices for the most and least significant parameters stayed consistent beyond the model run of 33000 for all the models. Based on these experimental findings, it can be concluded that the minimum number of the model runs required to produce robust results for the sensitivity analysis of fire spread models vary based on both the SA method chosen as well as the wildfire spread model under consideration. The PAWN method takes fewer model runs to converge when compared to variance-based (Sobol and FAST) and the Morris methods.

Robustness Check Figure 4.7 shows 95% confidence interval (CI) of the sensitivity indices calculated using different GSA methods. The indices calculated in the Morris method have a narrow confidence interval in all the fire spread models. The width of the CI is proportional to the values of the indices calculated for each parameter for both models. For the Dry Eucalypt model, Morris and FAST methods were more robust when compared to Sobol and PAWN methods. The 95% CI is the widest for the relative humidity and the narrowest for the temperature. The maximum widths of 95% CI for the Sobol and PAWN stood at 0.024 and 0.048 respectively, when compared to 0.006 and 0.002 for the Morris and FAST methods respectively. For the Rothermel model, *md* had the widest 95% CI, while *mc* and *tp* had the narrowest width. The 95% CIs for the Morris and FAST methods were quite narrow (with a maximum width of ˜0.004) and thus, these methods can be labeled as robust. The Sobol method had a maximum

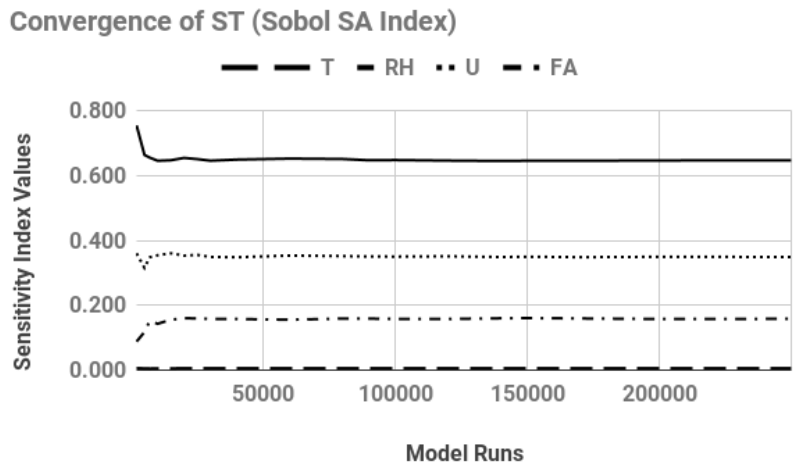


(a)

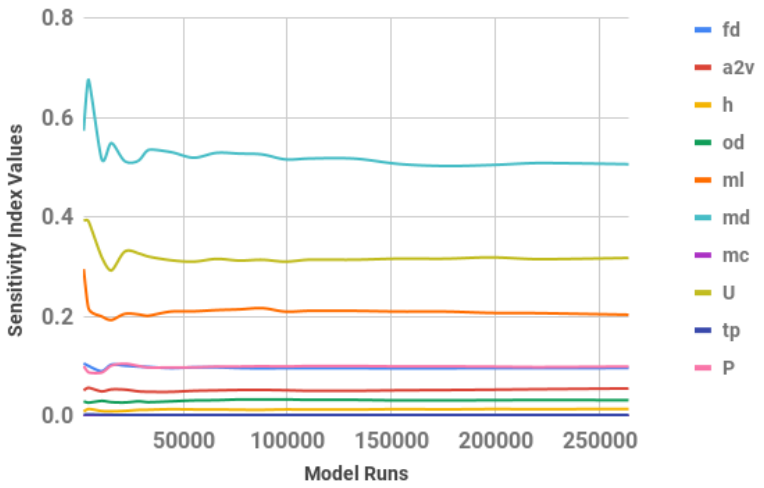


(b)

Figure 4.3: Convergence of Morris Index (mean of absolute elementary effects μ) (a) Dry Eucalypt Model (b) Rothermel Model. The values of μ start converging after 25000 model runs for the Dry Eucalypt model and after 44,000 model runs for the Rothermel model. The rank of input parameters (based on the relative impact on fire spread rate) has remained consistent over the entire analysis. The acronyms for the parameters are listed in Tables 4.2 and 4.3



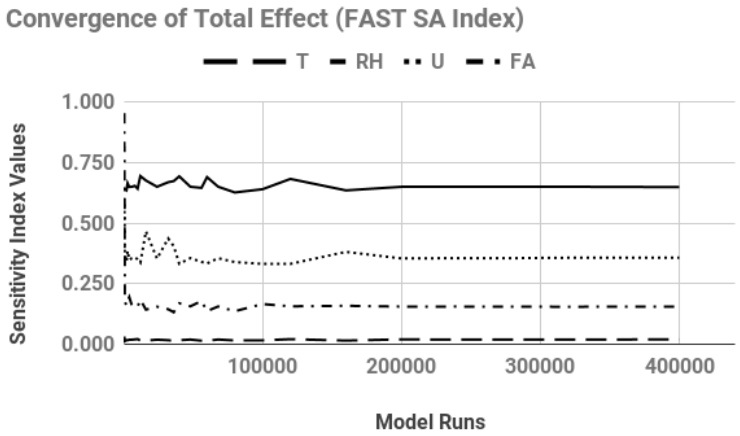
(a)



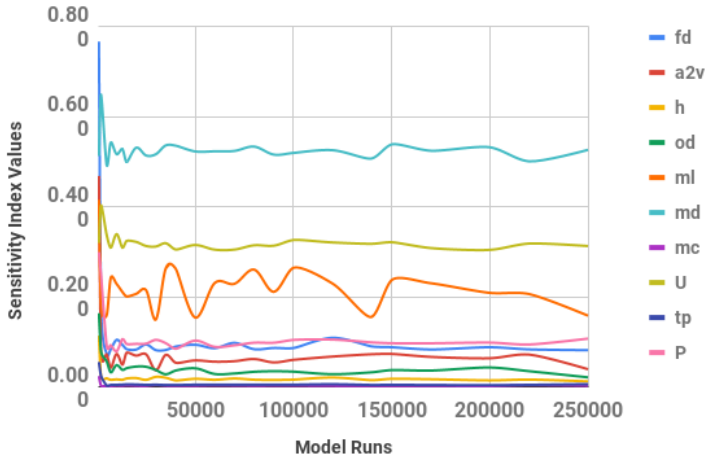
(b)

Figure 4.4: Convergence of Sobol Index (Total Effect ST) (a) Dry Eucalypt Model (b) Rothermel Model. The rank has changed a couple of times for the Rothermel model after the indexes start converging at 110,000 model runs. The indices start converging after 50,000 model runs for Dry Eucalypt models. The acronyms for the parameters are listed in Tables 4.3 and 4.2.

95% CI width of 0.028, while the same for the PAWN method was 0.035. It can thus be concluded that the OAT and variance-based approach are more robust than the density-based approach. To increase the robustness of the methods, the bootstrapping technique can be coupled together with convergence analysis, as the CIs become narrower with the increase in the number of model runs. Nevertheless, the trade-offs between the computational complexities of the increased model runs and the desired robustness have to be considered.



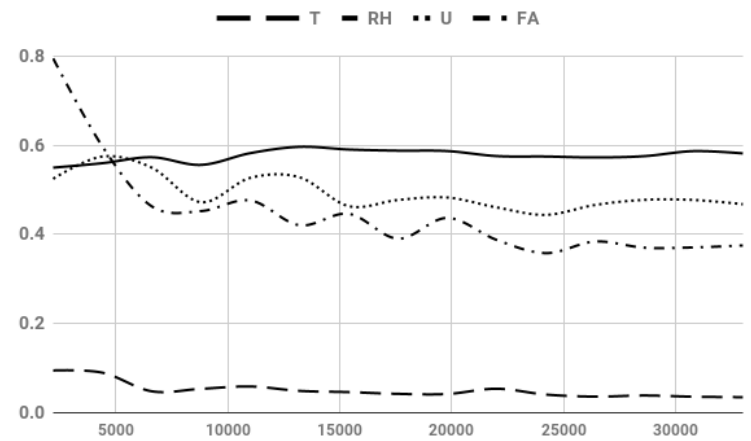
(a)



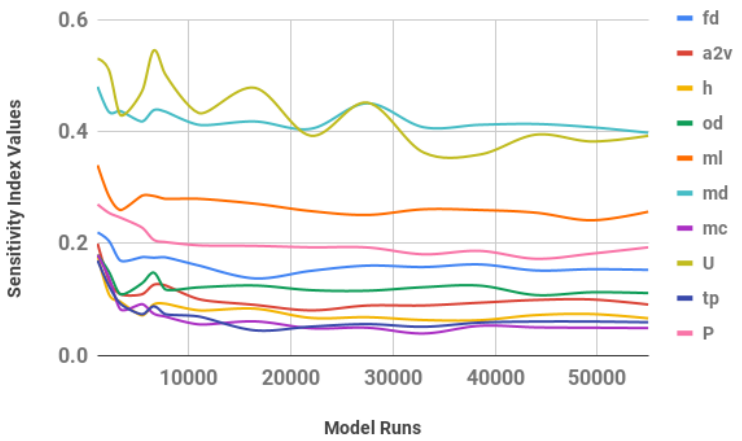
(b)

Figure 4.5: Convergence of FAST Index (Total Effect *Total*) (a) Dry Eucalypt Model (b) Rothermel Model. The values of indices keep fluctuating for model runs less than 100000, after which they start converging for both fire spread models. For the Rothermel model, the values of indexes kept fluctuating, influencing their rank only to converge after 220,000 model runs. The acronyms for the parameters are listed in Tables 4.2 and 4.3.

Our comparative analysis of different SA methods applied to fire spread models has consistent results. All of the four SA methods established relative humidity as the parameter with the highest influence and temperature as the parameter with the least influence on the fire spread rate in the Dry eucalypt model. Additionally, the rank of the parameters based on their influence on the model output is consistent with all the four SA methods. These findings align with the results we obtained in our previous work [28] where we conducted the sensitivity analysis of fire simulation tool - Spark [3]. In Spark, the Dry eucalypt model is one of the fire models considered within the framework to estimate the fire spread rate for eucalypt forests for determining the total



(a)

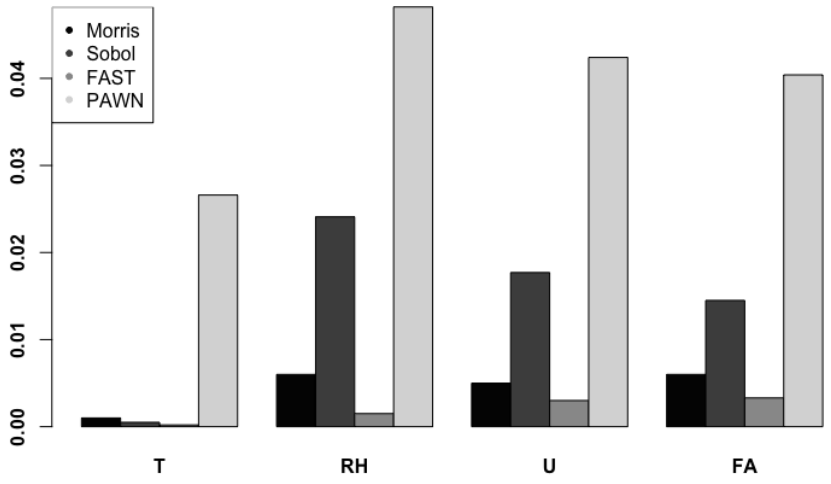


(b)

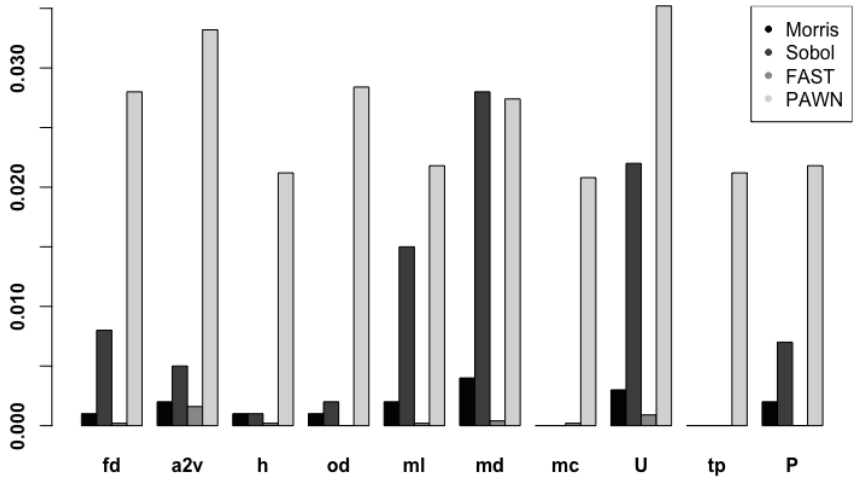
Figure 4.6: Convergence of PAWN Indices) (a) Dry Eucalypt Model (b) Rothermel Model. The rank of the parameters in terms of their relative impact on fire spread rate has remained consistent thereafter for the Dry Eucalypt Model after 6600 model runs, which is less compared to variance-based methods. The rank of least significant parameters kept changing for the Rothermel model until 33000 model runs after which the indexes start converging. The acronyms for the parameters are listed in Tables 4.2 and 4.3.

area burned by fire after a particular time. The consistency and the similarity of the results obtained to the previous findings in the Dry eucalypt model verify the correctness of our experimental setups.

For the Rothermel model, the estimation of SA indices as done in our experiment is consistent for the parameters with the highest and strongest influence on the fire spread rate. The three parameters namely moisture content of dead fuel (md), wind speed (U), and moisture content of live fuel (ml) are the top-ranked parameters based on their influence on the fire spread rate for all the four SA methods. Similarly, fuel particle



(a)



(b)

Figure 4.7: 95% confidence interval of the sensitivity indices estimated (a) Dry Eucalypt Model (b) Rothermel Model. Morris and FAST methods have narrower widths, which indicates the more robustness of the methods. The acronyms for the parameters are listed in Tables 4.2 and 4.3.

total mineral content (mc), slope (tp), and fuel particle low heat content (h) are the parameters found to have the least influence on the fire spread rate with consistent rank in all four SA methods. The influence of the parameters on the fire spread rate in the Rothermel model was studied in [237] using Shapley values [240] under three categories - independence, weak dependency, and strong dependency (between md and U indicating the fact that stronger the winds, drier the fuel gets). The parameters - mc, tp, and h were found to have the highest influence, and the parameters - tp and h had the weakest influence on the fire spread rate on all the cases. The influence of mc was found to decrease with the introduction of the dependency between md and U. These findings on the parameters with the highest and the lowest influence on the model output are consistent with the results obtained in our analysis. Interestingly, in the same work, od was found to be no Shapley effects thereby establishing the parameter as one of the least influential parameters. In our analysis, od was found to have some influence on the fire spread rate but with a rank in the bottom half (6 and 7) based on its influence on the fire spread rate, od can still be labeled as one of the least influential parameters. Our findings are also consistent with the results reported in [16, 231, 241, 242].

Under our objective to establish the suitability of SA methods based on several assessment factors, we performed convergence and robustness-check analyses. The results obtained from those analyses are interesting with implications on how sensitivity analysis should be applied to wildfire spread models. From our convergence analysis, it is clear that the PAWN method converges quickly compared to other SA methods. Similarly, the Morris method is also one of the computationally efficient methods when it comes to quick convergence. The FAST method took unusually long to converge, which could be due to interference between the frequencies considered in the algorithm used to estimate the indices. Our convergence analysis for the two fire spread models shows that despite the increase in the number of parameters, there is no change in the convergence patterns of the SA methods. Similarly, during our robustness check, the FAST and the Morris methods were found, in general, to be the most robust SA methods for converged indices. On the other hand, the PAWN and the Sobol methods were the least robust methods as the 95% CI for the indices were wider. As a general trend, the 95% CIs for highly influential parameters were found to be wider compared to the least influential parameters. One of the interesting findings in our analysis is the CIs of md where the CI with the Sobol method is wider than the CI with the PAWN method. These findings

did not follow the usual finding where the Sobol indices were found to be more robust than the PAWN indices. Thus, the robustness of SA methods may change when the number of parameters in fire models increases, and this fact should be considered while choosing the method for any analysis.

Figure 4.8 summarizes our findings in determining the suitability of SA methods while applying them to wildfire models. The suitability of SA methods was assessed under four factors namely robustness, convergence, the number of parameters (model runs required for base sample size), and details of sensitivity information. As can be seen in the figure, each assessment factor has a pecking order for the four SA methods. The FAST and the PAWN are the two methods to be prioritized for high robustness, while the PAWN and the Sobol and the Morris methods are the methods suitable for more details on the sensitivity information. Similarly, for more parameters in the model, based on the same base sample size, the PAWN and the FAST methods are suitable for the estimation of the sensitivity indices. For quicker convergence, the PAWN and the Morris methods are more suitable methods while the FAST is the least suitable method. The choice of the SA methods depends on a balanced trade-off between these assessment factors and such choice can be quickly made in reference to Figure 4.8. Nevertheless, the Morris method should be prioritized for initial parameter screening in wildfire models under limited computational resources as the method quickly estimates robust indices with fewer model runs. For the additional information on the influence of the second-order interactions between parameters on the model output for worst scenario analyses, the Sobol method should be prioritized where the computational resources do not pose to be a significant constraint.

The sensitivity analysis results as obtained in our work have further implications on how the fire models can be optimized for operational uses. The Dry eucalypt model can be further optimized by prioritizing relative humidity, wind, and fuel age while the complexity of the Rothermel model can be reduced by fixing the least influential parameters (mc, tp and h) to nominal values for operational uses. These findings as obtained during the sensitivity analysis can lead to new operational tools by cutting down the parameter space of the least influential models or dropping the least influential parameters.

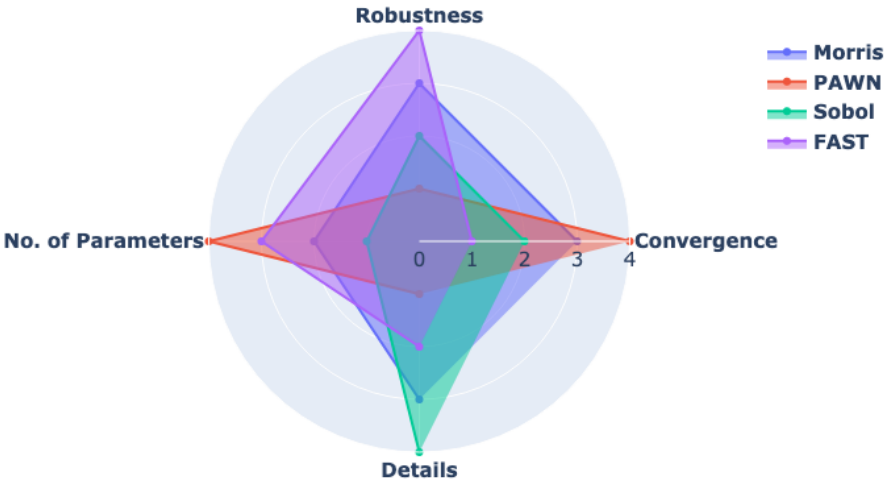


Figure 4.8: Suitability of SA methods based on their ranks for the factors (the number of parameter, robustness, and convergence and details of the sensitivity information). A Higher value of an SA method for a factor represents a better suitability of the method for that factor.

4.2 Cloud-based Framework for Sensitivity Analysis of Wildfire Models

Conventionally, SA analyzes the variability of deterministic model outputs produced by possible combinations of the input parameters [243]. Computational natural hazard models are characterized by different, often complex, mathematical relationships that must be calculated multiple times for each combination of input parameters to produce a set of outputs. As natural hazard models often require a large number of input parameters, accurate sensitivity analyses require a large number of combinations, making such analyses compute-intensive and time-consuming. These analyses can take several hours to days to complete for complex models. Such analyses also practically require a high degree of maintenance for data handling, orchestration, and management of results for the calculation of the final required metrics. The ability to automate SA and reduce the time taken for such analysis could benefit operational disaster management by rapidly determining the dominant factors affecting a particular local natural hazard to guide efficient response and planning.

Different methods such as variance-based sensitivity analysis [25][26], Bayesian analysis [244][245][246], Generalized likelihood uncertainty estimation (GLUE) framework and Metropolis algorithm [247][248], neural networks [249][250] and Taylor Series methods [251] have also been used for uncertainty quantification in an environmental context.

Nossent et al. [25] performed Sobol' SA for flow simulations given by a SWAT model to calculate the sensitivity indices of 26 different input parameters. Similarly, sensitivity analysis of SWAT model was carried out in [252][253][41][254]. Yang et al. [18] assessed five different SA techniques applied to a hydrologic model. Brohus et al. [217] used the Morris method to analyze the sensitivity of fire dynamics simulation, while Hilton et al. [227] used polynomial chaos for similar models. Similar works have been done to perform SA of different fire models in [231][255][15]. These mentioned works have applied different sensitivity analysis methods to environmental models without directly considering the computational needs of such analyses.

Researchers have developed several methods and tools including Matlab-based [221][239] and Python-based libraries [210] to calculate the sensitivity indices of input parameters of any environmental models. Wagener et al. [256] developed the Monte Carlo Analysis Toolbox (MCAT) enabled by a Matlab library of different visual and numerical analysis tools for sensitivity analyses of hydrological and environmental models. Another Matlab-based toolbox called *Eikos* [221] was developed by Ekstrom, which is capable of calculating the sensitivity indices of different models developed in Matlab/Simulink environments. D'Augustine has developed MATLODE [257] as a tool for SA of the models described by ordinary differential equations (ODEs) in direct and adjoint approaches. Pianosi et al. [239] constructed a Matlab/Octave-based toolbox called SAFE (Sensitivity Analysis For Everybody) (available now in R and Python as well) to improve the diffusion and quality of global SA in the environmental modeling community. Herman and Usher developed a Python framework called SALib [210], that facilitate the sensitivity analysis of environmental models using different existing SA methods. Roy et al. [258] developed a python-based Bayesian tool for uncertainty quantification. Andrianov et al. [259] developed an open-source software platform called *OpenTURNS (Open source Treatment of Uncertainty, Risk 'N Statistics)* that could treat uncertainty by dedicated to uncertainty treatment by probabilistic methods. Simlab [260] was developed as a free software package by the Joint Research Centre (JRC) of the European Commission. It generates a set of random samples of different parameters and the simulations can be run to compute the measure of sensitivity based on the method used. A package called *sensitivity* in R was developed by Iooss et al. [261] that can calculate the sensitivity indices using various popular methods. These tools and libraries can easily estimate the measure of sensitivity for mathematical models and even for computational models but

only after the sets of input and output values are available after model runs.

To deal with the high computational needs of the global SA of computational models, researchers have adapted a wide range of approaches. Stanfill et al. [262] proposed an easy to set up and inexpensive emulator based sensitivity indices estimators and applied the estimator to perform the sensitivity analysis to APSIM [263]. To deal with the curse of dimensionality in Global Sensitivity analysis, Sheikholeslami et al. [264] proposed a grouping strategy using boot-strapping-based clustering to enable GSA to high-dimensional environmental models. Saltelli et al. [26] highlighted the importance of using surrogate models with a subset of input factors that contribute to most of the variability of model output for model simplification. Efforts have been made to estimate different measures of sensitivity using generic sets of model input and output sets. Pianosi and Wagener [265] improvised their density-based sensitivity measure method (PAWN [29]) with an approximation measure such that the method was applicable to a generic sample of inputs and output for a model. Borgonovo et al. [266] proposed an ensemble of sensitivity measures, based on the different purposes (parameter prioritization, trend identification, and interaction quantification), to provide insights into environmental models without increasing the computational burden. The approach in the work used data-driven estimation of global sensitivity measure along with hybrid local-global method DELSA [267] such that the ensemble of sensitivity measures could be estimated simultaneously. Eldred et al. [268] proposed a multi-level parallel object-oriented framework called *DAKOTA* that provided an extensible interface between simulation runs and iterative sensitivity methods. The framework enabled a problem-solving environment for performance analysis of computational models, but on high-performance computers. All of these efforts addressed the high computational needs of global sensitivity analysis with various approximation methods and approaches to better estimate the effects.

Cloud Computing has come forward as an attractive solution to support high computational demands with its almost unlimited scalable compute resources, storage, and network capacity. Several studies have verified the capability of Cloud Computing to accommodate the computational complexities of different environmental models [18][165][269]. Consequently, global SA of computational models, previously thought to be very difficult (or infeasible) [270, 271], can be conducted on the Cloud. However, to authors' knowledge, there are no systems or services that offer such analyses in a scalable, time-efficient, and convenient manner. As such, this study proposes a cloud-based framework

that can efficiently handle the high computational need of a large number of environmental model simulations. The framework uses scalable Cloud resources to run the computational models with sampled input set to obtain the set of output values for further analyses in a time-efficient manner, which would take several hours to days in a conventional system. The set of input values to the model can be sampled as required and the set of output values obtained after numerous model runs, along with input sets, can be used for various mathematical analyses including sensitivity analyses using different global SA methods. In our work, to validate and demonstrate the capability of the framework, we utilize the sets of input and output values of the model to calculate the sensitivity indices of input parameters to model output using a set of different popular SA methods. These are the Morris method [216], the Sobol' method [272] and the Fast Amplitude Sensitivity Test (FAST) [273]). These methods are chosen as a modular block in the framework based on the standard comparison presented in [274] that highlights the suitability of SA methods for different purposes (ranking, screening, and mapping) with the trade-offs between accuracy and cost taken into consideration. The sampling strategy and index calculation are customized based on the user input and method chosen before a job is launched in the framework. All data management and intermediate calculations are automatically handled to produce the metrics from the SA method. The framework is demonstrated specifically here for sensitivity analysis of wildfire models using the Spark wildfire modeling system, although the method can easily be extended to other natural hazard models. The model input and output set obtained after the model runs in the framework can be further analyzed using any suitable approaches.

4.2.1 Sensitivity Analysis Methods

Sensitivity Analysis (SA) deals with the study of the variation or uncertainty in the model output due to the variation in one or more input parameters. The global SA methods overcome the limitations of local SA such as linearity, normality assumptions, and local variation and are widely used for sensitivity analysis of parameters in different models [275]. We consider three widely adapted global SA methods (one-at-a-time and variance-based) [25, 239], detailed in the following sections.

4.2.1.1 Morris Method

Morris Method [216] is one of the screening-based SA methods. It is often called 'one at a time' (OAT) analysis as each input parameter is varied while keeping the other parameters constant during the model runs. This method classifies the input parameters into three distinct categories - input parameters with negligible effect, parameters with large linear effects without interactions, and parameters with large non-linear and/or interaction effects. The method calculates the sensitivity indices for the parameters j in terms of mean (μ_j^*) and standard deviation (σ_j) of the absolute value of the elementary effects. μ_j^* is the measure of the effect of jth input parameter on the output, where greater values indicate a greater influence of jth input parameter on the variability of the output. σ_j is the measure of the non-linear and interaction effects of the jth input parameter. Smaller values of σ_j signify fewer interaction effects, while higher values of σ_j signify higher interaction effects with at least one other input parameter and/or non-linearities.

For a sample size argument of N (N samples within the range of input and k parameters in a model, calculation of sensitivity indices in Morris method requires $(N+1) \times k$ model runs [216].

4.2.1.2 Sobol' Indices

Sobol' SA [222] is a variance-based SA method that quantifies the input and output variability as probability distributions. The analysis breaks the output variability into the individual input variability and the variability caused by the interaction between the inputs. Consequently, the method quantifies the variability of the input parameters in terms of first-order indices, second-order indices, and total sensitivity indices. The first order index $S1_j$ defines the variability of the model output caused by the variability of input parameter j without considering any interaction with other input parameters. The second-order index $S2_{i,j}$ explains the variability in the model output caused by the non-linear interaction between parameter i and parameter j. The total sensitivity index ST_j defines the total variability caused by the variability in the input parameter j and its non-linear interaction with one or more other input parameters.

For a sample size argument of N and k parameters in a model, calculation of the sensitivity indices requires $2N(k+1)$ model runs if the calculation of second-order indices is enabled [276]. The number of model runs needed is $N(k+2)$ if the calculation of second-order indices is disabled [276]. The second-order index calculation is enabled throughout this study.

4.2.1.3 Fourier Amplitude Sensitivity Test (FAST)

Fourier amplitude sensitivity test (FAST) is a variance-based global sensitivity analysis method. It defines the sensitivity indices based on the conditional variance of the input parameters indicating the individual or joint effects of the parameters on the model output. FAST first uses coefficients of multiple Fourier series expansion of the model output function to represent the conditional variances of the inputs. It then applies the ergodic theorem to transform the multi-dimensional integral to a one-dimensional integral for the evaluation of the Fourier coefficients [273]. The continuous integral function can be recovered from a set of finite sampling points if the Nyquist-Shannon sampling theorem [277] is satisfied. The integral can be evaluated from the summation of the function values at the generated sampling points. FAST gives the indices in terms of first-order indices $S1$ and total effect indices ST. $S1$ quantifies the standalone impact of an input parameter, while ST measures the overall impact of the parameter, including the effects of its non-linear interactions with other parameters.

For a sample size argument of N and k parameters, the calculation of the sensitivity indices in FAST requires $N \times k$ model runs [234].

4.2.2 Cloud-based Framework

Our Cloud-based framework enables sensitivity analyses of natural hazard models using various well-established methods, as explained in the previous section in a time-efficient and convenient manner to address the prohibitively time-consuming issue of such analyses. The components of our Cloud-based SA framework are shown in Figure 4.9. The framework handles the computational complexities of multiple model runs among the distributed Cloud resources and calculates the sensitivity indices for the input parameters to the model. The user uploads a configuration file for running the models and

enters the required inputs into a web interface. These are - 1) the SA method to be used, 2) the required sample size and 3) the number of input parameters. In the framework, three different SA methods are implemented. The sample size input allows the user to specify the total number of samples of the inputs within a predefined range. The user can also specify the number of input parameters for the model through the interface, which, together with the sample size, defines the total model runs required for the analysis. It should be noted that the number of total model runs can be different for different SA methods due to differences between the SA algorithms.

A *Master* retrieves the user input and generates the required samples from the possible input parameter combinations for the SA method selected. The Master then distributes the required model runs to several *Workers* (or Cloud instances) to complete all the required model runs in a time-efficient manner. The Master finally collects the model outputs from all the workers and calculates the sensitivity indices for the input parameters. The calculated indices are stored and can be downloaded from the web interface by the user. In addition to the calculated indices, the user can download the model input and output set of values to perform further relevant analyses. The components description and the features offered by the framework are described further as follows.

4.2.2.1 Web Interface

Users initiate a service request for the calculation of sensitivity indices through a Web Interface. The Web Interface is the only point of interaction between the users and the framework, encapsulating all operations within a graphical user interface. Users can initiate a request by uploading the required configuration and input files into the web interface and launching a job. The interface reflects the status of the service request at different instants of time during the operation. Finally, users can download a text file containing sensitivity indices after the execution of the model runs from the web-interface. Moreover, the user can also download the input and output set of values for the model from the master using the interface.

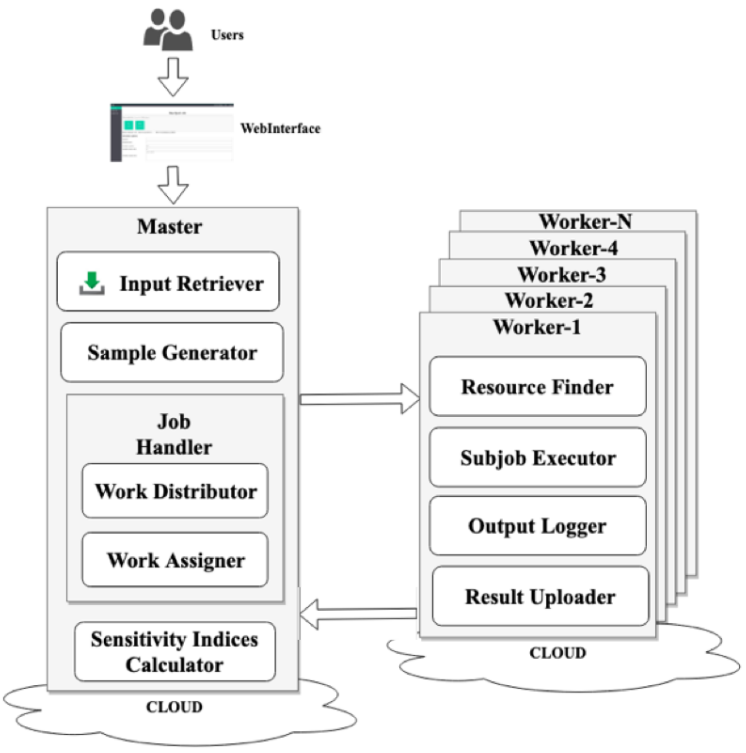


Figure 4.9: Proposed Framework. A master-slave based framework where master assumes all the control functions and slaves executes multiple model runs and sends the output variable to the master for the calculation of sensitivity indices.

4.2.2.2 Master

The Master is the central point of the proposed service framework, controlling how the system serves the service requests in an efficient, scalable, and timely manner. Based on the user input, the Master generates required input parameter combinations. It then divides the required model runs into several sub-jobs, assigns these sub-jobs to multiple Workers, collects the model outputs from the workers upon the completion of the execution, and calculates the sensitivity indices using these outputs. The Master makes use of different mechanisms to distribute the computational complexity of a large number of model runs over multiple Cloud Workers.

The *Input Retriever* retrieves key information from the files uploaded and input fields in the web-interface as per the service request (*job*) initiated by the user. Based on the information retrieved by the Input Retriever, the *Sample Generator* generates sets of input parameter combinations within predefined ranges for the SA method. Each combination results in one model run, producing one model output. It should be noted that for different SA methods chosen, the total number of samples (combinations) generated

is different even for the same sample size. For example, for a sample size of 1000, the total number of input parameter combinations generated for the Morris method is 4000, while the number is only 3000 for the FAST method (for three input parameters in the model).

The *Job Handler* manages the computational complexity of each job by creating multiple independent tasks with a fixed number of model runs, referred to as a *subjob*. Each subjob contributes a fraction to the job. The subjobs are independently executed in multiple workers. The Job Handler consists of two sub-components - the *Subjob Creator* and the *Subjob Assigner*. The Subjob Creator creates several independent subjobs ($S_1, S_2, ..S_N$) with each subjob possessing their respective sample combinations. The Subjob Assigner finds suitable workers for each subjob and assigns the subjob to the worker for the required number of model runs. In the framework, a suitable worker can be a new Cloud instance or an idle worker within the system.

Upon completion of all the required model runs, the *SA Indices Calculator* aggregates the model outputs from the files uploaded by the workers. This component uses SALib python library to calculate the sensitivity indices for the input parameters of the model. The calculated indices are stored and can be downloaded by users through the web-interface.

4.2.2.3 Workers

Workers are the Cloud instances created by the Master to execute the model runs to produce outputs. After the subjobs are assigned, the Workers find and download the required files. The Workers then execute the models multiple times (under a subjob), collect the model outputs, and upload the input combinations along with the respective outputs and time information to the Master. Each worker operates independently within the framework. It is noteworthy that the workers should have the computational model tool pre-installed on them. The workers have sub-components assuming different functions.

The *Resource Finder* finds all the necessary relevant files in the Master, based on the identifier attached to the subjob assigned for the worker and downloads them in the respective directories in the worker. The *Subjob Executor* runs the model in the worker

for as many input parameter combinations in the file downloaded by Resource Finder. The model runs can run as an ensemble to save the time required for multiple data fetch, as one data fetch is enough for all the model runs in such mode. The *Output Logger* employs a text processor to extract the reduced information on the input parameters' combinations, the model output produced by the respective combination, and the time taken for each model run. The reduced information makes the data exchange between Workers and Master more efficient. The *Result Uploader* sends out the requested information extracted by the Output Logger to Master, where the results are stored in a centralized fashion.

4.2.2.4 System Setup

Algorithm 5 outlines the steps used to perform the sensitivity analysis of an environmental model in the framework. The symbols used in the algorithm are listed in Table 4.4. Java is the main programming language used to enable different mechanisms within the framework. Python scripts are used to generate the samples of input parameters' combinations and calculate the sensitivity indices using SALib. Python is used as a programming tool for text processing and synthesis. Nectar Cloud [170], an OpenStack-based Cloud infrastructure, is used to provide the Cloud resources for the model runs to produce the model outputs. For simplicity, we use only one kind of instance flavor (m2.small) for the experiments. The setup can be easily extended to accommodate different types of instance flavors for further optimizing the resource utilization and operation time and cost within the framework. The creation of new Cloud instances is handled by JClouds, which provides Java-based wrapper APIs for OpenStack. The web-interface of the proposed service framework is implemented using VueJS to offer concurrent access to multiple users. The Spark modeling framework is pre-installed on the Cloud image, based on which the new instances are created.

4.2.3 Framework Application Use Case

In this section, we describe the application of our Cloud-based SA framework to wildfire models and analyze the performance of the framework for different SA methods and sample size.

Algorithm 5 Calculation of Sensitivity Indices

Input: $[u, N, k, Method]$
Output: $[Si_1, Si_2, ..S_k]$ (Sensitivity Indices)
Master:
For every service request u_k, Retrieve the values of N, k and $Method$
if Method == 'Sobol' **then**
  Generate $2N(k+1)$ input parameters combinations
else if Method=='Morris' **then**
  Generate $N(k+1)$ input parameters combinations
else if Method=='Fast' **then**
  Generate $N \times k$ input parameters combinations
end if
Calculate $N_S = min\{10, \lceil \frac{\#Samples}{x} \rceil\}$
Divide samples into N_S batches and create N_S subjobs $S_i ... S_{N_S}$
Find N_S workers (W_i) and assign S_i to worker W_i,
For every file uploaded by worker W_i, check if $\#files == N_S$
if $\#files == N_S$ **then**
  Calculate sensitivity indices $Si_1, Si_2, ..S_k$
end if
Worker W_i:
Find Configuration file and sample file F_i in the Master
Download files in respective directories
Execute subjob S_i
For each model run r_c, Extract input combination, model output and time information
if S_i == completed **then**
  Upload reduced result file rf_i to Master
end if
Make worker W_i free and available for other subjobs

Table 4.4: Description of Symbols used

Symbols	Description
u	User Request
N	Sample Size Argument
k	Number of model parameters
$Method$	SA Method
Si_i	Sensitivity Index for parameter k_i
N_S	Number of subjobs for a user request u
$\#samples$	Size of combinations generated
$\#files$	Number of uploaded result files
x	Number of model runs in each subjob
S_i	i^{th} subjob
W_i	i^{th} worker for user request u
F_i	Sample File for subjob S_i
r_c	c^{th} model run in any subjob
r_{f_i}	Reduced result file for subjob S_i

4.2.3.1 Wildfire model

The Spark [3] wildfire modeling system is used to simulate the example of natural hazards for the SA Cloud framework. Spark is a flexible platform for simulating wildfires allowing different types of fire behaviour to be defined using scripts, including rates-of-spread in different fuel types, firebrand dynamics, and risk metrics for fire impact and severity. Simulations in Spark typically require several input data sets for the fire behaviour models, including maps of the land classification, fuel type, topography, fuel information, and meteorological data. Calculations in Spark are parallelized using the

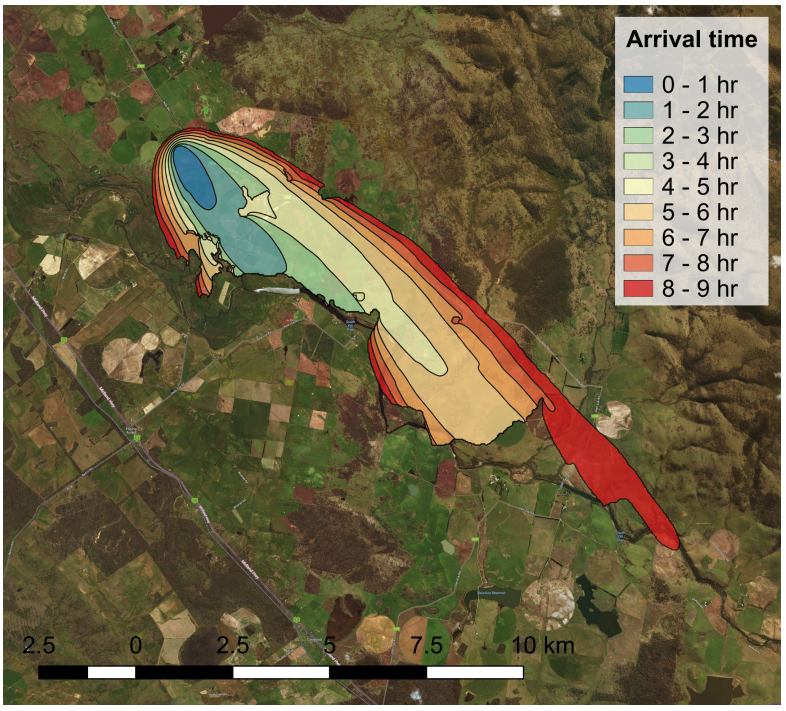


Figure 4.10: Visualization of the spread of fire in Spark for a location in Tasmania, Australia. The colour scale indicates the time of arrival of the fire, with blue being the area covered in the first hour and red the final hour of a nine-hour simulation. The fire is constrained to the south by river.

OpenCL framework to enable the efficient execution of the simulations. Figure 4.10 shows an example simulation for the predicted areas burnt over different periods of time.

For an example of SA analysis, an area in Tasmania, Australia was chosen. Tasmania is one of the most wildfire-prone regions in Australia during the fire season. From 2018 to 2019, 841 wildfires were reported, and 310,311 hectares were burnt by wildfires [278]. As a part of their ongoing effective wildfire management strategy, the Tasmania Fire Service (TFS) and State Emergency Service (SES) have been actively working to create and manage high-quality land data sets relevant to wildfires which were used for this study. The simulations used a number of different empirical fire models for fuels found in Tasmania. Vegetation types from the TasVeg data set [279] were mapped to a number of Australian empirical fire spread models. These were the McArthur [280] and Dry Eucalypt model [188] for forest, a model for buttongrass moorland [281], a model for heathland [195] and grasslands [193].

The parametric sensitivity study was conducted for the meteorological data inputs common to all the empirical models used: the air temperature, relative humidity and wind

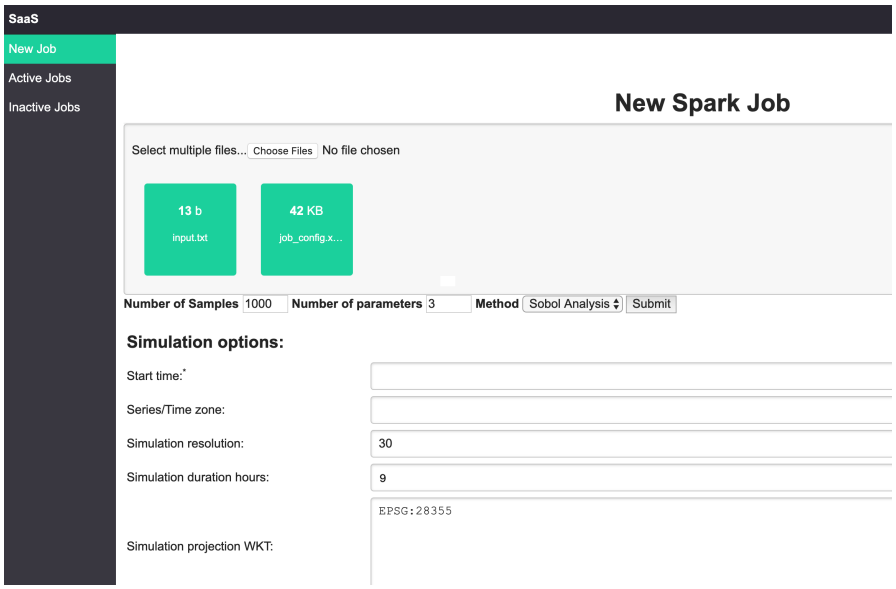


Figure 4.11: User Interface. A user uploads the required configuration file for Spark simulation and enters the sample size argument and desired SA method to run the analysis as a new job in the framework.

speed. All other non-meteorological data inputs were fixed as per the TFS configuration files. The simulations were run for nine hours at a specified single start location within Tasmania. The total fire area (in hectares) burned by the wildfire was considered as the output variable for each simulation in Spark. The ranges of weather data used were based on observations by McArthur [238] and reported in [190]; these are listed in Table 4.5. For simplicity, we assigned a uniform distribution to the parameters while creating samples for the analysis. These distributions, as well as the ignition location of the wildfire, can straightforwardly be changed and the values used here are simply to demonstrate the utility of the framework.

Table 4.5: Probability Density Function (PDF) of Input Parameters

Parameters	pdf	Range
Temperature	Uniform Distribution	[10, 40]
Relative Humidity	Uniform Distribution	[10, 90]
Wind Speed	Uniform Distribution	[10,60]

4.2.3.2 Calculation of Sensitivity Indices at Sample Size Argument = 1000

For a SA calculation of sample size $N = 1000$ the numbers of model runs required were 8000, 4000, and 3000 respectively for Sobol, Morris, and FAST method. Here, the sample size argument of 1000 has been chosen to reflect the high computational demand for sensitivity analyses. Further analysis on the choice of the sample argument

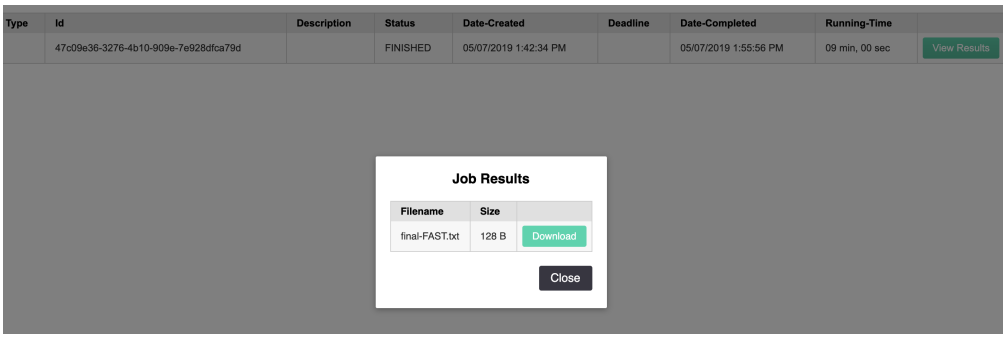


Figure 4.12: A Sample Downloadable File. After the completion of the job execution, the user gets to download a text file with the values of sensitivity indices calculated based on the chosen SA method.

for convergence is included in Section 4.2.4.2. The value can be changed to suit the nature of analysis to be carried out. To perform the SA, a service request was initiated in the framework by uploading a configuration XML file and input file (with information about the sample size argument, number of parameters, and SA method) into the web-interface as shown in Figure 4.11. For this study, the value of x (total number of total model runs in a worker) as defined in Algorithm 5, was taken as 100. The effect of x on the overall time performance of the framework is detailed in a subsequent section. Based on the value of x and the total numbers of samples created, the Master creates a corresponding number of subjobs and assigns them to the Workers. Table 4.6 lists the values of sample size and the total number of subjobs/workers created for different SA methods. Upon completion of the models runs in the workers, Master combines the result files and calculates the sensitivity indices, which can be downloaded from the web-interface, as shown in Figure 4.12. In the framework, we use the cloud instances of flavor type *m2.small* with 1 VCPU, 4 GB RAM, and 10 GB memory Ubuntu 16.04 LTS 'Xenial' amd64. The discussion on the analysis of the sensitivity indices is made in the next section.

Figure 4.13 represents the total time taken by the Cloud framework using a sample argument of 1000. The total time includes the time taken for the creation of new Cloud instances, downloading the files, required model runs, and calculation of the indices. The infrastructure used for the study, Nectar Cloud [170], can experience delays when required to create a large number of instances simultaneously. Such delays appear due to various hardware and physical limitations, including memory size. As such, the time required for the creation of new instances varies from 1 minute to 5 minutes. Due to

Table 4.6: Total model runs (N) and workers for different SA methods

S.N	SA Method	Model Runs	Workers/Subjobs
1	Sobol	8000	80
2	Morris	4000	40
3	FAST	3000	30

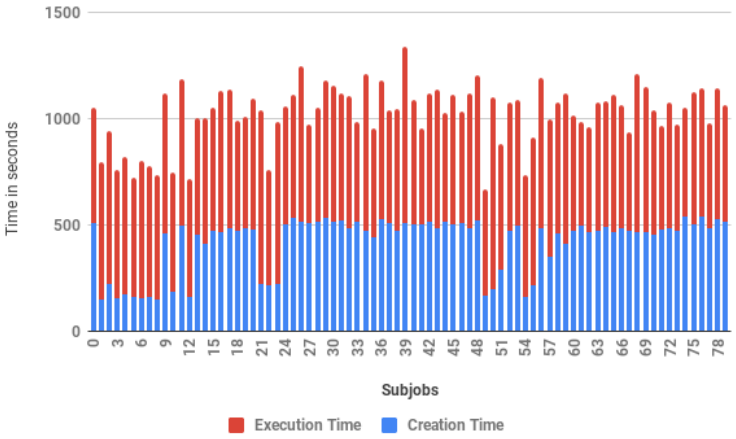
selective downloads, the time needed for downloading the required files and resources is minimal (a few seconds).

Since the indices are calculated only once after the completion of all the subjobs, the time required for indices calculation is also minimal (1 second). A typical user request for calculation of Sobol indices for a sample argument of 1000 takes around 22 minutes, while the same for Morris method takes around 36 minutes. The calculation of the indices using the FAST method takes around 17 minutes. The values of input parameters govern the fire simulations in Spark, and the overall simulation time is strongly dependent on the various combinations of these input parameters. This dependency explains the difference in the time performance of the framework even when Workers have subjobs with the same number of simulations.

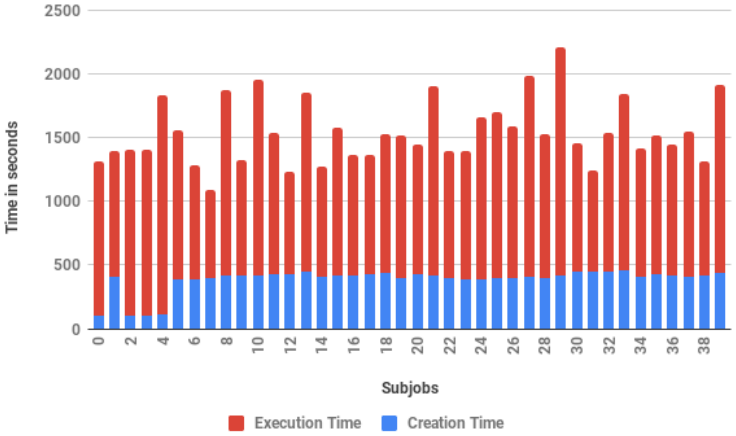
Figure 4.14 compares the total time taken for the SA using the proposed Cloud framework and performing the analysis on a single local machine for a sample argument of 1000 and three different SA methods. For rational comparison, we consider a single local machine with the same hardware specifications as the Cloud instance has (4 GB RAM, 1 VCPU, and 10 GB memory, Ubuntu 16.04 LTS 'Xenial' amd64). For the same set of input parameter combinations, the Cloud framework takes only 3.0% of the time taken by a comparable local system for calculating the indices using Sobol Method. This comparison includes the time taken to create the instances within the framework. Moreover, the Cloud framework further decreases the waiting time for SA using Morris and FAST method as the Cloud framework takes only 4.5% and 6.3% of the time taken by a single machine. In addition to the improvement in waiting time, the Cloud framework offers the benefits of flexibility, scalable resources, ease of use, and efficient handling of model outputs.

4.2.3.3 Performance Analysis

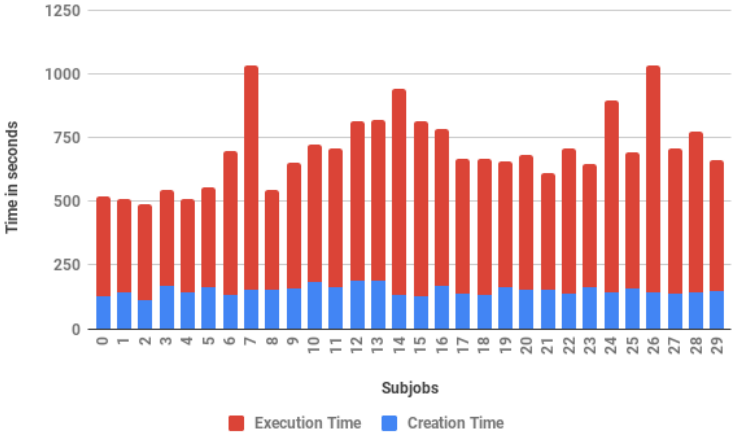
In this section, we analyze the performance of the framework by varying the sample size argument (N) and the number of simulations (x) in a subjob. In our study, creation time



(a) Sobol Method



(b) Morris Method



(c) FAST Method

Figure 4.13: Time required for calculation for SA indices ($x = 100$). The time required for the calculation of the SA indices varies based on the SA method chosen, which is contributed by different sampling methods. The Cloud instances in Nectar Cloud take more to start up when subjected to a large number of simultaneous spun-off requests.

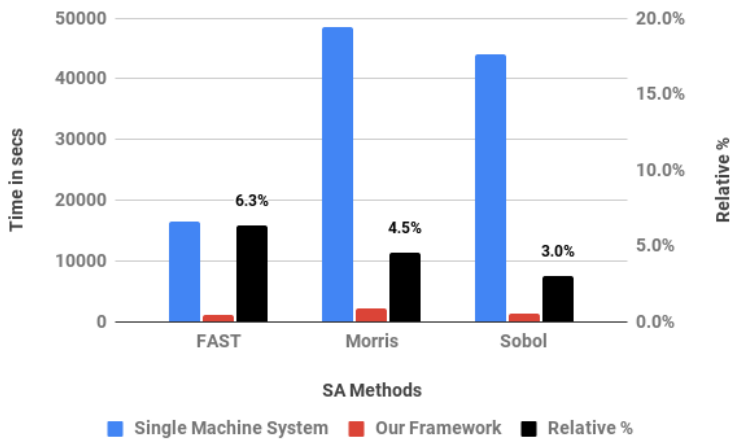
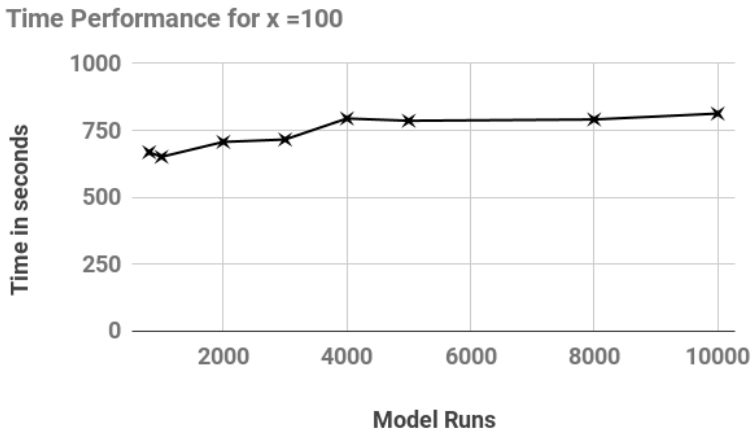


Figure 4.14: Time Performance Comparison of our framework against a single-machine system. Our framework completes the analysis in 3-7% of the total time taken by a local system with a single machine, which is at least 15 times faster. The framework offers additional benefits of flexibility and convenience.

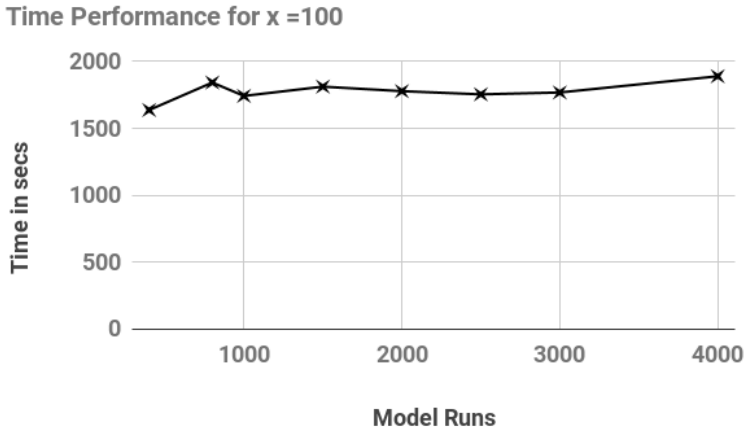
is the time required to create a cloud instance after the request has been initiated while execution time is the time taken by a worker to execute all simulations in a subjob. The execution time includes data fetch time and computative cycle time for all the simulations as explained in [269]. Additionally, we present the impact of parallelizing the model runs in a distributed computing environment of the Cloud. The Cloud instance creation time does not affect the distribution of simulations among the workers and thus, the time taken for the creation of the instances is not considered for the analysis of the impact of parallelization of the model runs. The Cloud instances are assumed to be available and ready to run the models.

The change in the number of sample size argument ultimately changes the total number of model runs for the analysis. The time taken for the calculation of the SA indices for the input parameters for the varied number of samples (simulation runs) is represented in Figure 4.15. Figure 4.16 represents the time taken for the framework to complete the analyses for different values of x.

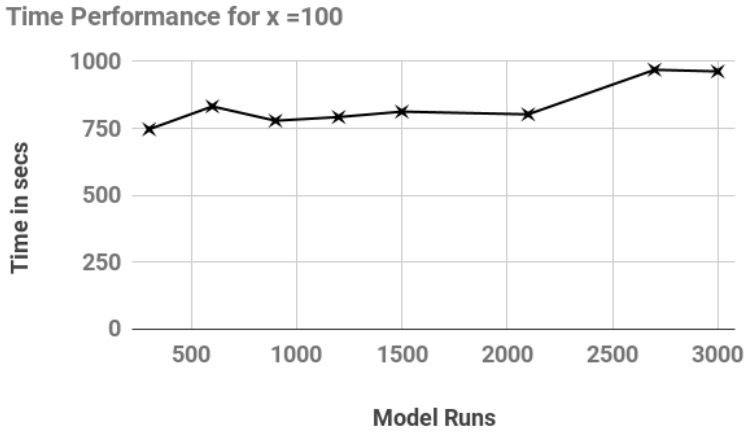
In Figure 4.15, it is evident that the change in the total time taken for the sensitivity analysis is not directly proportional to the change in the number of model runs in a job. The maximum absolute difference in the operation time for a varied number of sample sizes (model runs) for the Sobol method is 162 seconds (Figure 4.15(a)). For Morris, the performance analysis shows that the total operation time has changed by 252 seconds



(a) Sobol Method



(b) Morris Method



(c) FAST Method

Figure 4.15: Variation of total operation time with the sample size (for x=100). There is a variation of total operation time with the change in the value of x but, even when the total model runs (N) increased by a factor of 10, the framework distributes the computational complexity of the analysis over more number of Cloud instances and finishes the entire operation in a time-efficient manner.

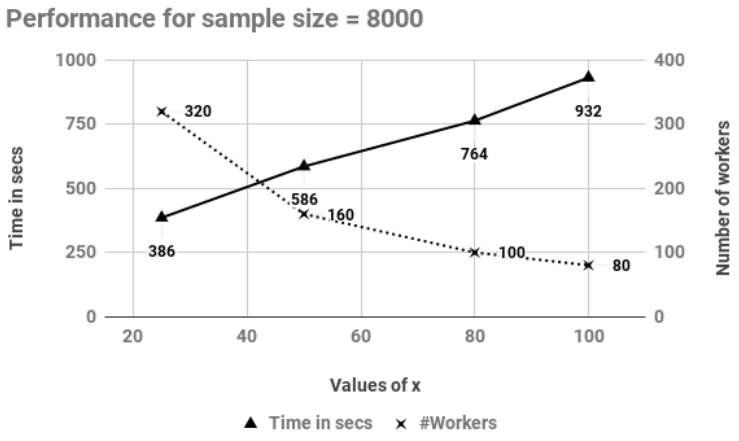
(see Figure 4.15(b)) when the total model runs changed from 400 to 4000. The same statistics for the FAST method stands at 222 seconds (Figure 4.15(c)). Even when the total model runs increased by a factor of 10, the total operation time in the framework did not increase in the same proportion. The Cloud framework distributes the increase in the computational complexity with increasing model runs over multiple Cloud instances. As such, the entire analysis is completed in a time-efficient manner for a large sample size argument. However, there are relative differences between the operation time for each method, which are the result of various combinations of parameter samples resulting in longer simulations in the same worker.

In Figure 4.16, it is clear that the number of model runs in a Worker, x, has a significant impact on the total time taken for a SA request. The total time taken for the completion of a subjob (with multiple model runs) increases with an increase in the number of the model runs in the subjob. The same applies to all the methods in the framework, where the total operation time consistently increases with the increase in the value of x. The number of workers required to serve the requests decreases with an increase in the value of x, as shown in Figure 4.16(b). The increase in the operation time is non-linear and appears to be due to competing data fetching and computing requests on the Cloud instance from the multiple subjobs. Future work will aim to investigate this effect to optimize the size of the subjobs and allocation to the Cloud resources.

4.2.3.4 Impact of Parallelization of Model runs

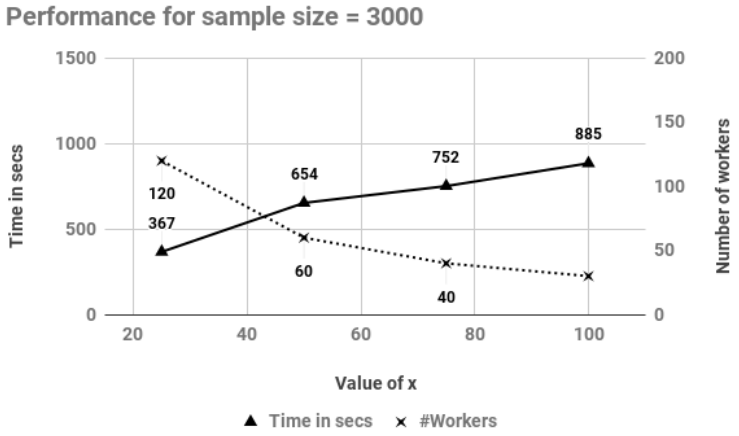
Spark consists of a data fetch and computative cycle [269]. The system can be configured to run N simulations on a single machine, requiring only a single data fetch followed by N sequential simulations. On the Cloud, a job with N simulations can be divided into batches of size n where only one data fetch cycle is required for all simulations in the batch. Each simulation batch can be considered to be a parallelizable task and run in individual workers. The choice of the value of n depends on the availability of the workers, the desired time of job completion, and resource utilization within the system.

As each batch, rather than the components of each simulation, can be parallelized on the Cloud the classic Amdahl's law relation [282] cannot be applied to calculate a relative speed up factor. Instead, we define a speed up factor involving the distribution of jobs to M nodes and the possible execution of n multiple simulations on each node. The



(a) Sobol Method

(b) Morris Method



(c) FAST Method

Figure 4.16: Variation of total operation time with values of x (for $N = 1000$). The total operation time increases with the increase in the number of model runs in a subjob (running in a worker) but, the total workers allocated for the job decreases with the increase in the value of x.

speed up factor, s used here is the ratio of the time taken to complete the job in a single-machine system, T_{single}, to the time taken to complete the job in our framework with multiple Cloud workers, T_{cloud}. The speed up factor s represents the factor by which the time required for the completion of the entire job improves when compared to execution in a single-machine system.

The time taken for a single simulation consists of the fetch time T_{fetch} plus an average time for a simulation, T_{sim} (for this analysis, we generalize the time taken for model runs and use an average unit execution time for T_{sim}). The fetch time T_{fetch} can be considered to be a constant term based on the type of the instance used. For N total simulations on a single-machine system the total time, T_{single}, is therefore $(T_{fetch} + NT_{sim})$. For the Cloud system, all the workers run in parallel (the time for the completion of the job would be the maximum of the time taken by each worker) and thus, the time taken, T_{cloud}, for N total simulations distributed over M nodes each carrying out $n = N/M$ simulations is:

$$T_{cloud} = T_{fetch} + nT_{sim} \tag{4.1}$$

The speed-up factor is therefore:

$$s = \frac{T_{single}}{T_{cloud}} = \frac{T_{fetch} + NT_{sim}}{T_{fetch} + nT_{sim}} \tag{4.2}$$

At the greatest possible cloud utilisation, $M = N$ giving $n = 1$ and an overall theoretical maximum speed up factor of:

$$s = \frac{T_{fetch} + NT_{sim}}{T_{fetch} + T_{sim}} \tag{4.3}$$

In the large simulation limit of $N \to \infty$ Eq. (4.2) gives:

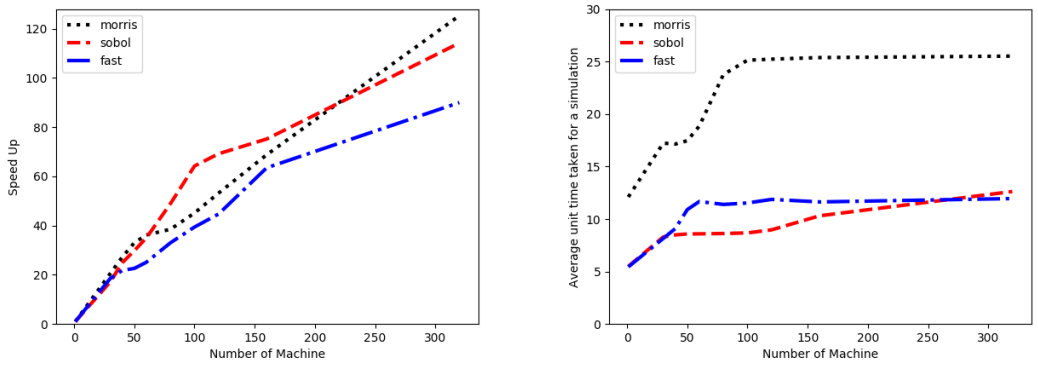
$$\lim_{N\to\infty} \frac{T_{fetch} + NT_{sim}}{T_{fetch} + (N/M)T_{sim}} = M \tag{4.4}$$

Showing that the speed-up should be linear with the number of Cloud nodes, M, for large numbers of simulations.

It should be noted that the simulations can take different times for different input combinations and the fire start location. For example, fires that burn larger areas (due to a combination of high air temperatures and wind speeds with low relative humidity) take longer when compared to those with smaller burned areas (due to low relative values of air temperature and wind speed with high relative humidity). Fires starting closer to the water bodies cease quicker even in favorable weather conditions when compared to the fire starting at a location farther away from water sources. Due to this fact, the speed up factor calculated for a real system is usually less than the theoretical values of the speed up factor and should be considered as a reference point (upper limit) to further optimize the real system.

Figure 4.17 shows the speed up factor and the variation in unit simulation execution time with the increase in the number of workers for a sample size argument of 1000. For the Morris method, the number of total model runs required for the analysis, N, is 4000, taking 48,475 seconds to complete in a single machine system. Assuming 70 seconds on average for the data fetch cycle and 12.10 seconds as the average unit simulation execution time, the maximum possible speed up factor with an arbitrary number of workers (at least 4000) is 590 (calculated using Equation 4.4). As can be seen in Figure 4.17(a), in our framework, the speed up factor linearly increases from 1 to 33 until 50 workers after which the value increases steadily to about 128 for 320 workers (the analysis was limited to this maximum number of workers by our quota of computing nodes on the Cloud system used). The analysis continued for worker sizes beyond 320 would produce a similar increase in the speed up factor. Similar trends are evident with the Sobol and FAST methods, where the gradient in the speed up factor decreases earlier for the FAST method. The linear increase in the speed up factor demonstrates the effectiveness of the framework within the ranges considered.

We also studied the efficiency of using multiple workers in the framework by further analyzing the unit simulation execution time for different methods with an increase in worker size as summarized in Figure 4.17(b). The unit simulation execution time represents the time required for the computative cycle of the simulations in the subjob. The data fetch time for any worker cannot be further reduced or parallelized and hence/, is not considered as a part of the unit simulation execution time. The unit simulation execution time is the least when all the simulations are run in a single machine.



(a) Speed Up factor vs Number of workers

(b) Unit Simulation execution time

Figure 4.17: Analysis of the impact of parallelization of simulations in the framework. Initially, the framework scales linearly with the addition of more workers, but the gradient flattens after a certain point. The linear scaling demonstrates the effectiveness of our framework. The framework can be best utilized at different sizes for different methods.

Consequently, running such a high number of model runs costs the least in a single-machine system, but takes several days to complete. Such delays are not acceptable in an operational environment. With the facilitation of multiple distributed workers in the framework, there has to be a data fetch cycle in each worker, which is then followed by model runs.

Adding more workers in the framework does not necessarily mean an improvement in the unit simulation execution time. Adding more workers can decrease the total time for the completion of the job but, such addition cannot always ensure maximum resource utilization. Due to this fact, the value of unit simulation execution time saturates after a particular value of worker size. For example, the average time spent to run a simulation for Morris method with 100 workers is almost the same for a worker size of 200 for the same job, despite the entire job taking less to complete with worker size of 200. It is also clear from Figure 4.17(b) that workers can be best utilized (maximum resource utilization with a balanced trade-off between time and resources) at a size of 50, 100 and 30 for Morris, Sobol, and FAST methods respectively. Beyond these worker sizes, unit simulation execution time saturates indicating to the fact This can be further studied to define a suitable trade-off between the worker size and time for various situations ensuring better resource utilization.

4.2.4 Sensitivity Analysis Results

In this section, we explain in detail the results of the sensitivity analyses of wildfire models using our framework and discuss the implications of the findings.

4.2.4.1 Sensitivity Indices

The first order (FO) and the total effect of the input parameters on the area burned by the fire are summarized in Table 4.7. The analysis shows that relative humidity has the highest effect on the variability of fire size and the temperature has the least influence. The wind also has a significant effect, but the effect is less than that of relative humidity.

Similar to the first-order indices, the total sensitivity indices also confirm relative humidity as the parameter with the highest impact and temperature with the least impact on the model output variability. The interaction of wind with other parameters is shown by the Sobol analysis to have the greatest effect on the output variability when compared with other interactions. All three methods indicate the interaction of the temperature with other parameters has the least influence in the variance of the fire area. Even though the Morris and FAST methods show that interactions of relative humidity, with other parameters, have the greatest impact, the interactions of wind, with other parameters, also have a significant impact on the model output variability. Relative humidity contributes to 52-67% in the variability of fire area while temperature contributes to just 6-17% of the fire area variability.

Table 4.7: Sensitivity Indices for wildfire simulations (Sample Size Argument $N = 1000$)

Input	**Morris Method**			**Sobol Analysis**			**FAST**		
Parameters	μ	σ	%	**FO**	**Total**	%	**FO**	**Total**	%
Temperature	0.1	0.20	16.6%	0.01	0.09	8.8%	0.01	0.09	6.4%
Rel. Humidity	**0.31**	**0.41**	**51.8%**	**0.65**	**0.91**	**69.3%**	**0.59**	**0.91**	**67.4%**
Wind	0.19	0.31	31.6%	0.07	0.29	21.8%	0.07	0.35	26.2%

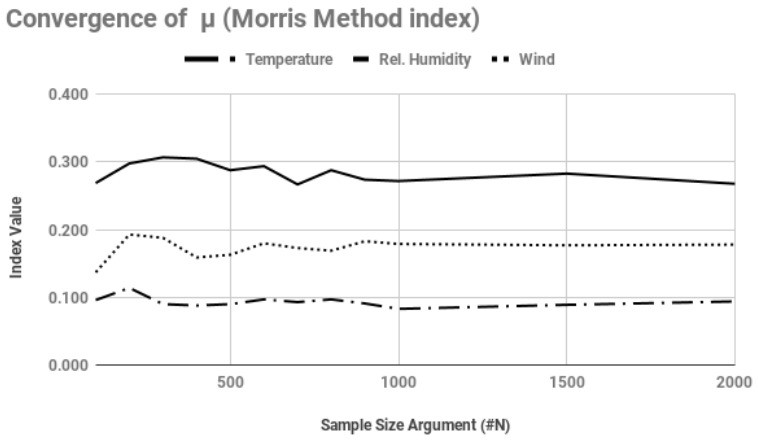
4.2.4.2 Convergence Test

For the convergence of sensitivity indices, we follow the three criteria defined by Sarazzin et al. [209] (consistent sensitivity indices values, parameter ranking, and partitioning between sensitive and least sensitive parameters). The ranks (order of the input parameters with the highest to the lowest impact) of the input parameters for the wildfire

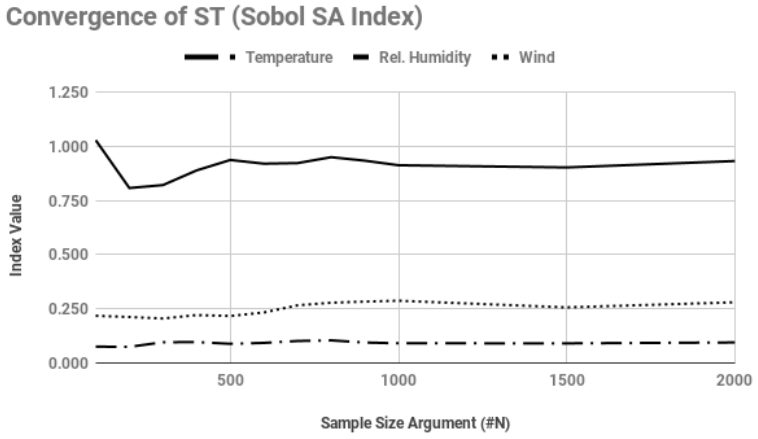
model are quite consistent for every sample size (see Figure 4.18). The difference between the SA indices calculated using Sobol and FAST for the same input parameter is significant (more than 0.05) until the base sample size is 1000. Beyond the value of the base sample size (N) greater or equal to 1000, the indices converge as per the consistent value criterion. The consistent value criterion is fulfilled for Morris method at smaller sample size (at around 500) as Morris method is a semi-quantitative measure and can effectively be used as a proxy for variance-based SA methods with low computation cost and for ranking and screening of the input parameters [26, 283]. Similarly, the distance between the most significant and the least significant impact of the parameters is almost constant for all the methods after $N \geq 500$. Thus, for this study, the minimum base size of the sample for the convergence of SA indices is 1000 for Sobol and FAST and 500 for the Morris method, which requires 8000, 2000 and 3000 model runs respectively.

4.2.4.3 Repeatability Analysis

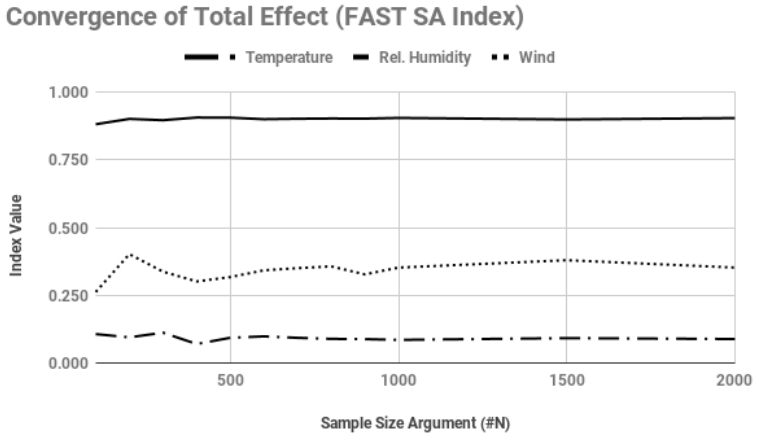
Figure 4.19 represents the scatter plot of the repeatability test for fire simulations where the fire area is calculated once by considering the variability of the temperature and then without considering the variability of the temperature. As represented in the figure, the values of correlation coefficients between the sets of fire areas are 0.92, 0.95, and 0.95 for the input parameter combinations obtained through the Morris, Sobol, and FAST methods respectively. These values (closer to 1) represent the degree of similarity between the two data sets, which again concludes that the temperature has the least impact on the variability of the simulated wildfire area. Such findings could, in practice, help to define a trade-off between the precision of results and the computational time for operational situations. Moreover, new operational tools could be built by cutting down the parameter space of less important input parameters.



(a) Morris Method

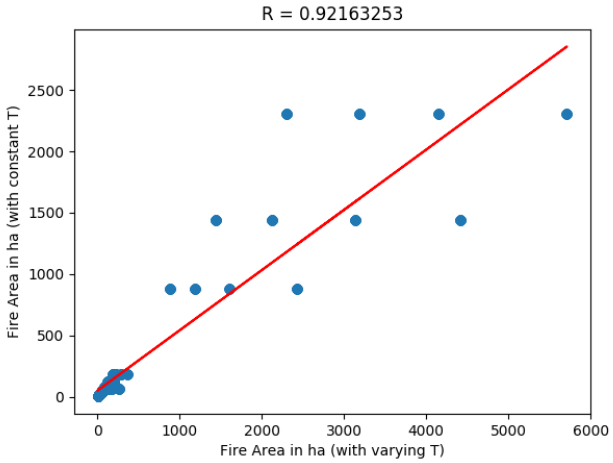


(b) Sobol Method

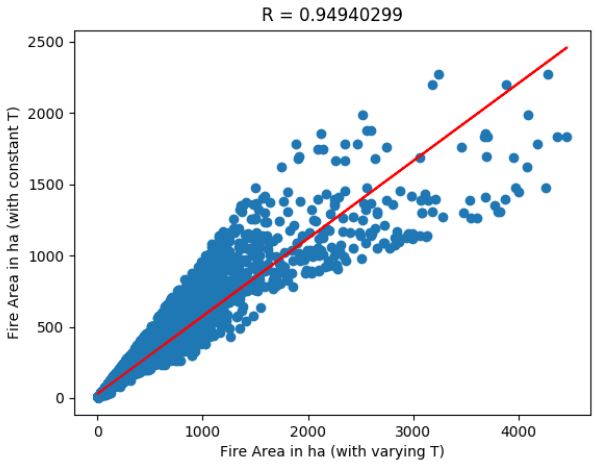


(c) FAST Method

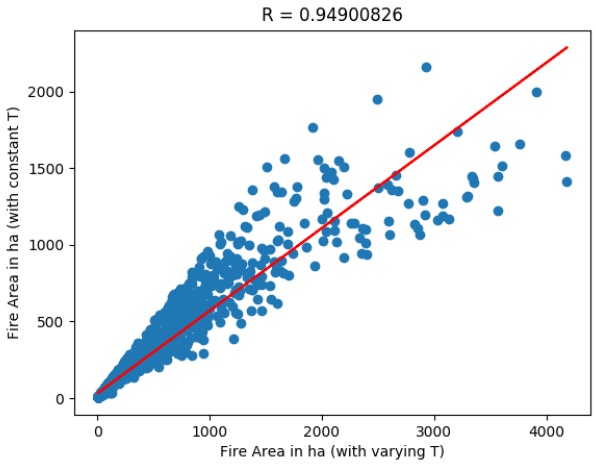
Figure 4.18: Convergence of SA indices for Spark input parameters. The minimum model runs required for the convergence of the indices vary according to the methods. It is fair to say the indices start converging for the value of sample argument ($N \geq 1000$) for all the methods.



(a) Morris Method



(b) Sobol Method



(c) FAST Method

Figure 4.19: Scatter Plot of Repeatability Test for Spark Simulations. The high values (closer to 1) of correlation coefficients calculated for all methods represent the similarities between two different data sets considered for repeatability analysis, thereby confirming the insignificant impact of temperature in fire area.

4.3 Summary

In this chapter, we first demonstrated how risk analysis for determining the conditions with significant threats can be achieved with sensitivity analysis by applying it to wildfire models in the Australian Fire Danger Rating system (AFDRS). Next, we presented a comprehensive comparative analysis of different SA methods facilitate a better choice of methods when it comes to using SA for risk analysis. Finally, we introduced a Cloud framework for rapidly performing a large number of simulations for sensitivity analysis (SA) on such models for rapid risk analysis. Such analysis can help the practitioner identify the input conditions with significant threats and form effective strategies to prepare and respond against them. Furthermore, sensitivity analysis of operational wildfire models also allows the dominant components and degree of connection between the input parameters to be characterized. This characterization can be applied to either improve understanding of a natural hazard in progress by categorizing the current dominant factors driving the event and guide mitigation efforts, or allowing the parameter space for inessential input parameters to be reduced for risk modeling. Such practice can leverage the current state-of-the-art natural hazard modeling systems. The data sets obtained after each analysis can be used for further analyses for better insights into the models. We demonstrated the efficiency of our framework with the scalability achieved while calculating sensitivity indices for simulated fires in Tasmania using the Spark wildfire modeling system. The framework was able to achieve a significant speed improvement (at least about 15 times faster) over a similar analysis on a local machine. The next chapter describes the adaptation of search strategy within conventional ensemble predictions for rapidly identifying the areas of high risks.

Chapter 5

An Adaptive Quadtree-based Approach for Rapidly Determining Areas of Wildfire Risk

In this chapter, to enable rapid risk identification in conventional ensemble predictions, we investigate the possible integration of search mechanisms. As a part of the investigation, we propose a novel quadtree-based approach that adaptively identifies potential high fire-risk areas and produces an increasingly detailed risk map within a given time frame. We present a comprehensive performance analysis of different search patterns within the quadtree-based approach to analyze the trade-off between coverage of risk areas and time efficiency. Our findings show that the performance of the proposed mechanism is statistically better than a random search operation, with up to 80 % of the high fire-risk areas in a large geographic region identified by the method in around 20 % less time than the conventional comprehensive sweep methods. Consequently, our investigation establishes the integration of intelligent search mechanisms in ensemble predictions as an efficient way to rapidly identify high-risk areas. Such a mechanism could help practitioners and operational managers prioritize response activities based on rapidly available risk information.

This chapter is derived from the following work.

KC, U., Garg, S., Hilton, J., & Aryal, J. An adaptive quadtree-based approach for rapidly determining areas of wildfire risk. Nature Sustainability (Submission Draft).

5.1 Introduction

The 2019-2020 Australian wildfire season was one of the worst on record, with around 17 million hectares burnt, 3094 houses destroyed, 33 lives lost and over a billion mammals killed [284]. Around 1600 firefighters and 6386 interstate personnel were reported to be involved in operations around the country. Such devastating wildfire seasons are challenging for any fire authorities to manage, and any predictive information on wildfire risk that can be provided quickly can be crucial operational management.

Recently, wildfire risk models have been widely used to identify high fire-risk locations by predicting the fire spread rate or estimating various risk metrics in an operational framework. To closely quantify the risks associated with fires in a given region, a large number of model runs (*fire simulations*), collectively referred to as an *ensemble*, is run, and statistical analyses on the simulation outputs are carried out. The output of a single simulation could be, for example, the locations burnt by the fire, the maximum intensity of the fire, the height of the flames or the smoke generated by the fire. Due to the complexity and high number of simulations involved such ensembles are computationally expensive [28, 165, 269]. Nonetheless, fire models have been integrated with landscape fire planning, fire suppression, and operational incidental fire management to provide more information to fire responders during emergencies [48–50].

The propagation and behaviour of fire, and the resulting areas affected, are dependent on several factors including topography, fuel and weather. Weather conditions are critical in determining whether a fire will spread from an ignition point and during propagation of the fire [28, 285, 286]. Wildfires frequently occur on 'bad fire days' with hot and dry weather combined with strong winds [287, 288]. The behaviour of a fire can quickly change with weather conditions and the ability to accurately rapidly predict fire behaviour for a set of given weather conditions is required for effective planning and management. One of the most challenging tasks for fire authorities is to identify high risk locations and position the resources effectively ahead of time [289], as during operational fire management practitioners have only limited time for assessment and adaption

to any evolving conditions. Relying on historical records to identify high fire-risk areas may yield inaccurate results too as the contributing factors may have changed over time and consequently, extrapolating the past data may not serve as an alternative to simulating the wildfires [290].

The advancement of computing technologies such as Cloud Computing has significantly decreased the overall time required to derive predictive risk metrics from sets of complex fire simulations [165, 269]. The Cloud provides scalable computational resources allowing multiple simulations to be run simultaneously. However, for large geographical regions the time taken to compute an ensemble predictions at the scale required for an accurate assessment of risk may still be larger than the time window required for planning and response. In addition, a naive sweep method of the entire region, for example simulating fires at a regularly spaced grid of points over the region, requires all points to be run before areas of risk can be identified.

Under the current state-of-the-art of disaster management, several methods have been in use to identify the areas with high fire risks. These methods include the use of satellite images for fire danger assessment, fire-danger/susceptibility rating calculation, and wildfire modeling. Satellite images have been primarily used to characterize the condition and state of fuel (biomass, moisture content, canopy cover and so on) at any given location. As reported, sensors such as AVHRR [32], ATSR [33], MODIS [34], and MSG [35] have been used for various fuel characterization applications [291]. Recently, such applications coupled together with aerial images have been used to draw comprehensive susceptibility maps for regions of interest leading into effective fire danger assessment [36, 37]. Using satellite images for identifying high fire-risk areas is possible only during the events of fires when the satellites are over-passing those areas. Moreover, analyzing satellite or aerial images can be computationally complex which can take a longer time on a limited pool of computing resources [38, 39]. As such, remote sensing techniques with satellites may not be one of the most effective methods to predict or identify high fire-risk areas before or during fire emergencies, especially when the satellite is not over-passing the area of interest. Nevertheless, such techniques are more suited for other applications such as burned area estimation [40], gas emission estimation [41], fire hotspots detection [31] and analysis of fire regimes [42]. Fire danger rating calculation based on meteorological data has also been in practice to identify the areas with high fire-risks. Canadian Fire Weather Index System (CFWIS) [43], US National Fire Danger

Rating System (NFDRS) [44], Russian Nesterov Index [45], the Italian RISICO (RISchio Incendi e Coordinamento) Index [46] and the McArthur model used in Australia [280] all use weather data from weather station or weather forecast model to assess the risk of possible fires for any region of interest for any given day in a year. Similarly, wildfire risk modeling has also been used to identify high fire-risk locations by predicting the fire spread rate or estimating various risk metrics in an operational framework. Such fire models have been integrated with landscape fire planning, fire suppression, and operational incidental fire management to provide more information to fire responders during emergencies [48–50]. Consequently, wildfire models have been one of the key decision-making tools for fire risk management during various stages of fire emergencies.

The central idea to the method presented in this chapter is the observation that only a small fraction of the possible fires over a geographical area will be high risk fires. For example, for a geographical region such as the one used in this study (Tasmania) with around 70,000 possible fire start locations, the number of locations where the resulting risk is extremely high under certain weather conditions may only be around 1% of these. A conventional comprehensive sweep method must run fire simulations at all possible start locations to identify these high fire risk areas, which can delay the overall time to generate risk information. A more efficient search strategy can quickly identify high fire-risk areas in less time but, to the best to our knowledge, such an approach is not used within current operational fire management systems and tools, despite the potential benefits.

Quadtree-based search scheme is one of the widely used search methods in various applications such as image processing, spatial search, and information retrieval. Such a scheme divides a space into four equally-sized sub-spaces and determines if the desired object falls into any of the sub-spaces. Each sub-space with desired objects is further divided into four spaces and searched for desired object until the process reaches the depth of the quadtree specified by the user or no more further space division is possible. Quadtree-based search mechanism exponentially refines the smaller regions and saves time in logarithmic [292]. The time saving achieved with quadtree-based search may be transferred to the search for high fire-risk areas. The search for high fire-risk areas can be centered around an identified high-fire risk area or any randomly selected fire start locations and taken deeper by exponentially dividing a space into four smaller sub-spaces. Such a quadtree-based search mechanism may cover maximum number of high fire-risk

areas in less time by avoiding unfavorable search spaces (low fire-risk areas) during the search operation.

As such, in this chapter we detail a novel approach that employs a quadtree-based search strategy to adaptively identify as many high fire-risk areas as possible for any fire weather within a given time frame, referred to as *planning time* hereafter for clarity. The proposed adaptive approach can help harness the benefits of a quadtree-based search strategy in any system (single or multi-machine) and produce an increasingly detailed fire risk map over time. Additionally, we incorporate the concept of conditional probability to estimate the likelihood of a fire turning highly risky before running any simulation, which saves more time. In the proposed mechanism, we define three different search methods, based on the chess moves (bishop, rook, and queen), to define how to move deeper into quadtree-based search operations. While moving deeper, all low fire-risk areas are dropped to prioritize high fire-risk areas after a level in the quadtree-based search strategy, referred to as *Drop Level*. Moreover, we present a comprehensive performance analysis of the mechanism with methods and demonstrate how the proposed mechanism can alternate between the methods to balance the trade-off between the total number of identified high fire-risk areas, defined as *the coverage of high fire-risk areas* hereafter, and the planning time. Furthermore, we apply the mechanism to the entire Tasmanian region to prove the efficacy of the proposed adaptive mechanism.

5.2 Problem Description

For any given geographical location R with N different possible fire start locations, the problem of high fire-risk areas identification can be stated as a problem of maximizing the total number of identified high fire-risk areas f_i (the risk metric r_{f_i} for f_i should be greater than a threshold Th, that identifies the possible fire damages as highly-risky) within the operating constraints of time t (response time) and computational resources p. Mathematically, it can be expressed as:

$$
\begin{aligned}
& Max \quad C = \sum_i f_i \\
s.t. \quad & \forall j \in \{1, 2, ..., p\}, t_{p_j} \leq t \\
& C \leq N \\
& f_i = \begin{cases} 0 & \text{if } r_{f_i} < Th \\ 1 & \text{if } r_{f_i} \geq Th \end{cases}
\end{aligned} \tag{5.1}
$$

where, t_{p_j} is the time for which the computational resource p_j runs (system time). There is a non-trivial trade-off between the time and the coverage (maximum number of the identified high fire-risk areas). Solving the defined problem would give a sub-optimal solution with a balanced trade-off between the time and the coverage for the operating constraints of time and computational resources.

5.3 Proposed Adaptive Model

We choose a quadtree-based search strategy in our proposed mechanism to solve the described problem due to its ability to quickly identify the desired search results by eliminating the unfavourable options. In the proposed mechanism, the search strategy starts with a bigger space and divides it into four smaller spaces at each level to keep exploring deeper. An example of the quadtree-based search strategy in the Tasmanian region is shown in Figure 5.1, where search operation focuses around an identified high fire-risk start location, represented by a yellow dot. To apply the quadtree-based search strategy, we represent the geographical area with grid points where each point resembles a possible fire start location. In the shallowest level of the quadtree-based strategy, space is represented by four corner points. Consequently, dividing the space into four smaller spaces in the following level is equivalent to finding the neighbours of the corner points where the distance between the points and neighbours keeps changing at each level. At Level 0, the search strategy finds the neighbouring points around the yellow dot at the farthest distance as shown by the largest grid in the figure. With the increase in the value of the level, closer neighbouring points are determined with the distance decreasing. The closer neighbouring points are determined based on three chess moves and the three methods are named accordingly - bishop, rook, and queen (shown in Figure 5.2). The same approach is followed for all the identified high fire-risk start locations. Before

running a fire simulation at any point, the likelihood of fire-risk is estimated using the calculated conditional probability (see Subsection 5.4.1 for details). The simulations are run only if likelihood is more than a certain value (30% in our study). All the identified low fire-risk areas are also dropped after the search operation reaches the *Drop Level* to prioritize the identified high fire-risk areas. As a result, the search strategy finds the high fire-risk areas from a coarse to a finer resolution based on the resources (time and computation) available at various time steps within the given response time. A fire start location is labeled as a high fire-risk location based on a threshold for a risk metric.

The identification of high fire-risk areas in the proposed model is algorithmically represented in Algorithm 6. In the algorithmic representation, *Level* is the drop level in the quadtree-based search, t is the response time, *method* is the method chosen to find the neighbours to any point, f_{h_p} is the list of identified high fire-risk areas, d_p is the list to check and ensure the simulations are not repeated on the same point, $P(H|x_i, fw_k)$ is the likelihood of a fire starting at location x_i turning highly risky, and gW is the width of the grid.

5.4 Experimental Setup

In this section, we describe the general methods used to calculate the conditional probability, test statistical significance and test setup.

5.4.1 Calculation of Conditional Probability

We use Naive Bayes Theorem [293] to estimate the likelihood ($P(H|x_i, fw_k)$) of a fire starting at a location x_i under any fire weather condition fw_k to turn highly risky with data collected during our experiments. In our experiment, fire simulations were run at all possible start locations within Tasmania under different combinations of three weather inputs considered (air temperature, relative humidity and wind speed). The details on the fire weather are given in Subsection 5.4.3.5. The final forest area burned by fires in hectares were recorded as output data from the simulations. A fire was labeled highly risky if the total area burned by the fire was greater than the threshold of 1000 hectares. Accordingly, $P(H|x_i, fw_k)$ can be expressed as follows.

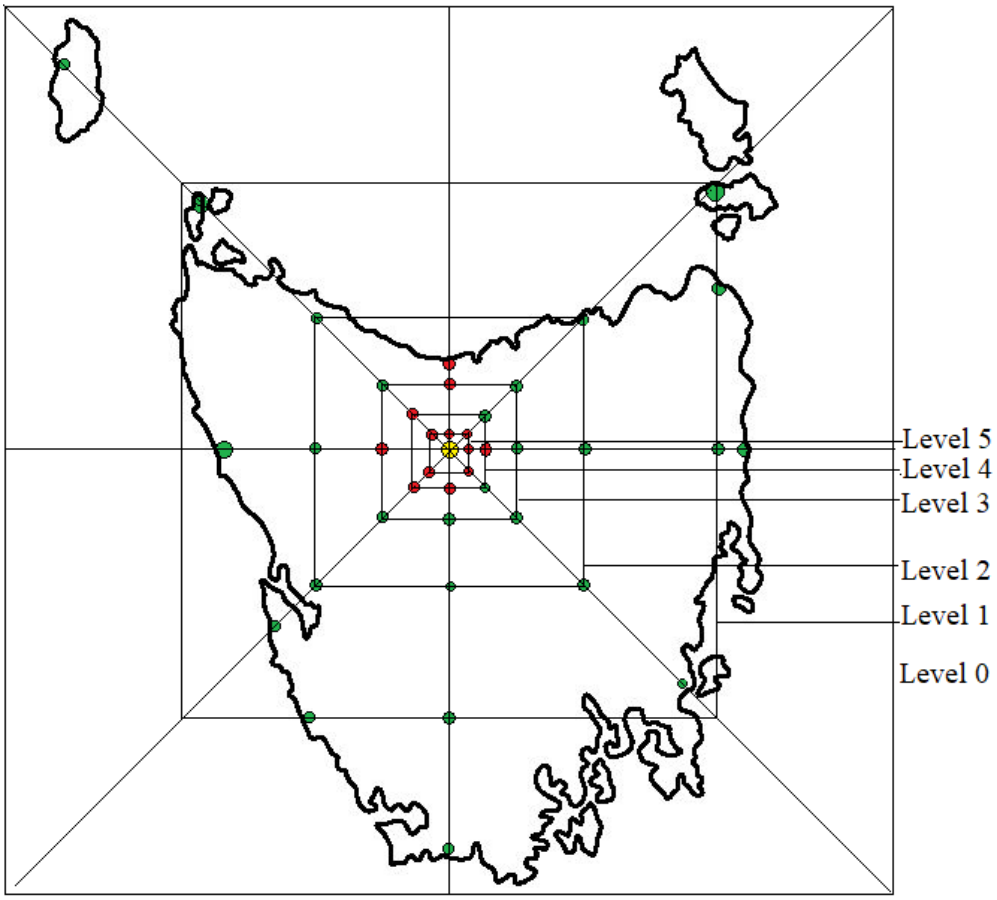
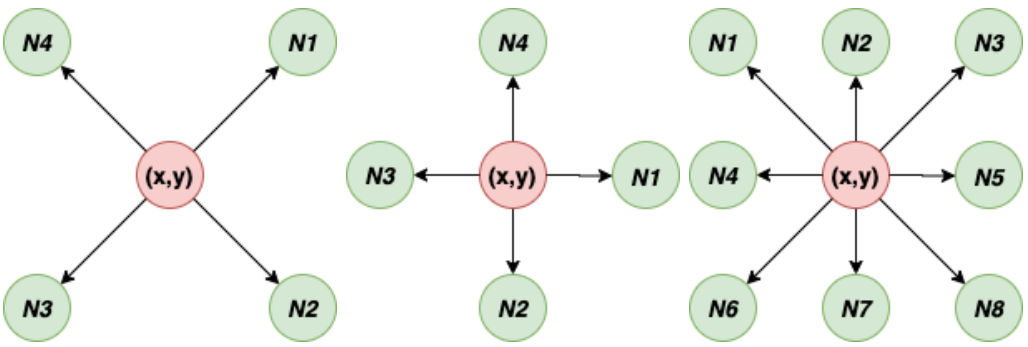


Figure 5.1: An example of quadtree-based search strategy in the Tasmanian region where yellow dot is the identified high fire-risk start location, red dots are the neighboring high fire-risk start locations, and green dots are the neighboring low fire-risk start locations for which further search operation is not carried out.



(a) Bishop (b) Rook (c) Queen

Figure 5.2: Different methods to find neighbors for point (x, y). The neighboring points are based on the chess moves and the methods are named accordingly.

Algorithm 6 Algorithm for the operation of the proposed model

Input: $R, Th, Level, t, method$
Output: f_{hp}

```
Initialize f_hp = [ ], d_p = [ ], l=0
Encode R into a Grid G,
if l == 0 then
    X = {(0,0), (0, gW − 1), (gW − 1, 0), (gW − 1, gW − 1)}
else
    while t >0 do
        wd = int(gW/(2^l))
        if wd == 0 then
            return
        end if
        for f_i in f_hp do
            get (x, y) from f_i
            if method == 'Rook' then
                X = {(x, y + wd), (x, y − wd), (x + wd, y), (x − wd, y)}
            else if method == 'Bishop' then
                X = {(x + wd, y + wd), (x + wd, y − wd), (x − wd, y + wd), (x − wd, y − wd)}
            else
                X = {(x, y+wd), (x, y−wd), (x+wd, y), (x−wd, y), (x+wd, y+wd), (x+wd, y−
                wd), (x − wd, y + wd), (x − wd, y − wd)}
            end if
        end for
        Remove x_i from X if x_i in d_p
        Remove x_i from X if P(H|x_i, fw_k)<30%
        Run simulations on x_i for x_i in X
        Add X to f_hp and d_p
        t = t − t_X, l = l + 1
        if l == level then
            Drop f_i from f_hp for r_{f_i} < Th
        end if
    end while
end if
return f_hp
```

$$P\{H|(x_i, fw_k)\} = \frac{P(x_i|H) \times P(fw_k|H)}{P(x_i, fw_k))} \tag{5.2}$$

5.4.2 Test of Significance for Experimental Results

To verify the statistical correctness of the experimental findings, we draw 30 random samples for the methods under comparison, and conduct a Wilcoxon test for a 95% confidence interval. We propose the null hypothesis as: *The proposed mechanism is as good as the random (or sequential) search operation* and alternative hypothesis as *The proposed mechanism has better performance than the random (or sequential) search operation.* As such, we compare the calculated p-value (p) against the standard value of

0.025 (for the one-tailed test) and accept the null hypothesis if p>0.025, OR reject the null hypothesis and support the alternative hypothesis if p<0.025.

5.4.3 Test Setup

5.4.3.1 Study Area

We choose the Tasmanian region for mechanism testing for several reasons. Firstly, Tasmania is one of the Australian regions with frequent wildfires during summer (841 wildfires in 2018-2019 wildfire season with 310,311 hectares of area burnt by wildfires [278]). Secondly, courtesy of the commitment of the Tasmania Fire Service (TFS) and State Emergency Service (SES) to the nationwide effective wildfire management strategy, Tasmania has high-quality land data sets. Lastly, Tasmania has a well-studied and systematic grid configuration for possible fire start locations within its entirety where fire simulations can easily be run with existing configurations. All the fire simulations are run to simulate the fire behaviours for five hours after the fires start.

5.4.3.2 Wildfire Simulation Tool - Spark

For this study, we consider a wildfire modelling system with the Spark [3] as the fire simulation tool. Spark is a flexible platform for simulating wildfires that allow different types of fire behaviour to be defined using scripts, including rates-of-spread in different fuel types, firebrand dynamics, and risk metrics for fire impact and severity. Fire simulations in Spark typically require several input data sets for the fire behaviour models, including maps of the land classification, topography, fuel information, and meteorological data. Calculations in Spark are parallelized using the OpenCL framework to enable the efficient execution of the simulations.

5.4.3.3 Experimental Platform

To harness the benefits of advanced computing technology with the parallel operation, we utilize a cloud-based framework as explained in our previous work [28] to run the fire simulations. The cloud-based framework is developed over the Cloud infrastructure of Nectar Cloud [170]. The simulations are run on m3.large instances with 8 VCPUs, 16

GB RAM, and 30 GB memory Ubuntu 16.04 LTS 'Xenial' amd64. All the simulation outputs obtained during the study have been stored in CSV format for any possible future uses in a cloud repository [294].

5.4.3.4 Comparable Systems

We compare the performance of the proposed mechanism against a conventional comprehensive sweep system and two different search operations - random and sequential. In a conventional comprehensive sweep system, fire simulations are run for all the possible fire start locations. In such a system, the order in which the locations are picked for running the fire simulation is not important as all the possible locations need to be covered. For a random search operation, fire simulations are run one after another at locations picked randomly within a given response time. In a sequential search operation, the first location to run the fire simulation is picked randomly and the locations thereafter are chosen based on the incremental value of the *seed* as maintained by TFS. As the labelling of the locations with *seed* values has been done by TFS based on a pattern, it is more likely that the high fire-risk areas are concentrated at a particular region and thus, the sequential search can be considered as a strategy with some prior information.

5.4.3.5 Fire Weather

We consider air temperature, relative humidity, and wind speed as the factors that define fire weather, as has been highlighted in the work [28] for Spark simulations. Based on the setup and results of the same work, we consider the discretized ranges ('High', 'Medium', and 'Low') for the values for parameters. The permissible range and the discrete labels assigned based on the values of the factors are summarized in Table 5.1. The range and discretization as done in this study can simply be altered and adapted to suit any analysis as per the requirement.

Table 5.1: Range and discretization of the factors for fire weather

Parameters	**Range**	**Labels with Interval**
Air Temperature	[10, 40]	Low (L) [10,18]
		Medium (M)(18, 33)
		High (H) (33, 40]
Relative Humidity	[10, 90]	Low (L) [70,90]
		Medium (M) (30, 70)
		High (H) [10, 30]
Wind Speed	[10,60]	Low (L) [10,23]
		Medium (M) (23,48)
		High (H) [48, 60]

5.5 Results and Findings

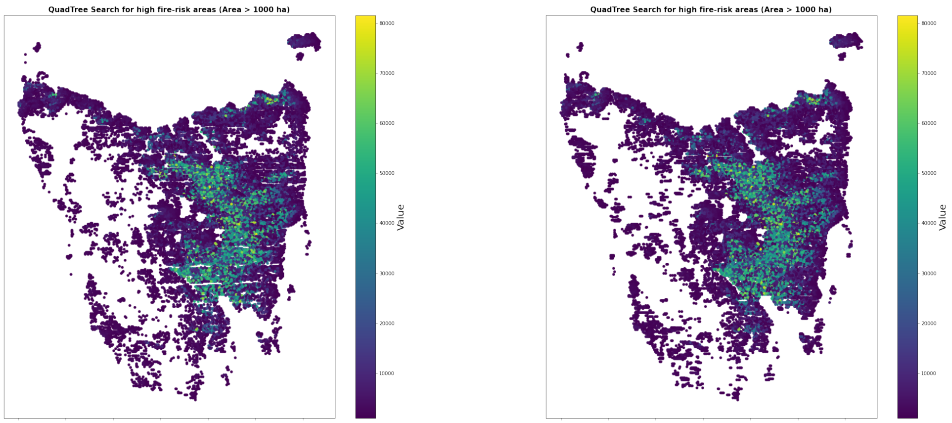
5.5.1 Application to the Tasmanian Region

We used a grid of 256 × 256 to represent 65536 different possible fire start locations within Tasmania. The high fire-risk areas as identified by the proposed mechanism for the fire weather **'HHH'** are shown in Figure 5.3 along with the result of a conventional comprehensive sweep, where a threshold of 1000 hectares to label a start location as a high fire-risk area. The fire weather condition **'HHH'** is the condition when the factors - temperature, relative humidity, and wind speed have the highest influence on fire growth.

The proposed mechanism using the method Bishop was able to identify 23727 out of 36346 high fire-risk areas (about 66% coverage) in about 30% less system time than that of a conventional comprehensive sweep in a single system.

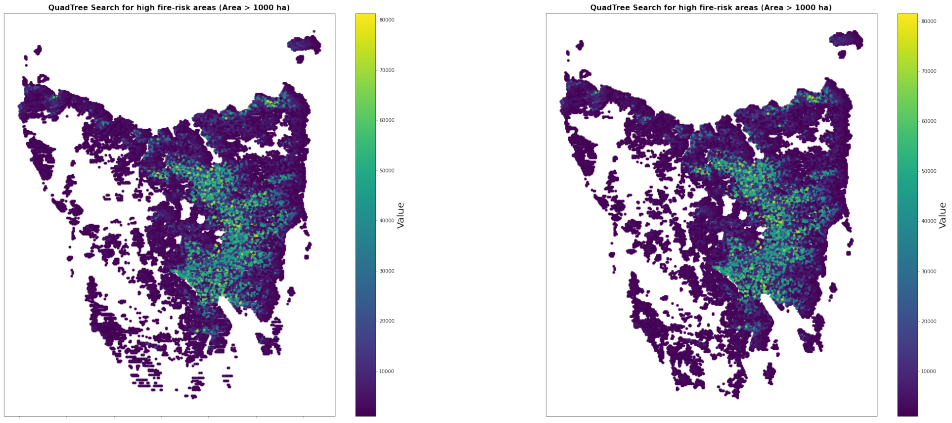
Similarly, the proposed mechanism with Rook and Queen methods was able to find 26621 high fire-risk areas (about 74%) in about 20% less system time and 35166 high fire-risk areas (about 97%) in about 4% less system time when compared to a comprehensive search strategy in a single machine. For a multi-machine cloud system, our proposed mechanism maintained the same coverage in about 35% (Bishop) and 26% (Rook) less system time. For the method - Queen, the proposed mechanism took about 30% more time than the conventional system.

Figure 5.4 shows the adaptive identification of high fire-risk areas with the method 'Bishop' from a coarse to a finer resolution at different time instants until an hour mark (specified planning time). As seen from the figure, the proposed mechanism was able to identify 134 high fire-risk areas in the first ten minutes with a total of 200, 453, 568, 932, and 1072 high fire-risk areas in each 10-minute time step until an hour mark. The



(a) Bishop

(b) Rook



(c) Queen

(d) Conventional comprehensive sweep

Figure 5.3: High fire-risk areas identification with the proposed mechanism for fire weather $FW(T_H, R_H, W_H)$ with different methods

findings for other methods are listed in Table 5.2. The proposed mechanism was able to identify more high fire-risk areas with the Queen method.

Table 5.2: Adaptive high fire-risk area identification of the proposed mechanism within a time limit of an hour with 100 machines in a cloud-based system

Time Step	Methods		
	Bishop	**Rook**	**Queen**
10 minutes	134	101	50
20 minutes	200	171	134
30 minutes	453	375	401
40 minutes	568	471	524
50 minutes	932	621	953
60 minutes	1072	789	1267

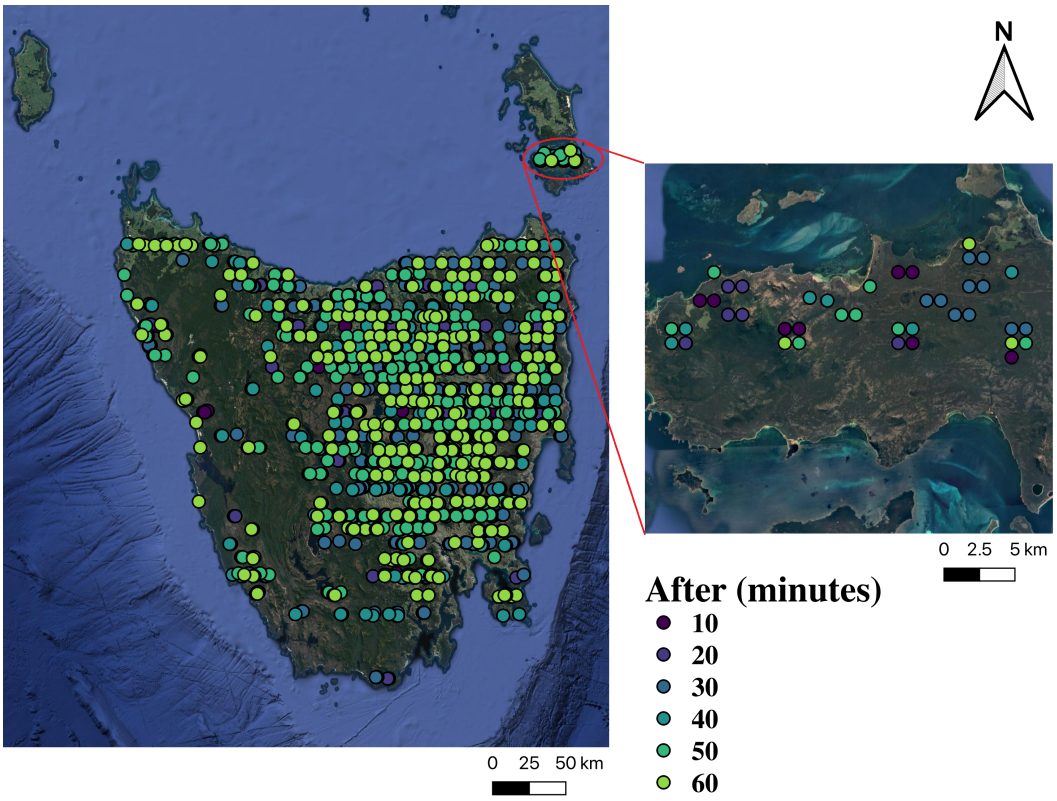


Figure 5.4: High fire-risk areas identified by the proposed mechanism at various time step for a given time window of an hour.

5.5.2 Performance Analysis of the Proposed Mechanism

5.5.2.1 Fire Weather

Figure 5.5(a) shows the performance (high fire-risk area coverage) of the proposed mechanism for 27 different fire weather combinations (Drop level = 6). The fire weather **'LLL'** (as included in the plot) indicates the fire weather with low influences of temperature, relative humidity, and wind speed on the fire growth (as described in Table 5.1). The fire weather **'HHH'** indicates the high influences of the parameters on the fire growth. As

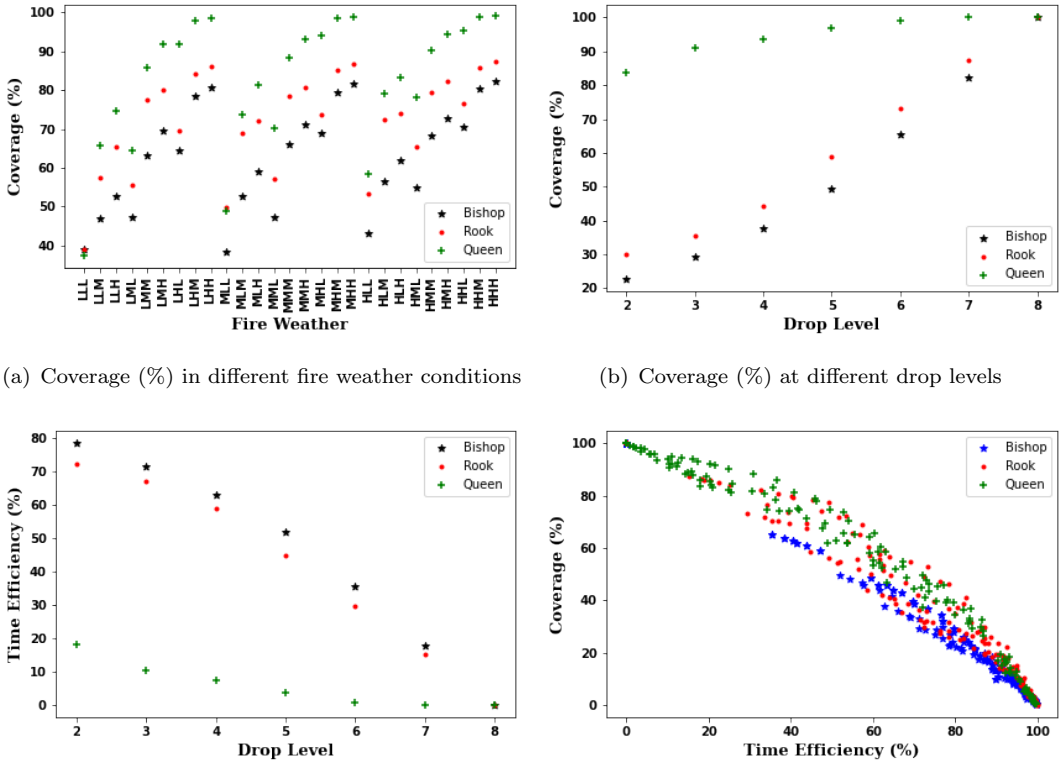
seen from the figure, the proposed mechanism performs better when the fire weather is highly favourable for the fire to grow with a high possibility of more high fire-risk areas. When the number of possible high fire-risk areas was low, the Queen method performed the best, while the methods - Bishop and Rook performed averagely with a minimum of about 40% coverage. For fire weather favourable for fire growth, the proposed mechanism performed well with all the methods. The Queen was found to be the best performing method within the proposed mechanism for most of the fire weather conditions, except for the LLL and MLL fire weather conditions in which Bishop and Rook respectively performed the best. In our analysis, the maximum coverage of the proposed mechanism with three methods stood at about 83%, 88%, and 99% respectively.

5.5.2.2 Coverage

Figure 5.5(b) represents the variation of the high fire-risk area coverage with the change in the value of drop level. As seen from the figure, the Queen method is more efficient than the two other methods, as the minimum coverage of high fire-risk areas with the method was about 84%, while the same for the Bishop and Rook methods stood at about 23% and 30% respectively at drop level of two. For the Queen method, the proposed mechanism started covering more than 90% of the possible high fire-risk areas after the drop level of three. For a drop level of seven, the proposed mechanism achieved the coverage of about 83% and 88% with the methods Bishop and Rook respectively.

5.5.2.3 Time-efficiency

Figure 5.5(c) represents the proportion of the system time that can be saved with the proposed mechanism when compared to a comprehensive sweep of all the fire start locations at different drop levels. The proposed mechanism is the most time-efficient when the method Bishop is used in finding the nearest neighbouring locations for any high fire-risk area while moving deeper in the search operations. The time efficiency with the proposed mechanism decreased with the increase in the value of the drop level. The proposed mechanism took as much as the time taken by a comprehensive swap at drop levels of 8 (Bishop and Rook) and 6 (Queen) when all the possible high fire-risk areas were identified.



(a) Coverage (%) in different fire weather conditions

(b) Coverage (%) at different drop levels

(c) Time efficiency (%) at different drop levels

(d) Trade-off between coverage and time efficiency

Figure 5.5: High fire-risk areas identification with the proposed mechanism for fire weather $FW(T_H, R_H, W_H)$ with different methods

5.5.2.4 Trade-off between Time-efficiency and Coverage

Figure 5.5(d) shows the analysis of the trade-off between the coverage and the time-efficiency within the proposed mechanism, which can be further synthesized to determine the best way to use the proposed mechanism. The proposed mechanism can cover a larger fraction of all the possible high fire-risk areas when there is a long planning time left for coordinating preparedness activities for fire management. For a balanced trade-off between the coverage and the system time efficiency, the proposed mechanism should be used with either the Bishop method or the Rook method as the proposed mechanism can cover over 80% of the high fire-risk areas with about 20% time efficiency. The proposed mechanism has a balanced linear trade-off between the coverage and the time-efficiency with an average of x% coverage of the high fire-risk areas in about y% less time than a conventional sweep, provided $x+y = 100$ when three methods are used in a complementing manner.

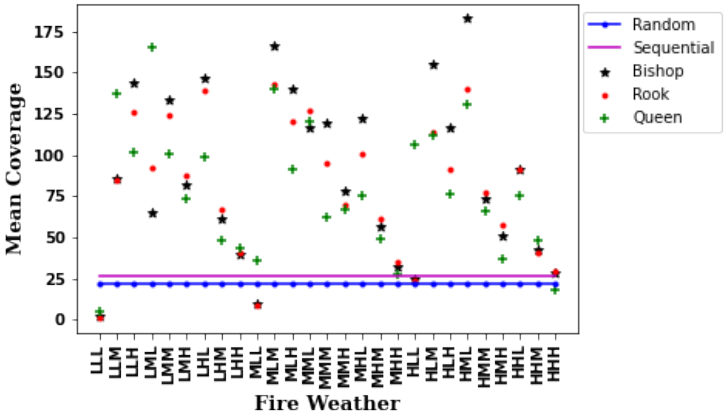


Figure 5.6: Comparison of the mean coverage (total number of identified high fire-risk areas) of the quadtree-based search (proposed mechanism) with a drop level of 6 against a random and a sequential search operation within a planning time of an hour

5.5.2.5 Comparison against a Random and a Sequential Search Operation

Figure 5.6 shows the comparison of the coverage of the proposed mechanism against a random search operation carried out for a total duration of an hour. Against a random search operation, the proposed mechanism started off the search operation without any prior information on possible high fire risk areas and still performed better for almost all the fire weather conditions (except the ones characterized by LLL, MLL, HLL). For the Queen method, the mean of the coverage is less than that of the random and sequential ones for the weather condition HHH. For all the weather conditions where the proposed mechanism performed better, the calculated values of p in Wilcoxon test were extremely smaller than 0.025 thereby statistically verifying the superior performance of the proposed mechanism over a random and a sequential search operation. The coverage of the Queen method was not statistically better than the random and sequential search for the fire weather 'MHH' and so was the case with the Bishop and the Rook method for the fire weather 'HHH'.

5.5.2.6 High fire-risk area identification with multi-machine system

Figure 5.7 depicts the total number of high fire-risk areas identified by the proposed model with the Rook method for a multi-machine system for the fire weather condition identified by $FW(T_H, R_H, W_H)$. As seen from the figure, the proposed mechanism can identify up to 1469 out of 36346 high fire-risk areas in the first hour in a system with

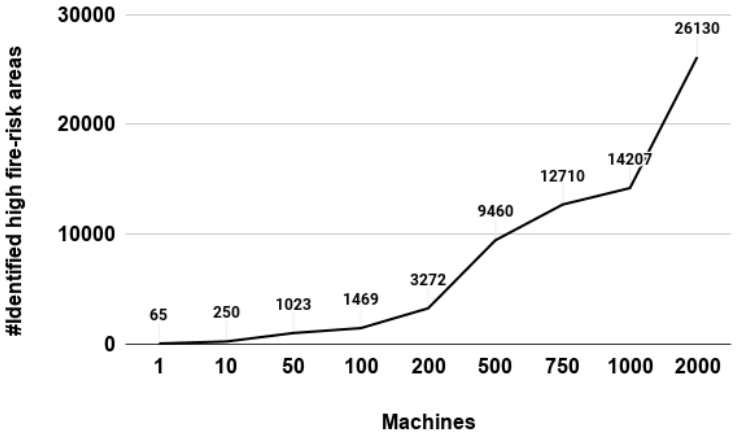


Figure 5.7: Identification of high fire-risk area using the proposed model in a multi-machine system within a specified one hour planning time. The proposed model was able to identify 65 high fire-risk locations with a single-machine system in an hour while, for a system with 2000 machines, the proposed model was able to identify 26130 such locations, thereby demonstrating the flexibility of the proposed model.

100 parallel machines. When operated with a multi-machine system with about 2000 machines, the proposed mechanism can cover more than 72% of the total high fire-risk areas. Such a system, which can be realized with cloud infrastructure, ensures the best use of the proposed mechanism, as a large number of possible high fire-risk areas can be rapidly identified for better wildfire management.

5.6 Discussion

During the mechanism application, the entire grid space represented the Tasmanian region at Level . For any fire weather $FW_i(T_i, R_i, W_i)$, at *Level 0*, the proposed mechanism runs the file simulations at locations represented by the corner points in the grid. The proposed mechanism explores deeper with a unit increment in level by finding the neighbouring points. The mechanism stops its search operation in one of the two conditions, whichever is earlier - the first case when the time constraint (planning time) given to the mechanism ceases and the second case when the maximum depth level is reached (Level 8) here in our mechanism application). Once the search operation is over, the mechanism maps the grid points to the physical geographical locations in the Tasmanian region along with the actual possible fire burnt area for all the identified high fire-risk areas.

The better coverage obtained with the method Queen during the mechanism application was due to more (eight) neighbours around an identified high fire-risk area. Taking more surrounding points around an identified high fire-risk area minimizes the miss of possible high fire-risk start locations. For a multi-machine cloud system, the proposed mechanism (with the Queen method) could cover all the high fire-risk areas but with 30% more time than the conventional system. This finding can be attributed to the necessity of finding eight neighbouring locations around a high fire-risk area that results in several same neighbouring locations for multiple high fire-risk areas. A large number of same neighbouring locations for multiple locations can influence how the simulations are distributed among the multiple machines and incur longer execution times. Such shortcomings with the Queen method can be overcome by intelligently distributing the simulations at each level within the search operation. The proposed mechanism was also able to produce a more detailed map of high fire-risk areas evolving with time for a given planning time of an hour. Such predictive information at coarse level resolution in quick time can help fire authorities to stay alert for better preparedness against an unfolding fire disaster.

In our previous studies [28], the relative humidity and the wind speed have been shown to have a higher influence on the fire growth, while the temperature was found to have a lesser influence. For weather conditions favourable for wildfire growth (high temperature and wind speed, and low relative humidity), the number of possible high fire-risk areas is high and vice versa. While analyzing the mechanism performance in various fire weather conditions, the performance is the worst at the weather conditions when there are a few possible high fire-risk areas. This performance is because the method keeps dropping the low fire-risk areas before the search operation reaches the deepest level. Such a method within the mechanism could miss a few high fire-risk start point located at a deeper level of a quadtree-based search operation. The worst performance of the proposed mechanism at these condition explains why the proposed mechanism was statistically less efficient than a random and a sequential search operation. As a result, the proposed mechanism should not be used for the fire weather conditions where it has been found to be statistically Nevertheless, the coverage, achieved by the proposed mechanism, can be improved by increasing the value of the drop level and prioritizing the Queen method. For the fire weather conditions where fires could grow quickly and burn massive areas, the performance is quite good with as high as about 99% coverage.

We also demonstrated how to use the proposed mechanism in a multi-machine system (local or cloud) for rapidly identifying high fire-risk areas for effective wildfire management. Fire behaviours change drastically with fire weather conditions and thus, predicting the fire behaviours without running fire simulations can be a difficult task. Additionally, some wildfires can grow to burn thousands of hectares in a few hours, and simulating those fires can significantly take longer. While running ensembles in a multi-machine system, several machines may stay idle while a few machines are still running the batches simulating larger fires. Consequently, optimizing computing resource utilization while running ensembles of fire simulations in a multi-machine system is still an open challenge. The proposed mechanism can minimize the number of simulations to be run for analysis but has to be coupled with methods to efficiently distribute the simulations among the machines in a multi-machine system.

5.7 Summary

In this chapter, we proposed a quadtree-based adaptive mechanism that practitioners can use in existing systems (single or multiple machines) to rapidly identify high fire-risk areas within the desired response time during emergencies. To validate the proposed mechanism, we applied the proposed mechanism to the Tasmanian region with a proof-of-concept system to identify such high fire-risk areas at different time steps. The experimental results showed that the proposed mechanism can better handle the trade-off between the coverage of high fire-risk areas and system time in any system. Moreover, compared to conventional comprehensive sweep in ensembles, the proposed mechanism was able to identify more than 80% of high fire-risk areas in about 20% less time. Thus, our investigative effort has demonstrated that incorporating a search strategy into current ensemble disaster predictions can help quickly identify the high-risk areas and give statistically better results than random and sequential methods. Additionally, the proposed mechanism is flexible too, as the value of threshold to define a high fire-risk area, system to realize the mechanism, the number of machines in the system, the drop level for low fire-risk areas, and even the disaster simulation framework (flood simulation tool or others) can easily be changed for comparable mechanism performance. This chapter concludes our initial goal of enabling ensemble predictions for rapid risk estimation, analysis, and identification with cloud-based solutions.

Chapter 6

Conclusions And Future Directions

In this chapter, we first map how each of the objectives defined initially was achieved with the research works conducted. Then, we highlight the key areas where research can be focused in the future as extensions to the solutions presented in the thesis.

6.1 Conclusions

Natural disasters like wildfires are a global problem and require accurate and timely simulations for operational prediction and risk mitigation. The conventional wildfire operational management systems employ ensemble predictions, which on local computers, may take longer than the time window available for the preparation and response against the disaster. Despite the real potentials of the adaptation of operational disaster (wildfire) models in natural hazard (wildfire) modeling and management systems, such an adaptation has challenged the conventional systems of local machines or small pools of computers in terms of computational requirements. As such, in this thesis, we proposed a series of technical and analytical solutions to achieve the goal of rapid risk estimation, analysis, and identification in conventional wildfire management.

Our work first proposed a generic cloud-based framework to support the ensemble predictions for rapid risk estimation. Rapid risk estimation using ensemble prediction requires the ability to efficiently schedule and launch an ensemble of simulations within

a resource or time-constrained envelope. We demonstrated a cloud-based solution for the same where the scheduling of simulations in ensemble prediction was formulated as an optimization problem and solved to determine the most efficient distribution of the simulations. The cost of operations was first minimized by the efficient simulation distribution and then further minimized by intelligent choice of cloud instances based on different pricing models. The validation results were quite promising with operating costs comparable to conventional and cheapest on-premise setup and up to 300% when compared to bag-of-tasks type execution.

Next, we extended the generic framework to support the sensitivity analysis (SA) of disaster models for rapid risk analysis. We demonstrated how the results of sensitivity analysis can be interpreted for risk analysis by applying two SA methods to measure the sensitivity of fire spread rate in empirical fire spread models recommended for operational use in Australian vegetation (AFDRS). The choice of the SA methods in the framework is based on our findings from our analytic comparison of different popular sensitivity analysis methods applied to two empirical fire models (Dry Eucalypt and Rothermel). These two preliminary works set up the foundation for sensitivity analysis of disaster models over our generic framework for our initial goal of rapid risk analysis in operational environment. The efficacy of the framework for the sensitivity analysis of wildfire simulations was tested for simulated fires in Tasmania using the Spark wildfire modeling system. The framework was able to achieve a significant speed improvement (at least about 15 times faster) over a similar analysis on a local machine. The SA in our demonstration investigated the variation in the fire area caused by the input parameters temperature, relative humidity, and wind speed. Relative humidity was found to have the greatest impact on the area burned by the fire, while temperature was the parameter with the least impact.

Finally, on top of the generic cloud-based framework, we added a quadtree-based search strategy within conventional ensemble predictions to enable rapid risk identification without having to run simulations at all possible start locations. The strategy could be used by the practitioners in existing systems (single or multiple and local or cloud machines) to rapidly identify high fire-risk areas within the desired response time during emergencies. The solution was applied to the Tasmanian region with a proof-of-concept system to identify such high fire-risk areas at different time steps. Our findings showed

that the proposed mechanism compared to conventional comprehensive sweep in ensembles was able to identify more than 80% of high fire-risk areas in about 20% less time. Thus, our investigative effort demonstrated that incorporating a search strategy into current ensemble disaster predictions can help quickly identify the high-risk areas and give statistically better results than random and sequential methods.

Additionally, the summary of how each chapter included in this thesis corresponds to the objectives defined earlier is given as follows.

Chapter 2 achieves the objective of reflecting the picture of the current state-of-the-art of Cloud solutions in natural hazard modeling systems with a comprehensive survey of related works that categorized the works based on various aspects of disaster management and supported functionalities.

Chapter 3 meets the objective of building an efficient Cloud-based framework for ensembles of natural disaster simulations in a convenient and resource-efficient manner by proposing a validated cloud framework that minimizes the cost and resources of operation in two distinct phases (of efficiently distributing simulations and intelligent choice of Cloud instances based on price models) for rapid risk estimation.

Chapter 4 extends the framework proposed in Chapter 3 to meet the objective of performing sensitivity analysis of inputs to operational disaster models for rapid risk analysis by proposing a well-validated cloud-based framework that enables such analyses in a convenient and time-efficient manner.

Chapter 5 satisfies the objective of devising novel and innovative mechanisms in ensemble predictions to rapidly identify high risk areas by proposing a quadtree-based search mechanism whose performance for rapid risk identification was superior to the comparable systems with the sequential and random search operation.

The solutions proposed in this thesis are model-agnostic and can be easily transferred to other natural hazard models. We expect these solutions to contribute to the role of ensemble predictions in rapid risk estimation, identification, assessment, and analysis in current disaster management systems.

6.2 Future Works

6.2.1 Comprehensive Disaster Management Framework

Several fire management tools and services including fire spread models have been developed independently to facilitate better-informed decisions at various phases of fire management. Such tools and services have provided crucial information to and made various aspects of fire management more efficient over time. These tools when integrated in a complimenting manner within a framework can significantly improve the effectiveness of the current state-of-the-art in fire management. But when it comes down to forming a coordinated task force, a holistic approach that binds all the available tools and services is non-existent. Non-standard data storage formats, the requirement of high computational and storage resources, non-modular and non-interoperable services, and the absence of standard workflows are some of the challenges for such a holistic framework. Taking the conceptual cloud-based framework as proposed in Chapter 2, future research can focus on customizing cloud solutions further to support a massive holistic framework that supports currently available data and compute-intensive processes in the form of modular blocks within an integrated architecture to offer comprehensive disaster management.

6.2.2 Integration of Big Data and IoT

There is a wide range of data sources that can be utilized in various ways to support a component or multiple components of disaster management. This includes the real-time spatiotemporal data from location services, social media, volunteer geographic information, satellites, and UAVs. The data from sensor web and IoT including airborne and terrestrial Light Detection and Ranging (LiDAR), simulation tools, spatial earth observation data, crowdsourcing and call data records are shown to be important for disaster management. Given how big data and IoT networks have evolved concerning their use and applicability, future research works can be centered around the integration of these technologies in disaster management. The extensive use of IoT sensor networks and Big Data can be made possible for real-time risk mitigation by seamlessly integrating them into existing systems as proposed in Chapter 2 that envisions a conceptual integration

of various technologies. The massive operational simulation data coming out of ensemble predictions and sensitivity analyses as explained in Chapter 3 and Chapter 4 can be explored further in detail to improve the understanding and efficiency of the models with innovative data-driven solutions. The lack of interoperability between different data types, optimization of big data repeatedly used by different components of disaster management, and integration of crowdsourced data with Geospatial data are the areas that are yet to be explored in detail.

6.2.3 Heterogeneous Cloud Infrastructure

The use of heterogeneous Cloud infrastructure for various scientific and commercial applications has been explored in depth. But, the transfer of the same knowledge is yet to be realized in disaster management systems. Future research can be directed in determining optimal mechanisms to accommodate the diverse components of disaster management systems over multiple Cloud infrastructure. The established state-of-the-art of heterogeneous Cloud infrastructure should be customized to fit the inherent features of natural hazard models. The use of heterogeneous Cloud infrastructure in the Cloud-based frameworks proposed in Chapters 3 and 4 for ensemble predictions and sensitivity analysis can be the next step forward that ensures further cost and resource optimization.

6.2.4 Investigation of Search Mechanisms in Ensemble Predictions

This thesis briefly demonstrates the potentials of integrating search strategies in conventional ensemble predictions in Chapter 5. Further research can focus on investigating the effectiveness of various other well-established search strategies in Computer Science for ensemble predictions. Research works can be conducted to clearly outline the scenarios for effective use of such search strategies.

6.2.5 Addition of Sampling-independent Sensitivity Analysis Methods

In this thesis, we explored the application of sampling-based sensitivity analysis methods to fire models and simulations for risk analysis in Chapter 4. Further research can focus on other sampling independent methods such that risk analysis can be done for other

natural hazards as well for which sampling-based methods are difficult or impractical. Additionally, non-meteorological input factors can be considered to expand the scope of such analyses.

Bibliography

[1] Emergency Mangement Australia. Australian emergency management glossary. *Canberra: EMA*, 41, 1998.

[2] CRED Center for Research on Epidemiology of Disasters. Economic losses. https://www.cred.be/sites/default/files/CRED_Economic_Losses_10oct.pdf, 2018.

[3] Claire Miller, James Hilton, Andrew Sullivan, and Mahesh Prakash. SPARK – a bushfire spread prediction tool. In *IFIP Advances in Information and Communication Technology*, pages 262–271. Springer International Publishing, 2015. doi: 10.1007/978-3-319-15994-2_26. URL https://doi.org/10.1007%2F978-3-319-15994-2_26.

[4] Kevin Tolhurst, Brett Shields, Derek Chong, et al. Phoenix: development and application of a bushfire risk management tool. *Australian Journal of Emergency Management, The*, 23(4):47, 2008.

[5] Mark A. Finney. FARSITE: Fire area simulator-model development and evaluation. Technical report, 1998. URL https://doi.org/10.2737%2Frmrs-rp-4.

[6] Scion. Prometheus. https://www.scionresearch.com/rural-fire-research/tools/prometheus, 2018. Accessed: 2018-07-12.

[7] Raymond Cohen, J.E. Hilton, Shahriar Hasan Khan, Yunze Wang, and Mahesh Prakash. Swift: A gpu based coupled hydrodynamic/hydraulic framework for urban flood prediction. 12 2015. doi: 10.13140/RG.2.1.1965.5281.

[8] Julien Lhomme, Paul Sayers, Ben Gouldby, Martin Wills, and Jonatan Mulet-Marti. Recent development and application of a rapid flood spreading method. In *Flood Risk Management: Research and Practice*, pages 15–24. CRC Press,

oct 2008. doi: 10.1201/9780203883020.ch2. URL https://doi.org/10.1201%2F9780203883020.ch2.

[9] Paul Ginoux, Mian Chin, Ina Tegen, Joseph M. Prospero, Brent Holben, Oleg Dubovik, and Shian-Jiann Lin. Sources and distributions of dust aerosols simulated with the GOCART model. *Journal of Geophysical Research: Atmospheres*, 106 (D17):20255–20273, sep 2001. doi: 10.1029/2000jd000053. URL https://doi.org/10.1029%2F2000jd000053.

[10] Guy Schumann, Dalia Kirschbaum, Eric Anderson, and Kashif Rashid. Role of earth observation data in disaster response and recovery: From science to capacity building. In *Springer Remote Sensing/Photogrammetry*, pages 119–146. Springer International Publishing, 2016. doi: 10.1007/978-3-319-33438-7_5. URL https://doi.org/10.1007%2F978-3-319-33438-7_5.

[11] Sundararaman Gopalakrishnan, Qingfu Liu, Timothy Marchok, Dmitry Sheinin, Naomi Surgi, Mingjing Tong, Vijay Tallapragada, Robert Tuleya, Richard Yablonsky, and Xuejin Zhang. Hurricane weather research and forecasting (hwrf) model: 2011 scientific documentation. *L. Bernardet, Ed*, 2011.

[12] Weihong Qian, Xiaolong Shan, Haoyuan Liang, Jing Huang, and Cheuk-Hin Leung. A generalized beta-advection model to improve unusual typhoon track prediction by decomposing total flow into climatic and anomalous flows. *Journal of Geophysical Research: Atmospheres*, 119(3):1097–1117, feb 2014. doi: 10.1002/2013jd020902. URL https://doi.org/10.1002%2F2013jd020902.

[13] Y. K. Lin and Y. Yong. Evolutionary kanai-tajimi type earthquake models. In *Lecture Notes in Engineering*, pages 174–203. Springer Berlin Heidelberg, 1987. doi: 10.1007/978-3-642-83252-9_11. URL https://doi.org/10.1007%2F978-3-642-83252-9_11.

[14] Don L Anderson and James H Whitcomb. The dilatancy-diffusion model of earthquake prediction. In *Proceedings of the Conference on Tectonic Problems of the San Andreas Fault System*, volume 13, pages 417–426. Stanford University Press Palo Alto, Calif., 1973.

[15] Longyan Cai, Hong S. He, Yu Liang, Zhiwei Wu, and Chao Huang. Analysis of the uncertainty of fuel model parameters in wildland fire modelling of a boreal forest

in north-east china. *International Journal of Wildland Fire*, 28(3):205, 2019. doi: 10.1071/wf18083. URL https://doi.org/10.1071%2Fwf18083.

[16] Yaning Liu, Edwin Jimenez, M. Yousuff Hussaini, Giray Ökten, and Scott Goodrick. Parametric uncertainty quantification in the rothermel model with randomised quasi-monte carlo methods. *International Journal of Wildland Fire*, 24(3):307, 2015. doi: 10.1071/wf13097. URL https://doi.org/10.1071%2Fwf13097.

[17] P M Mell and T Grance. The NIST definition of cloud computing. Technical report, 2011. URL https://doi.org/10.6028%2Fnist.sp.800-145.

[18] Chaowei Yang, Yan Xu, and Douglas Nebert. Redefining the possibility of digital earth and geosciences with spatial cloud computing. *International Journal of Digital Earth*, 6(4):297–312, may 2013. doi: 10.1080/17538947.2013.769783. URL https://doi.org/10.1080%2F17538947.2013.769783.

[19] Saurabh Garg, Jagannath Aryal, Hao Wang, Tejal Shah, Gabor Kecskemeti, and Rajiv Ranjan. Cloud computing based bushfire prediction for cyber–physical emergency applications. *Future Generation Computer Systems*, 79:354–363, feb 2018. doi: 10.1016/j.future.2017.02.009. URL https://doi.org/10.1016%2Fj.future.2017.02.009.

[20] Saurabh Garg, Nicholas Forbes-Smith, James Hilton, and Mahesh Prakash. SparkCloud: A cloud-based elastic bushfire simulation service. *Remote Sensing*, 10 (2):74, jan 2018. doi: 10.3390/rs10010074. URL https://doi.org/10.3390%2Frs10010074.

[21] Qunying Huang, Chaowei Yang, Karl Benedict, Songqing Chen, Abdelmounaam Rezgui, and Jibo Xie. Utilize cloud computing to support dust storm forecasting. *International Journal of Digital Earth*, 6(4):338–355, jan 2013. doi: 10.1080/17538947.2012.749949. URL https://doi.org/10.1080%2F17538947.2012.749949.

[22] Zhenlong Li, Chaowei Yang, Qunying Huang, Kai Liu, Min Sun, and Jizhe Xia. Building model as a service to support geosciences. *Computers, Environment and Urban Systems*, 61:141–152, jan 2017. doi: 10.1016/j.compenvurbsys.2014.06.004. URL https://doi.org/10.1016%2Fj.compenvurbsys.2014.06.004.

[23] Babak Behzad, Anand Padmanabhan, Yong Liu, Yan Liu, and Shaowen Wang. Integrating CyberGIS gateway with windows azure. In *Proceedings of the ACM SIGSPATIAL Second International Workshop on High Performance and Distributed Geographic Information Systems - HPDGIS '11*. ACM Press, 2011. doi: 10.1145/2070770.2070774. URL https://doi.org/10.1145%2F2070770.2070774.

[24] J.E. Hilton, C. Miller, A.L. Sullivan, and C. Rucinski. Effects of spatial and temporal variation in environmental conditions on simulation of wildfire spread. *Environmental Modelling & Software*, 67:118 – 127, 2015. ISSN 1364-8152. doi: https://doi.org/10.1016/j.envsoft.2015.01.015. URL http://www.sciencedirect.com/science/article/pii/S1364815215000468.

[25] Jiri Nossent, Pieter Elsen, and Willy Bauwens. Sobol' sensitivity analysis of a complex environmental model. *Environmental Modelling & Software*, 26(12):1515–1525, dec 2011. doi: 10.1016/j.envsoft.2011.08.010. URL https://doi.org/10.1016%2Fj.envsoft.2011.08.010.

[26] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global Sensitivity Analysis. The Primer*. John Wiley & Sons, Ltd, dec 2007. doi: 10.1002/9780470725184. URL https://doi.org/10.1002%2F9780470725184.

[27] Xiaobo Qin, Hong Wang, Yu'e Li, Yong Li, Brian McConkey, Reynald Lemke, Changsheng Li, Kelsey Brandt, Qingzhu Gao, Yunfan Wan, Shuo Liu, Yuntong Liu, and Chao Xu. A long-term sensitivity analysis of the denitrification and decomposition model. *Environmental Modelling & Software*, 43:26–36, may 2013. doi: 10.1016/j.envsoft.2013.01.005. URL https://doi.org/10.1016%2Fj.envsoft.2013.01.005.

[28] Ujjwal KC, Saurabh Garg, James Hilton, and Jagannath Aryal. A cloud-based framework for sensitivity analysis of natural hazard models. *Environmental Modelling & Software*, 134:104800, dec 2020. doi: 10.1016/j.envsoft.2020.104800. URL https://doi.org/10.1016%2Fj.envsoft.2020.104800.

[29] Francesca Pianosi and Thorsten Wagener. A simple and efficient method for global sensitivity analysis based on cumulative distribution functions. *Environmental*

Modelling & Software, 67:1–11, may 2015. doi: 10.1016/j.envsoft.2015.01.004. URL https://doi.org/10.1016%2Fj.envsoft.2015.01.004.

[30] Matieyendou Lamboni, David Makowski, Simon Lehuger, Benoit Gabrielle, and Hervé Monod. Multivariate global sensitivity analysis for dynamic crop models. *Field Crops Research*, 113(3):312–320, sep 2009. doi: 10.1016/j.fcr.2009.06.007. URL https://doi.org/10.1016%2Fj.fcr.2009.06.007.

[31] Stijn Hantson, Marc Padilla, Dante Corti, and Emilio Chuvieco. Strengths and weaknesses of modis hotspots to characterize global fire occurrence. *Remote Sensing of Environment*, 131:152–159, 2013.

[32] Thomas Theis Nielsen, Cheikh Mbow, and Racine Kane. A statistical methodology for burned area estimation using multitemporal avhrr data. *International Journal of Remote Sensing*, 23(6):1181–1196, 2002.

[33] Olivier Arino, Stephen Plummer, and Stefano Casadio. Fire disturbance: the twelve years time series of the atsr world fire atlas. In *Proceedings of the ENVISAT Symposium*, 2007.

[34] Louis Giglio. Modis collection 5 active fire product user's guide version 2.4. *Science Systems and Applications, Inc*, 2010.

[35] M Amraoui, CC DaCamara, and JMC Pereira. Detection and monitoring of african vegetation fires using msg-seviri imagery. *Remote sensing of Environment*, 114(5): 1038–1052, 2010.

[36] S Escuin, R Navarro, and P Fernandez. Fire severity assessment by using nbr (normalized burn ratio) and ndvi (normalized difference vegetation index) derived from landsat tm/etm images. *International Journal of Remote Sensing*, 29(4): 1053–1073, 2008.

[37] Fernando Carvajal-Ramírez, José Rafael Marques da Silva, Francisco Agüera-Vega, Patricio Martínez-Carricondo, João Serrano, and Francisco Jesús Moral. Evaluation of fire severity indices based on pre-and post-fire multispectral imagery sensed from uav. *Remote Sensing*, 11(9):993, 2019.

[38] Rajeswari Balasubramaniam, Srivalsan Namboodiri, Gorthi RK Sai Subrahmanyam, and Rama Rao Nidamanuri. Batch-mode active learning-based superpixel library generation for very high-resolution aerial image classification. In *Computational Intelligence: Theories, Applications and Future Directions-Volume II*, pages 307–318. Springer, 2019.

[39] Chi Yuan, Zhixiang Liu, Anim Hossain, and Youmin Zhang. Fire detection using both infrared and visual images with application to unmanned aerial vehicle forest fire surveillance. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 59292, page V009T12A053. American Society of Mechanical Engineers, 2019.

[40] L Giglio, JT Randerson, GR Van der Werf, PS Kasibhatla, GJ Collatz, DC Morton, and RS DeFries. Assessing variability and long-term trends in burned area by merging multiple satellite fire products. *Biogeosciences*, 7(3):1171–1186, 2010.

[41] A. van Griensven, T. Meixner, S. Grunwald, T. Bishop, M. Diluzio, and R. Srinivasan. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of Hydrology*, 324(1-4):10–23, jun 2006. doi: 10.1016/j.jhydrol.2005.09.008. URL https://doi.org/10.1016%2Fj.jhydrol.2005.09.008.

[42] Sandy P Harrison, Jennifer R Marlon, and Patrick J Bartlein. Fire in the earth system. In *Changing climates, earth systems and society*, pages 21–48. Springer, 2010.

[43] CE Van Wagner, Petawawa Forest, et al. Development and structure of the canadian forest fireweather index system. In *Can. For. Serv., Forestry Tech. Rep.* Citeseer, 1987.

[44] Jack D Cohen. *The national fire-danger rating system: basic equations*, volume 82. US Department of Agriculture, Forest Service, Pacific Southwest Forest and ..., 1985.

[45] VG Nesterov. Flammability of the forest and methods for its determination,(gorimost lesa i metodi eio opredelenia), goslesbumizdat, ussr state ind. *Press, Moscow*, 1949.

[46] Paolo Fiorucci, Francesco Gaetani, Riccardo Minciardi, and Eva Trasforini. Forest fire dynamic hazard assessment and pre-operational resource allocation. *IFAC Proceedings Volumes*, 38(1):91–96, 2005.

[47] IR Noble, AM Gill, and GAV Bary. Mcarthur's fire-danger meters expressed as equations. *Australian Journal of Ecology*, 5(2):201–203, 1980. URL https://doi.org/10.1111/j.1442-9993.1980.tb01243.x.

[48] Joe H Scott, Matthew P Thompson, and David E Calkin. A wildfire risk assessment framework for land and resource management. 2013.

[49] Matthew P Thompson, Joe Scott, Don Helmbrecht, and Dave E Calkin. Integrated wildfire risk assessment: framework development and application on the lewis and clark national forest in montana, usa. *Integrated Environmental Assessment and Management*, 9(2):329–342, 2013.

[50] I Mitsopoulos, G Mallinis, S Zibtsev, M Yavuz, B Saglam, O Kucuk, V Bogomolov, A Borsuk, and G Zaimes. An integrated approach for mapping fire suppression difficulty in three different ecosystems of eastern europe. *Journal of Spatial Science*, 62(1):139–155, 2017.

[51] Dong Cui, Yunlong Wu, and Qiang Zhang. Massive spatial data processing model based on cloud computing model. In *2010 Third International Joint Conference on Computational Science and Optimization*. IEEE, 2010. doi: 10.1109/cso.2010.168. URL https://doi.org/10.1109%2Fcso.2010.168.

[52] Chaowei Yang, Michael Goodchild, Qunying Huang, Doug Nebert, Robert Raskin, Yan Xu, Myra Bambacus, and Daniel Fay. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? *International Journal of Digital Earth*, 4(4):305–329, jul 2011. doi: 10.1080/17538947.2011.587547. URL https://doi.org/10.1080%2F17538947.2011.587547.

[53] Peter Bodik, Armando Fox, Michael J. Franklin, Michael I. Jordan, and David A. Patterson. Characterizing, modeling, and generating workload spikes for stateful services. In *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*. ACM Press, 2010. doi: 10.1145/1807128.1807166. URL https://doi.org/10.1145%2F1807128.1807166.

[54] Qunying Huang, Jing Li, and Zhenlong Li. A geospatial hybrid cloud platform based on multi-sourced computing and model resources for geosciences. *International Journal of Digital Earth*, 11(12):1184–1204, oct 2017. doi: 10.1080/17538947.2017.1385652. URL https://doi.org/10.1080%2F17538947.2017.1385652.

[55] Vagelis Hristidis, Shu-Ching Chen, Tao Li, Steven Luis, and Yi Deng. Survey of data management and analysis in disaster situations. *Journal of Systems and Software*, 83(10):1701–1714, oct 2010. doi: 10.1016/j.jss.2010.04.065. URL https://doi.org/10.1016%2Fj.jss.2010.04.065.

[56] Wilfried Haeberli, Andreas Kääb, Daniel Vonder Mühll, and Philip Teysseire. Prevention of outburst floods from periglacial lakes at grubengletscher, valais, swiss alps. *Journal of Glaciology*, 47(156):111–122, 2001. doi: 10.3189/172756501781832575. URL https://doi.org/10.3189%2F172756501781832575.

[57] Penny Allan and Martin Bryant. The critical role of open space in earthquake recovery: a case study. In *EN: Proceedings of the 2010 NZSEE Conference (2010, Nueva Zelandia)*, pages 1–10, 2010.

[58] F Fiedrich, F Gehbauer, and U Rickers. Optimized resource allocation for emergency response after earthquake disasters. *Safety Science*, 35(1-3):41–57, jun 2000. doi: 10.1016/s0925-7535(00)00021-7. URL https://doi.org/10.1016%2Fs0925-7535%2800%2900021-7.

[59] Stéphane Hallegatte and Valentin Przyluski. *The economics of natural disasters: Concepts and methods*. The World Bank, dec 2010. doi: 10.1596/1813-9450-5507. URL https://doi.org/10.1596%2F1813-9450-5507.

[60] Sandy A Mehlhorn. *Method for prioritizing highway routes for reconstruction after a natural disaster*, volume 71. 2009.

[61] Claussen M., Mysak L., Weaver A., Crucifix M., Fichefet T., Loutre M.-F., Weber S., Alcamo J., Alexeev V., Berger A., Calov R., Ganopolski A., Goosse H., Lohmann G., Lunkeit F., Mokhov I., Petoukhov V., Stone P., and Wang Z. Earth system models of intermediate complexity: closing the gap in the spectrum of climate system models. *Climate Dynamics*, 18(7):579–586, mar 2002. doi: 10.1007/s00382-001-0200-1. URL https://doi.org/10.1007%2Fs00382-001-0200-1.

[62] Stephen H. Schneider and Robert E. Dickinson. Climate modeling. *Reviews of Geophysics*, 12(3):447, 1974. doi: 10.1029/rg012i003p00447. URL https://doi.org/10.1029%2Frg012i003p00447.

[63] Sachi Pandey, Vibhore Tyagi, and Kriti Pathak. Role of open source software and ict in disaster management. *Available from International Journal of Scientific and Engineering Research (http://xplore. ijser. org/)*, 2013.

[64] Alexander Mills, Rui Chen, JinKyu Lee, and H. Raghav Rao. Web 2.0 emergency applications: How useful can twitter be for emergency response? *Journal of Information Privacy and Security*, 5(3):3–26, jul 2009. doi: 10.1080/15536548.2009.10855867. URL https://doi.org/10.1080%2F15536548.2009.10855867.

[65] M. Malathi. Cloud computing concepts. In *2011 3rd International Conference on Electronics Computer Technology*. IEEE, apr 2011. doi: 10.1109/icectech.2011.5942089. URL https://doi.org/10.1109%2Ficectech.2011.5942089.

[66] R. L. Grossman. The case for cloud computing. *IT Professional*, 11(2):23–27, March 2009. ISSN 1520-9202. doi: 10.1109/MITP.2009.40.

[67] Di Liu, Ying Wang, and Lian Guang Liu. Discussion on power grid magnetic storm disaster monitoring system based on cloud computing. *Advanced Materials Research*, 341-342:641–645, sep 2011. doi: 10.4028/www.scientific.net/amr.341-342.641. URL https://doi.org/10.4028%2Fwww.scientific.net%2Famr.341-342.641.

[68] John L. Schnase, William P. Webster, Lynn A. Parnell, and Daniel Q. Duffy. The NASA center for climate simulation data management system. In *2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, may 2011. doi: 10.1109/msst.2011.5937235. URL https://doi.org/10.1109%2Fmsst.2011.5937235.

[69] Q. Li, T. Zhang, and Y. Yu. Using cloud computing to process intensive floating car data for urban traffic surveillance. *International Journal of Geographical Information Science*, 25(8):1303–1322, aug 2011. doi: 10.1080/13658816.2011.577746. URL https://doi.org/10.1080%2F13658816.2011.577746.

[70] Kevin Montgomery and Carsten Mundt. A new paradigm for integrated environmental monitoring. In *Proceedings of the 1st International Conference and*

Exhibition on Computing for Geospatial Research & Application - COM.Geo '10. ACM Press, 2010. doi: 10.1145/1823854.1823888. URL https://doi.org/10.1145%2F1823854.1823888.

[71] Kostas Kalabokidis, Nikolaos Athanasis, Fabrizio Gagliardi, Fotis Karayiannis, Palaiologos Palaiologou, Savas Parastatidis, and Christos Vasilakos. Virtual fire: A web-based GIS platform for forest fire control. *Ecological Informatics*, 16:62–69, jul 2013. doi: 10.1016/j.ecoinf.2013.04.007. URL https://doi.org/10.1016%2Fj.ecoinf.2013.04.007.

[72] P. Soille, A. Burger, D. De Marchi, P. Kempeneers, D. Rodriguez, V. Syrris, and V. Vasilev. A versatile data-intensive computing platform for information retrieval from big geospatial data. *Future Generation Computer Systems*, 81:30–40, apr 2018. doi: 10.1016/j.future.2017.11.007. URL https://doi.org/10.1016%2Fj.future.2017.11.007.

[73] Michael F. Goodchild, May Yuan, and Thomas J. Cova. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science*, 21(3):239–260, mar 2007. doi: 10.1080/13658810600965271. URL https://doi.org/10.1080%2F13658810600965271.

[74] Yoshitaka Shibata, Noriki Uchida, and Yuji Ohashi. Problem analysis and solutions of information network systems on east japan great earthquake. In *2012 26th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, mar 2012. doi: 10.1109/waina.2012.225. URL https://doi.org/10.1109%2Fwaina.2012.225.

[75] Yezhou Wang, Chen Chen, Jianhui Wang, and Ross Baldick. Research on resilience of power systems under natural disasters—a review. *IEEE Transactions on Power Systems*, 31(2):1604–1613, mar 2016. doi: 10.1109/tpwrs.2015.2429656. URL https://doi.org/10.1109%2Ftpwrs.2015.2429656.

[76] Tania Lorido-Botran, Jose Miguel-Alonso, and Jose A. Lozano. A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of Grid Computing*, 12(4):559–592, oct 2014. doi: 10.1007/s10723-014-9314-7. URL https://doi.org/10.1007%2Fs10723-014-9314-7.

[77] Vijay Sarathy, Purnendu Narayan, and Rao Mikkilineni. Next generation cloud computing architecture: Enabling real-time dynamism for shared distributed physical infrastructure. In *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*. IEEE, 2010. doi: 10.1109/wetice.2010.14. URL https://doi.org/10.1109%2Fwetice.2010.14.

[78] Jinesh Varia. Cloud architectures. *White Paper of Amazon, jineshvaria. s3. amazonaws. com / public / cloudarchitectures-varied. pdf*, 16, 2008.

[79] Sambit Sahu, Anca Sailer, and Hidayatullah H Shaikh. Modular cloud computing system, May 17 2016. US Patent 9,342,368.

[80] Brian JS Chee and Curtis Franklin Jr. *Cloud computing: technologies and strategies of the ubiquitous data center*. CRC Press, 2010.

[81] Ronald L Krutz and Russell Dean Vines. *Cloud security: A comprehensive guide to secure cloud computing*. Wiley Publishing, 2010.

[82] Sergio Di Martino, Clemente Giorio, and Raffaele Galiero. A rich cloud application to improve sustainable mobility. In *Web and Wireless Geographical Information Systems*, pages 109–123. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-19173-2_10. URL https://doi.org/10.1007%2F978-3-642-19173-2_10.

[83] Jinhang Yu, Joongbin Lim, and Kyoo seock Lee. Investigation of drought-vulnerable regions in north korea using remote sensing and cloud computing climate data. *Environmental Monitoring and Assessment*, 190(3), feb 2018. doi: 10.1007/s10661-018-6466-0. URL https://doi.org/10.1007%2Fs10661-018-6466-0.

[84] AT Jeyaseelan. Droughts & floods assessment and monitoring using remote sensing and gis. In *Satellite remote sensing and GIS applications in agricultural meteorology*, volume 291. World Meteorol. Org. Dehra Dun, India. Geneva, Switz, 2003.

[85] Zhanming Wan, Yang Hong, Sadiq Khan, Jonathan Gourley, Zachary Flamig, Dalia Kirschbaum, and Guoqiang Tang. A cloud-based global flood disaster community cyber-infrastructure: Development and demonstration. *Environmental Modelling & Software*, 58:86–94, aug 2014. doi: 10.1016/j.envsoft.2014.04.007. URL https://doi.org/10.1016%2Fj.envsoft.2014.04.007.

[86] Lorena Montoya. Geo-data acquisition through mobile GIS and digital video: an urban disaster management perspective☆. *Environmental Modelling & Software*, 18(10):869–876, dec 2003. doi: 10.1016/s1364-8152(03)00105-1. URL https://doi.org/10.1016%2Fs1364-8152%2803%2900105-1.

[87] Ashish Rauniyar, Paal Engelstad, Boning Feng, and Do Van Thanh. Crowdsourcing-based disaster management using fog computing in internet of things paradigm. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, nov 2016. doi: 10.1109/cic.2016.074. URL https://doi.org/10.1109%2Fcic.2016.074.

[88] Geoscience Australia. Sentinel hotspots. https://sentinel.ga.gov.au/, 2018. Accessed: 2018-07-12.

[89] Justin L. Huntington, Katherine C. Hegewisch, Britta Daudert, Charles G. Morton, John T. Abatzoglou, Daniel J. McEvoy, and Tyler Erickson. Climate engine: Cloud computing and visualization of climate and remote sensing data for advanced natural resource monitoring and process understanding. *Bulletin of the American Meteorological Society*, 98(11):2397–2410, nov 2017. doi: 10.1175/bams-d-15-00324.1. URL https://doi.org/10.1175%2Fbams-d-15-00324.1.

[90] Deepak Puthal, Surya Nepal, Rajiv Ranjan, and Jinjun Chen. A secure big data stream analytics framework for disaster management on the cloud. In *2016 IEEE 18th International Conference on High Performance Computing and Communications IEEE 14th International Conference on Smart City IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, dec 2016. doi: 10.1109/hpcc-smartcity-dss.2016.0170. URL https://doi.org/10.1109%2Fhpcc-smartcity-dss.2016.0170.

[91] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication - SIGCOMM '08*. ACM Press, 2008. doi: 10.1145/1402958.1402967. URL https://doi.org/10.1145%2F1402958.1402967.

[92] Stephen J. Visser and Anwar S. Dawood. Real-time natural disasters detection and monitoring from smart earth observation satellite. *Journal of*

Aerospace Engineering, 17(1):10–19, jan 2004. doi: 10.1061/(asce)0893-1321(2004)17:1(10). URL https://doi.org/10.1061%2F%28asce%290893-1321%282004%2917%3A1%2810%29.

[93] Muneo Hori, Tsuyoshi Ichimura, Maddegedara L. L. Wijerathne, and Kouhei Fujita. Application of HPC to earthquake hazard and disaster estimation. In *Sustained Simulation Performance 2014*, pages 203–220. Springer International Publishing, nov 2014. doi: 10.1007/978-3-319-10626-7_17. URL https://doi.org/10.1007%2F978-3-319-10626-7_17.

[94] N. Dimakis, A. Filippoupolitis, and E. Gelenbe. Distributed building evacuation simulator for smart emergency management. *The Computer Journal*, 53(9):1384–1400, feb 2010. doi: 10.1093/comjnl/bxq012. URL https://doi.org/10.1093%2Fcomjnl%2Fbxq012.

[95] C. Rossi, M. H. Heyi, and F. Scullino. A service oriented cloud-based architecture for mobile geolocated emergency services. *Concurrency and Computation: Practice and Experience*, 29(11):e4051, nov 2016. doi: 10.1002/cpe.4051. URL https://doi.org/10.1002%2Fcpe.4051.

[96] Huibo Bi. Emergency navigation without an infrastructure. In *IEEE SENSORS 2014 Proceedings*. IEEE, nov 2014. doi: 10.1109/icsens.2014.6985038. URL https://doi.org/10.1109%2Ficsens.2014.6985038.

[97] Felix Wex, Guido Schryen, Stefan Feuerriegel, and Dirk Neumann. Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, 235(3):697–708, jun 2014. doi: 10.1016/j.ejor.2013.10.029. URL https://doi.org/10.1016%2Fj.ejor.2013.10.029.

[98] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis. New options and insights for survivable transport networks. *IEEE Communications Magazine*, 40(1):34–41, 2002. doi: 10.1109/35.978047. URL https://doi.org/10.1109%2F35.978047.

[99] Stanley W. Gilbert, David T. Butry, Jennifer F. Helgeson, and Robert E. Chapman. Community resilience economic decision guide for buildings and infrastructure systems. Technical report, dec 2015. URL https://doi.org/10.6028%2Fnist.sp.1197.

[100] Teresa Gomes, Janos Tapolcai, Christian Esposito, David Hutchison, Fernando Kuipers, Jacek Rak, Amaro de Sousa, Athanasios Iossifides, Rui Travanca, Joao Andre, Luisa Jorge, Lucia Martins, Patricia Ortiz Ugalde, Alija Pasic, Dimitrios Pezaros, Simon Jouet, Stefano Secci, and Massimo Tornatore. A survey of strategies for communication networks to protect against large-scale natural disasters. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, sep 2016. doi: 10.1109/rndm.2016.7608263. URL https://doi.org/10.1109%2Frndm.2016.7608263.

[101] Toshikazu Sakano, Satoshi Kotabe, Tetsuro Komukai, Tomoaki Kumagai, Yoshitaka Shimizu, Atsushi Takahara, Thuan Ngo, Zubair Md. Fadlullah, Hiroki Nishiyama, and Nei Kato. Bringing movable and deployable networks to disaster areas: development and field test of MDRU. *IEEE Network*, 30(1):86–91, jan 2016. doi: 10.1109/mnet.2016.7389836. URL https://doi.org/10.1109%2Fmnet.2016.7389836.

[102] Gianluca Aloi, Luca Bedogni, Luciano Bononi, Orazio Briante, Marco Di Felice, Valeria Loscrì, Pasquale Pace, Fabio Panzieri, Giuseppe Ruggeri, and Angelo Trotta. STEM-NET: How to deploy a self-organizing network of mobile end-user devices for emergency communication. *Computer Communications*, 60:12–27, apr 2015. doi: 10.1016/j.comcom.2015.01.023. URL https://doi.org/10.1016%2Fj.comcom.2015.01.023.

[103] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, mar 2010. doi: 10.1007/s00502-010-0717-2. URL https://doi.org/10.1007%2Fs00502-010-0717-2.

[104] Stelios Timotheou and Georgios Loukas. Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios. In *Proceedings of the 2009 ACM symposium on Applied Computing*

- *SAC '09*. ACM Press, 2009. doi: 10.1145/1529282.1529542. URL https://doi.org/10.1145%2F1529282.1529542.

[105] Huibo Bi and Erol Gelenbe. Cloud enabled emergency navigation using faster-than-real-time simulation. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, mar 2015. doi: 10.1109/percomw.2015.7134084. URL https://doi.org/10.1109%2Fpercomw.2015.7134084.

[106] M. Pidd, F.N. de Silva, and R.W. Eglese. A simulation model for emergency evacuation. *European Journal of Operational Research*, 90(3):413–419, may 1996. doi: 10.1016/0377-2217(95)00112-3. URL https://doi.org/10.1016%2F0377-2217%2895%2900112-3.

[107] Chen Yong, QF Chen, N Frolova, V Larionov, A Nikolaev, J Pejcoch, S Suchshev, and A Ugarov. Web based decision support tool in order to response to strong earthquakes. *Proceedings of TIEMS2001, Oslo, Norway*, 2001.

[108] C J van Westen. Remote sensing and geographic information systems for natural disaster management. In *Environmental Modelling with GIS and Remote Sensing*, pages 200–226. Taylor & Francis. doi: 10.4324/9780203302217_chapter_10. URL https://doi.org/10.4324%2F9780203302217_chapter_10.

[109] Melinda Laituri and Kris Kodrich. On line disaster response community: People as sensors of high magnitude disasters using internet GIS. *Sensors*, 8(5):3037–3055, may 2008. doi: 10.3390/s8053037. URL https://doi.org/10.3390%2Fs8053037.

[110] Luiz A. Manfré, Eliane Hirata, Janaína B. Silva, Eduardo J. Shinohara, Mariana A. Giannotti, Ana Paula C. Larocca, and José A. Quintanilha. An analysis of geospatial technologies for risk and natural disaster management. *ISPRS International Journal of Geo-Information*, 1(2):166–185, aug 2012. doi: 10.3390/ijgi1020166. URL https://doi.org/10.3390%2Fijgi1020166.

[111] Susan L Cutter. GI science, disasters, and emergency management. *Transactions in GIS*, 7(4):439–446, oct 2003. doi: 10.1111/1467-9671.00157. URL https://doi.org/10.1111%2F1467-9671.00157.

[112] Norman Kerle and Clive Oppenheimer. Satellite remote sensing as a tool in lahar disaster management. *Disasters*, 26(2):140–160, jun 2002. doi: 10.1111/1467-7717.00197. URL https://doi.org/10.1111%2F1467-7717.00197.

[113] Stefan Voigt, Thomas Kemper, Torsten Riedlinger, Ralph Kiefl, Klaas Scholte, and Harald Mehl. Satellite image analysis for disaster and crisis-management support. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1520–1528, jun 2007. doi: 10.1109/tgrs.2007.895830. URL https://doi.org/10.1109%2Ftgrs.2007.895830.

[114] David M. Tralli, Ronald G. Blom, Victor Zlotnicki, Andrea Donnellan, and Diane L. Evans. Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(4):185–198, jun 2005. doi: 10.1016/j.isprsjprs.2005.02.002. URL https://doi.org/10.1016%2Fj.isprsjprs.2005.02.002.

[115] Chia-Ping Shen, Chinburen Jigjidsuren, Sarangerel Dorjgochoo, Chi-Huang Chen, Wei-Hsin Chen, Chih-Kuo Hsu, Jin-Ming Wu, Chih-Wen Hsueh, Mei-Shu Lai, Ching-Ting Tan, Erdenebaatar Altangerel, and Feipei Lai. A data-mining framework for transnational healthcare system. *Journal of Medical Systems*, 36(4):2565–2575, may 2011. doi: 10.1007/s10916-011-9729-7. URL https://doi.org/10.1007%2Fs10916-011-9729-7.

[116] Shaocan Jiang, Chaofan Shen, Yongjie Xiao, and Xiaoying Huang. Using MapReduce for data processing in the cloud for forest pest control. In *2010 First International Conference on Networking and Distributed Computing*. IEEE, oct 2010. doi: 10.1109/icndc.2010.88. URL https://doi.org/10.1109%2Ficndc.2010.88.

[117] Henrik Eriksson, Massimiliano Raciti, Maurizio Basile, Alessandro Cunsolo, Anders Fröberg, Ola Leifler, Joakim Ekberg, and Toomas Timpka. A cloud-based simulation architecture for pandemic influenza simulation. In *AMIA Annual Symposium Proceedings*, volume 2011, page 364. American Medical Informatics Association, 2011.

[118] Quan Zou. Research on cloud computing for disaster monitoring using massive remote sensing data. In *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, apr 2017. doi: 10.1109/icccbda.2017.7951879. URL https://doi.org/10.1109%2Ficccbda.2017.7951879.

[119] Klaus Bohm, Anett Mehler-Bicher, and Dennis Fenchel. GeoVisualAnalytics in the public health sector. In *Proceedings 2011 IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services*. IEEE, jun 2011. doi: 10.1109/icsdm.2011.5969049. URL https://doi.org/10.1109%2Ficsdm.2011.5969049.

[120] Juan Li, Qingrui Li, Samee Ullah Khan, and Nasir Ghani. Community-based cloud for emergency management. In *2011 6th International Conference on System of Systems Engineering*. IEEE, jun 2011. doi: 10.1109/sysose.2011.5966573. URL https://doi.org/10.1109%2Fsysose.2011.5966573.

[121] HF Al-Dahash, S Al-Shammari, U Kulatunga, MD Hardman, et al. Cloud computing for disaster response management stemming from terrorism in iraq. In *13th IPGRC 2017 Full Conference Proceedings*, pages 295–306. University of Salford, 2017.

[122] Kamran Ali, Huan X. Nguyen, Quoc-Tuan Vien, and Purav Shah. Disaster management communication networks: Challenges and architecture design. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, mar 2015. doi: 10.1109/percomw.2015.7134094. URL https://doi.org/10.1109%2Fpercomw.2015.7134094.

[123] Plamena Zlateva, Yoshihiro Hirokawa, and Dimiter Velev. An integrated approach for risk assessment of natural disasters using cloud computing. *International Journal of Trade, Economics and Finance*, pages 134–138, 2013. doi: 10.7763/ijtef.2013.v4.273. URL https://doi.org/10.7763%2Fijtef.2013.v4.273.

[124] Katarina Grolinger, Emna Mezghani, Miriam A.M. Capretz, and Ernesto Exposito. Collaborative knowledge as a service applied to the disaster management domain. *International Journal of Cloud Computing*, 4(1):5, 2015. doi: 10.1504/ijcc.2015.067706. URL https://doi.org/10.1504%2Fijcc.2015.067706.

[125] CSIRO Data61. Eric. https://data61.csiro.au/en/Our-Work/Safety-and-security/Disaster-Management/ERIC, 2018. Accessed: 2018-07-12.

[126] Ronny Klauck, Michael Kirsche, Jan Gaebler, and Sebastian Schoepke. Mobile XMPP and cloud service collaboration: An alliance for flexible disaster management. In *Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2011. doi: 10.4108/icst.collaboratecom.2011.247108. URL https://doi.org/10.4108%2Ficst.collaboratecom.2011.247108.

[127] Siti Hajar Othman and Ghassan Beydoun. Model-driven disaster management. *Information & Management*, 50(5):218–228, jul 2013. doi: 10.1016/j.im.2013.04.002. URL https://doi.org/10.1016%2Fj.im.2013.04.002.

[128] Meikang Qiu, Zhong Ming, Jihe Wang, Laurence T. Yang, and Yang Xiang. Enabling cloud computing in emergency management systems. *IEEE Cloud Computing*, 1(4):60–67, nov 2014. doi: 10.1109/mcc.2014.71. URL https://doi.org/10.1109%2Fmcc.2014.71.

[129] Zubaida Alazawi, Saleh Altowaijri, Rashid Mehmood, and Mohmmad B. Abdljabar. Intelligent disaster management system based on cloud-enabled vehicular networks. In *2011 11th International Conference on ITS Telecommunications*. IEEE, aug 2011. doi: 10.1109/itst.2011.6060083. URL https://doi.org/10.1109%2Fitst.2011.6060083.

[130] Zar Aye, Michel Jaboyedoff, Marc-Henri Derron, and Cees van Westen. Prototype of a web-based participative decision support platform in natural hazards and risk management. *ISPRS International Journal of Geo-Information*, 4(3):1201–1224, jul 2015. doi: 10.3390/ijgi4031201. URL https://doi.org/10.3390%2Fijgi4031201.

[131] Nabil R. Adam, Basit Shafiq, and Robin Staffin. Spatial computing and social media in the context of disaster management. *IEEE Intelligent Systems*, 27(6):90–96, nov 2012. doi: 10.1109/mis.2012.113. URL https://doi.org/10.1109%2Fmis.2012.113.

[132] Mansura Habiba and Shamim Akhter. A cloud based natural disaster management system. In *Grid and Pervasive Computing*, pages 152–161. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-38027-3_16. URL https://doi.org/10.1007%2F978-3-642-38027-3_16.

[133] Nik Bessis, Eleana Asimakopoulou, and Fatos Xhafa. A next generation emerging technologies roadmap for enabling collective computational intelligence in disaster management. *International Journal of Space-Based and Situated Computing*, 1 (1):76, 2011. doi: 10.1504/ijssc.2011.039109. URL https://doi.org/10.1504%2Fijssc.2011.039109.

[134] Eva Pajorová and Ladislav Hluchý. Scientific gateway and visualization tool. In *Computational Intelligence in Security for Information Systems*, pages 246–250. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-21323-6_31. URL https://doi.org/10.1007%2F978-3-642-21323-6_31.

[135] Xiaolu Ji, Bin Chen, Zhou Huang, Zhengwei Sui, and Yu Fang. On the use of cloud computing for geospatial workflow applications. In *2012 20th International Conference on Geoinformatics*. IEEE, jun 2012. doi: 10.1109/geoinformatics.2012.6270263. URL https://doi.org/10.1109%2Fgeoinformatics.2012.6270263.

[136] Jens-Sönke Vöckler, Gideon Juve, Ewa Deelman, Mats Rynge, and Bruce Berriman. Experiences using cloud computing for a scientific workflow application. In *Proceedings of the 2nd international workshop on Scientific cloud computing - ScienceCloud '11*. ACM Press, 2011. doi: 10.1145/1996109.1996114. URL https://doi.org/10.1145%2F1996109.1996114.

[137] N. Golpayegani and M. Halem. Cloud computing for satellite data processing on high end compute clusters. In *2009 IEEE International Conference on Cloud Computing*. IEEE, 2009. doi: 10.1109/cloud.2009.71. URL https://doi.org/10.1109%2Fcloud.2009.71.

[138] Ei Fujioka, Edward Vanden Berghe, Ben Donnelly, Julio Castillo, Jesse Cleary, Chris Holmes, Sean McKnight, and Patrick Halpin. Advancing global marine biogeography research with open-source GIS software and cloud computing. *Transactions in GIS*, 16(2):143–160, apr 2012. doi: 10.1111/j.1467-9671.2012.01310.x. URL https://doi.org/10.1111%2Fj.1467-9671.2012.01310.x.

[139] Claudio De Luca, Ivana Zinno, Michele Manunta, Riccardo Lanari, and Francesco Casu. Large areas surface deformation analysis through a cloud computing p-SBAS approach for massive processing of DInSAR time series. *Remote Sensing of Environment*, 202:3–17, dec 2017. doi: 10.1016/j.rse.2017.05.022. URL https://doi.org/10.1016%2Fj.rse.2017.05.022.

[140] Lizhe Wang, Yan Ma, Jining Yan, Victor Chang, and Albert Y. Zomaya. pipsCloud: High performance cloud computing for remote sensing big data management and processing. *Future Generation Computer Systems*, 78:353–368, jan 2018. doi: 10.1016/j.future.2016.06.009. URL https://doi.org/10.1016%2Fj.future.2016.06.009.

[141] Shaowen Wang. A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers*, 100(3):535–557, jun 2010. doi: 10.1080/00045601003791243. URL https://doi.org/10.1080%2F00045601003791243.

[142] Marc Miska and Masao Kuwahara. Sustainable management of data driven projects. In *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, sep 2010. doi: 10.1109/itsc.2010.5625120. URL https://doi.org/10.1109%2Fitsc.2010.5625120.

[143] Qunying Huang, Chaowei Yang, Doug Nebert, Kai Liu, and Huayi Wu. Cloud computing for geosciences. In *Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems - HPDGIS '10*. ACM Press, 2010. doi: 10.1145/1869692.1869699. URL https://doi.org/10.1145%2F1869692.1869699.

[144] Rahul Ramachandran, Katie Baynes, Kevin Murphy, Alireza Jazayeri, Ian Schuler, and Dan Pilone. Cumulus: Nasa's cloud based distributed active archive center prototype. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 369–372, 2017. doi: 10.1109/IGARSS.2017.8126972.

[145] Harshada Chavan, Rami Alghamdi, and Mohamed F. Mokbel. Towards a GPU accelerated spatial computing framework. In *2016 IEEE 32nd International Conference on Data Engineering Workshops (ICDEW)*. IEEE, may 2016. doi: 10.1109/icdew.2016.7495634. URL https://doi.org/10.1109%2Ficdew.2016.7495634.

[146] Katarina Grolinger, Miriam A.M. Capretz, Emna Mezghani, and Ernesto Exposito. Knowledge as a service framework for disaster data management. In *2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, jun 2013. doi: 10.1109/wetice.2013.48. URL https://doi.org/10.1109%2Fwetice.2013.48.

[147] Dung-Hai Liang, Peirchyi Lii, and Chun-Pin Chang. Environment planning and sustainable development. In *2011 International Conference on Electrical and Control Engineering*. IEEE, sep 2011. doi: 10.1109/iceceng.2011.6058138. URL https://doi.org/10.1109%2Ficeceng.2011.6058138.

[148] Federal Geographic Data Committee. Federal geographic data committee. emerging technologies and the geospatial landscape. a report of the national geospatial advisory committee. https://www.fgdc.gov/ngac/meetings/dec-2016/ngac-paper-emerging-technologies-and-the.pdf, 2018. Accessed: 2018-07-12.

[149] Manzhu Yu, Chaowei Yang, and Yun Li. Big data in natural disaster management: A review. *Geosciences*, 8(5):165, may 2018. doi: 10.3390/geosciences8050165. URL https://doi.org/10.3390%2Fgeosciences8050165.

[150] Tharam Dillon, Chen Wu, and Elizabeth Chang. Cloud computing: Issues and challenges. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010. doi: 10.1109/aina.2010.187. URL https://doi.org/10.1109%2Faina.2010.187.

[151] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. AIDR. In *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*. ACM Press, 2014. doi: 10.1145/2567948.2577034. URL https://doi.org/10.1145%2F2567948.2577034.

[152] Mesay Bejiga, Abdallah Zeggada, Abdelhamid Nouffidj, and Farid Melgani. A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, 9(2):100, jan 2017. doi: 10.3390/rs9020100. URL https://doi.org/10.3390%2Frs9020100.

[153] Austin Cooner, Yang Shao, and James Campbell. Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 haiti

earthquake. *Remote Sensing*, 8(10):868, oct 2016. doi: 10.3390/rs8100868. URL https://doi.org/10.3390%2Frs8100868.

[154] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, mar 2017. doi: 10.1111/mice.12263. URL https://doi.org/10.1111%2Fmice.12263.

[155] K. M. Asim, F. Martínez-Álvarez, A. Basit, and T. Iqbal. Earthquake magnitude prediction in hindukush region using machine learning techniques. *Natural Hazards*, 85(1):471–486, sep 2016. doi: 10.1007/s11069-016-2579-3. URL https://doi.org/10.1007%2Fs11069-016-2579-3.

[156] Jean-Pierre Belaud, Stéphane Negny, Fabrice Dupros, David Michéa, and Benoît Vautrin. Collaborative simulation and scientific big data analysis: Illustration for sustainability in natural hazards management and chemical process engineering. *Computers in Industry*, 65(3):521–535, apr 2014. doi: 10.1016/j.compind.2014.01.009. URL https://doi.org/10.1016%2Fj.compind.2014.01.009.

[157] Ruben Mayer, Harshit Gupta, Enrique Saurez, and Umakishore Ramachandran. The fog makes sense. In *Proceedings of the 2nd International Workshop on Social Sensing*. ACM, apr 2017. doi: 10.1145/3055601.3055614. URL https://doi.org/10.1145%2F3055601.3055614.

[158] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, and R. Buyya. Fog computing: principles, architectures, and applications. In *Internet of Things*, pages 61–75. Elsevier, 2016. doi: 10.1016/b978-0-12-805395-9.00004-6. URL https://doi.org/10.1016%2Fb978-0-12-805395-9.00004-6.

[159] Yuri Demchenko, Zhiming Zhao, Paola Grosso, Adianto Wibisono, and Cees de Laat. Addressing big data challenges for scientific data infrastructure. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, dec 2012. doi: 10.1109/cloudcom.2012.6427494. URL https://doi.org/10.1109%2Fcloudcom.2012.6427494.

[160] Ying Chen, JD Elenee Argentinis, and Griff Weber. IBM watson: How cognitive computing can be applied to big data challenges in life sciences research. *Clinical*

Therapeutics, 38(4):688–701, apr 2016. doi: 10.1016/j.clinthera.2015.12.001. URL https://doi.org/10.1016%2Fj.clinthera.2015.12.001.

[161] Kamalpreet Singh and Ravinder Kaur. Hadoop: Addressing challenges of big data. In *2014 IEEE International Advance Computing Conference (IACC)*. IEEE, feb 2014. doi: 10.1109/iadcc.2014.6779407. URL https://doi.org/10.1109%2Fiadcc.2014.6779407.

[162] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, jan 2015. doi: 10.1109/jproc.2014.2371999. URL https://doi.org/10.1109%2Fjproc.2014.2371999.

[163] Bruce G. Marcot, Matthew P. Thompson, Michael C. Runge, Frank R. Thompson, Steven McNulty, David Cleaves, Monica Tomosy, Larry A. Fisher, and Andrew Bliss. Recent advances in applying decision science to managing national forests. *Forest Ecology and Management*, 285:123–132, dec 2012. doi: 10.1016/j.foreco.2012.08.024. URL https://doi.org/10.1016%2Fj.foreco.2012.08.024.

[164] Qunying Huang, Jing Li, and Zhenlong Li. A geospatial hybrid cloud platform based on multi-sourced computing and model resources for geosciences. *International Journal of Digital Earth*, 11(12):1184–1204, oct 2017. doi: 10.1080/17538947.2017.1385652. URL https://doi.org/10.1080%2F17538947.2017.1385652.

[165] Ujjwal K.C., Saurabh Garg, James Hilton, Jagannath Aryal, and Nicholas Forbes-Smith. Cloud computing in natural hazard modeling systems: Current research trends and future directions. *International Journal of Disaster Risk Reduction*, 38:101188, aug 2019. doi: 10.1016/j.ijdrr.2019.101188. URL https://doi.org/10.1016%2Fj.ijdrr.2019.101188.

[166] Zoltán Ádám Mann. Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms. *ACM Computing Surveys*, 48(1):1–34, sep 2015. doi: 10.1145/2797211. URL https://doi.org/10.1145%2F2797211.

[167] He Huang, Liqiang Wang, Byung Chul Tak, Long Wang, and Chunqiang Tang. CAP3: A cloud auto-provisioning framework for parallel processing using on-demand and spot instances. In *2013 IEEE Sixth International Conference on Cloud Computing*. IEEE, jun 2013. doi: 10.1109/cloud.2013.41. URL https://doi.org/10.1109%2Fcloud.2013.41.

[168] H. C. Tijms, M. H. Van Hoorn, and A. Federgruen. Approximations for the steady-state probabilities in the m/g/c queue. *Advances in Applied Probability*, 13 (1):186–206, mar 1981. doi: 10.2307/1426474. URL https://doi.org/10.2307%2F1426474.

[169] E. J. Messerli. B.s.t.j. brief: Proof of a convexity property of the erlang b formula. *The Bell System Technical Journal*, 51(4):951–953, April 1972. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1972.tb01956.x.

[170] Nectar cloud. https://nectar.org.au/research-cloud/. Accessed: 2018-05-12.

[171] Aws pricing. https://aws.amazon.com/ec2/pricing/on-demand/. Accessed: 2018-09-12.

[172] Nithiapidary Muthuvelu, Christian Vecchiola, Ian Chai, Eswaran Chikkannan, and Rajkumar Buyya. Task granularity policies for deploying bag-of-task applications on global grids. *Future Generation Computer Systems*, 29(1):170–181, jan 2013. doi: 10.1016/j.future.2012.03.022. URL https://doi.org/10.1016%2Fj.future.2012.03.022.

[173] Kostas D. Kalabokidis, Stylianos Gatzojannis, and Spyros Galatsidas. Introducing wildfire into forest management planning: towards a conceptual approach. *Forest Ecology and Management*, 158(1-3):41–50, mar 2002. doi: 10.1016/s0378-1127(00)00715-5. URL https://doi.org/10.1016%2Fs0378-1127%2800%2900715-5.

[174] Mark A. Finney. FARSITE: Fire area simulator-model development and evaluation. Technical report, 1998. URL https://doi.org/10.2737%2Frmrs-rp-4.

[175] Kostas Kalabokidis, Nikolaos Athanasis, Christos Vasilakos, and Palaiologos Palaiologou. Porting of a wildfire risk and fire spread application into a cloud computing environment. *International Journal of Geographical Information Science*, 28(3):541–552, dec 2013. doi: 10.1080/13658816.2013.858257. URL https://doi.org/10.1080%2F13658816.2013.858257.

[176] Long Thai, Blesson Varghese, and Adam Barker. A survey and taxonomy of resource optimisation for executing bag-of-task applications on public clouds. *Future Generation Computer Systems*, 82:1–11, may 2018. doi: 10.1016/j.future.2017.11.038. URL https://doi.org/10.1016%2Fj.future.2017.11.038.

[177] Blesson Varghese and Rajkumar Buyya. Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, 79:849–861, feb 2018. doi: 10.1016/j.future.2017.09.020. URL https://doi.org/10.1016%2Fj.future.2017.09.020.

[178] David Candeia, Ricardo Araujo, Raquel Lopes, and Francisco Brasileiro. Investigating business-driven cloudburst schedulers for e-science bag-of-tasks applications. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. IEEE, nov 2010. doi: 10.1109/cloudcom.2010.67. URL https://doi.org/10.1109%2Fcloudcom.2010.67.

[179] Tekin Bicer, David Chiu, and Gagan Agrawal. Time and cost sensitive data-intensive computing on hybrid clouds. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE, may 2012. doi: 10.1109/ccgrid.2012.95. URL https://doi.org/10.1109%2Fccgrid.2012.95.

[180] Rubing Duan and Radu Prodan. Cooperative scheduling of bag-of-tasks workflows on hybrid clouds. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*. IEEE, dec 2014. doi: 10.1109/cloudcom.2014.58. URL https://doi.org/10.1109%2Fcloudcom.2014.58.

[181] Nithiapidary Muthuvelu, Ian Chai, Eswaran Chikkannan, and Rajkumar Buyya. On-line task granularity adaptation for dynamic grid applications. In *Algorithms and Architectures for Parallel Processing*, pages 266–277. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-13119-6_24. URL https://doi.org/10.1007%2F978-3-642-13119-6_24.

[182] T.F. Ang, W.K. Ng, T.C. Ling, L.Y. Por, and C.S. Liew. A bandwidth-aware job grouping-based scheduling on grid environment. *Information Technology Journal*, 8(3):372–377, mar 2009. doi: 10.3923/itj.2009.372.377. URL https://doi.org/10.3923%2Fitj.2009.372.377.

[183] Ng Wai Keat, Ang Tan Fong, Ling Teck Chaw, and Liew Chee Sun. Scheduling framework for bandwidth-aware job grouping-based scheduling in grid computing. *Malaysian Journal of Computer Science*, 19(2):117–126, 2006.

[184] Nithiapidary Muthuvelu, Ian Chai, and C. Eswaran. An adaptive and parameterized job grouping algorithm for scheduling grid jobs. In *2008 10th International Conference on Advanced Communication Technology*. IEEE, feb 2008. doi: 10.1109/icact.2008.4493929. URL https://doi.org/10.1109%2Ficact.2008.4493929.

[185] Matt P. Plucinski, Andrew L. Sullivan, Chris J. Rucinski, and Mahesh Prakash. Improving the reliability and utility of operational bushfire behaviour predictions in australian vegetation. *Environmental Modelling & Software*, 91:1–12, may 2017. doi: 10.1016/j.envsoft.2017.01.019. URL https://doi.org/10.1016%2Fj.envsoft.2017.01.019.

[186] Miguel G. Cruz and Martin E. Alexander. The 10% wind speed rule of thumb for estimating a wildfire's forward rate of spread in forests and shrublands. *Annals of Forest Science*, 76(2), apr 2019. doi: 10.1007/s13595-019-0829-8. URL https://doi.org/10.1007%2Fs13595-019-0829-8.

[187] Miguel G Cruz, Martin E Alexander, Paulo M Fernandes, Musa Kilinc, and Ângelo Sil. Evaluating the 10% wind speed rule of thumb for estimating a wildfire's forward rate of spread against an extensive independent set of observations. *Environmental Modelling & Software*, 133:104818, 2020.

[188] N. Phillip Cheney, James S. Gould, W. Lachlan McCaw, and Wendy R. Anderson. Predicting fire behaviour in dry eucalypt forest in southern australia. *Forest Ecology and Management*, 280:120–131, sep 2012. doi: 10.1016/j.foreco.2012.06.012. URL https://doi.org/10.1016%2Fj.foreco.2012.06.012.

[189] Miguel G Cruz, Andrew L Sullivan, Rosemary Leonard, Sarah Malkin, Stuart Matthews, Jim S Gould, and ME Alexander. Fire behaviour knowledge in australia: a synthesis of disciplinary and stakeholder knowledge on fire spread prediction capability and application. *Melbourne, Victoria: Bushfire Cooperative Research Centre*, 2014.

[190] Miguel G Cruz, James Stanley Gould, Martin E Alexander, Andrew L Sullivan, W Lachlan McCaw, and Stuart Matthews. *A guide to rate of fire spread models for Australian vegetation*. Australasian Fire and Emergency Service Authorities Council Limited, 2015.

[191] M.P. Plucinski, A.L. Sullivan, and R.J. Hurley. A methodology for comparing the relative effectiveness of suppressant enhancers designed for the direct attack of wildfires. *Fire Safety Journal*, 87:71–79, jan 2017. doi: 10.1016/j.firesaf.2016.12.005. URL https://doi.org/10.1016%2Fj.firesaf.2016.12.005.

[192] Stuart Mathews, Paul Fox-Hughes, Saskia Grootemaat, Jennifer J. Hollis, Belinda J. Kenny, and Sam Sauvage. Australian fire danger rating system: Research prototype. page 384pp, 2019.

[193] NP Cheney, JS Gould, and WR Catchpole. Prediction of fire spread in grasslands. *International Journal of Wildland Fire*, 8(1):1, 1998. doi: 10.1071/wf9980001. URL https://doi.org/10.1071%2Fwf9980001.

[194] Jon B. Marsden-Smedley and Wendy R. Catchpole. Fire modelling in tasmanian buttongrass moorlands. III. dead fuel moisture. *International Journal of Wildland Fire*, 10(2):241, 2001. doi: 10.1071/wf01025. URL https://doi.org/10.1071%2Fwf01025.

[195] Wendy R. Anderson, Miguel G. Cruz, Paulo M. Fernandes, Lachlan McCaw, Jose Antonio Vega, Ross A. Bradstock, Liam Fogarty, Jim Gould, Greg McCarthy, Jon B. Marsden-Smedley, Stuart Matthews, Greg Mattingley, H. Grant Pearce, and Brian W. van Wilgen. A generic, empirical-based model for predicting rate of fire spread in shrublands. *International Journal of Wildland Fire*, 24(4):443, 2015. doi: 10.1071/wf14130. URL https://doi.org/10.1071%2Fwf14130.

[196] ND Burrows, B Ward, A Robinson, et al. Fuel dynamics and fire spread in spinifex grasslands of the western desert. *Proceedings of the Royal Society of Queensland, The*, **115**:69, 2009.

[197] M.G. Cruz, W.L. McCaw, W.R. Anderson, and J.S. Gould. Fire behaviour modelling in semi-arid mallee-heath shrublands of southern australia. *Environmental Modelling & Software*, 40:21–34, feb 2013. doi: 10.1016/j.envsoft.2012.07.003. URL https://doi.org/10.1016%2Fj.envsoft.2012.07.003.

[198] Miguel G. Cruz, Martin E. Alexander, and Paulo A.M. Fernandes. Development of a model system to predict wildfire behaviour in pine plantations. *Australian Forestry*, 71(2):113–121, jan 2008. doi: 10.1080/00049158.2008.10676278. URL https://doi.org/10.1080%2F00049158.2008.10676278.

[199] Alan Grant McArthur. *Control burning in eucalypt forests*. Commonwealth of Australia, Forestry and Timber Bureau, 1962.

[200] Andrew L. Sullivan. Wildland surface fire spread modelling, 1990 - 2007. 1: Physical and quasi-physical models. *International Journal of Wildland Fire*, 18(4):349, 2009. doi: 10.1071/wf06143. URL https://doi.org/10.1071%2Fwf06143.

[201] Andrew L. Sullivan. Wildland surface fire spread modelling, 1990 - 2007. 2: Empirical and quasi-empirical models. *International Journal of Wildland Fire*, 18(4): 369, 2009. doi: 10.1071/wf06142. URL https://doi.org/10.1071%2Fwf06142.

[202] Miguel G. Cruz, James S. Gould, Martin E. Alexander, Andrew L. Sullivan, W. Lachlan McCaw, and Stuart Matthews. Empirical-based models for predicting head-fire rate of spread in australian fuel types. *Australian Forestry*, 78(3):118–158, jul 2015. doi: 10.1080/00049158.2015.1055063. URL https://doi.org/10.1080%2F00049158.2015.1055063.

[203] Miguel Cruz. Quick guide for fire behaviour prediction in semi-arid mallee-heath. *Simulation*, 1(2):3, 2010.

[204] RJ Sneeuwjagt and GB Peet. *Forest fire behaviour tables for Western Australia*. WA Department Conservation & Land Management, 1985.

[205] Noel Phillip Cheney, James Stanley Gould, and I Knight. *A prescribed burning guide for young regrowth forests of silvertop ash*. Research Division, Forestry Commission of New South Wales, 1992.

[206] PJ Byrne. *Prescribed burning in Queensland exotic pine plantations*. Department of Forestry, 1980.

[207] SM Hunt and MJ Crock. Fire behaviour modelling in exotic pine plantations: testing the queensland department of forestry prescribed burning guide mk iii'. *Australian forest research*, 17(3):179–189, 1987.

[208] H. H. Bartelink. A model of dry matter partitioning in trees. *Tree Physiology*, 18 (2):91–101, feb 1998. doi: 10.1093/treephys/18.2.91. URL https://doi.org/10.1093%2Ftreephys%2F18.2.91.

[209] Fanny Sarrazin, Francesca Pianosi, and Thorsten Wagener. Global sensitivity analysis of environmental models: Convergence and validation. *Environmental Modelling & Software*, 79:135–152, may 2016. doi: 10.1016/j.envsoft.2016.02.005. URL https://doi.org/10.1016%2Fj.envsoft.2016.02.005.

[210] Jon Herman and Will Usher. SALib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software*, 2(9):97, jan 2017. doi: 10.21105/joss.00097. URL https://doi.org/10.21105%2Fjoss.00097.

[211] Scott Baker, David Ponniah, and Simon Smith. Survey of risk management in major uk companies. *Journal of Professional Issues in Engineering Education and Practice*, 125(3):94–102, 1999.

[212] Dursun Delen, Leman Tomak, Kazim Topuz, and Enes Eryarsoy. Investigating injury severity risk factors in automobile crashes with predictive analytics and sensitivity analysis methods. *Journal of Transport & Health*, 4:118–131, 2017.

[213] Hani Alyami, Zaili Yang, Ramin Riahi, Stephen Bonsall, and Jin Wang. Advanced uncertainty modelling for container port risk analysis. *Accident Analysis & Prevention*, 123:411–421, 2019.

[214] Elco E Koks, Julie Rozenberg, Conrad Zorn, Mersedeh Tariverdi, Michalis Vousdoukas, SA Fraser, JW Hall, and Stephane Hallegatte. A global multi-hazard risk analysis of road and railway infrastructure assets. *Nature communications*, 10(1): 1–11, 2019.

[215] H Christopher Frey and Sumeet R Patil. Identification and review of sensitivity analysis methods. *Risk analysis*, 22(3):553–578, 2002.

[216] Max D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, may 1991. doi: 10.1080/00401706.1991.10484804. URL https://doi.org/10.1080%2F00401706.1991.10484804.

[217] Henrik Brohus, Peter V Nielsen, Arnkell J Petersen, and Kim Sommerlund-Larsen. Sensitivity analysis of fire dynamics simulation. In *Proceedings of Roomvent*. Citeseer, 2007.

[218] Kathrin Menberg, Yeonsook Heo, Godfried Augenbroe, and Ruchi Choudhary. New extension of morris method for sensitivity analysis of building energy models. *Build. Simul. Optim.*, 2016, 2016.

[219] Chenyang Wang, Minjun Peng, and Genglei Xia. Sensitivity analysis based on morris method of passive system performance under ocean conditions. *Annals of Nuclear Energy*, 137:107067, mar 2020. doi: 10.1016/j.anucene.2019.107067. URL https://doi.org/10.1016%2Fj.anucene.2019.107067.

[220] Lizhi Jiang, Qi Cai, Yongfa Zhang, and Hao Shi. Screening key parameters for passive system based on morris method. In *Volume 6: Thermal-Hydraulics*. American Society of Mechanical Engineers, jul 2017. doi: 10.1115/icone25-67646. URL https://doi.org/10.1115%2Ficone25-67646.

[221] Per-Anders Ekstrom. Eikos: a simulation toolbox for sensitivity analysis in matlab. *FACILIA AB*, 2005.

[222] Il'ya Meerovich Sobol'. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe modelirovanie*, 2(1):112–118, 1990.

[223] John H. Schaibly and Kurt E. Shuler. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. II applications. *The Journal of Chemical Physics*, 59(8):3879–3888, oct 1973. doi: 10.1063/1.1680572. URL https://doi.org/10.1063%2F1.1680572.

[224] Andrea Saltelli, Ksenia Aleksankina, William Becker, Pamela Fennell, Federico Ferretti, Niels Holst, Sushan Li, and Qiongli Wu. Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental modelling & software*, 114:29–39, 2019.

[225] G. Baroni and S. Tarantola. A general probabilistic framework for uncertainty and global sensitivity analysis of deterministic models: A hydrological case study. *Environmental Modelling & Software*, 51:26–34, jan 2014. doi: 10.1016/j.envsoft.2013.09.022. URL https://doi.org/10.1016%2Fj.envsoft.2013.09.022.

[226] E. Borgonovo. A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92(6):771–784, jun 2007. doi: 10.1016/j.ress.2006.04.015. URL https://doi.org/10.1016%2Fj.ress.2006.04.015.

[227] Polynomial chaos for sensitivity analysis in wildfire modelling. In *Syme, G., Hatton MacDonald, D., Fulton, B. and Piantadosi, J. (eds) MODSIM2017, 22nd International Congress on Modelling and Simulation.* Modelling and Simulation Society of Australia and New Zealand (MSSANZ), Inc., dec 2017. doi: 10.36334/modsim.2017.h10.hilton3. URL https://doi.org/10.36334%2Fmodsim.2017.h10.hilton3.

[228] Huibin Liu, Wei Chen, and Agus Sudjianto. Relative entropy based method for global and regional sensitivity analysis in probabilistic design. In *Volume 1: 30th Design Automation Conference*. ASMEDC, jan 2004. doi: 10.1115/detc2004-57500. URL https://doi.org/10.1115%2Fdetc2004-57500.

[229] K. Thonicke, A. Spessa, I. C. Prentice, S. P. Harrison, L. Dong, and C. Carmona-Moreno. The influence of vegetation, fire spread and fire behaviour on biomass burning and trace gas emissions: results from a process-based model. *Biogeosciences*, 7(6):1991–2011, jun 2010. doi: 10.5194/bg-7-1991-2010. URL https://doi.org/10.5194%2Fbg-7-1991-2010.

[230] Jose Gomez-Dans. A sensitivity analysis study of the SPITFIRE fire model. oct 2018. doi: 10.31223/osf.io/uev28. URL https://doi.org/10.31223%2Fosf.io%2Fuev28.

[231] R. Salvador, J. Piñol, S. Tarantola, and E. Pla. Global sensitivity analysis and scale effects of a fire propagation model used over mediterranean shrublands. *Ecological Modelling*, 136(2-3):175–189, jan 2001. doi: 10.1016/s0304-3800(00)00419-1. URL https://doi.org/10.1016%2Fs0304-3800%2800%2900419-1.

[232] Farkhondeh Khorashadi Zadeh, Jiri Nossent, Fanny Sarrazin, Francesca Pianosi, Ann van Griensven, Thorsten Wagener, and Willy Bauwens. Comparison of variance-based and moment-independent global sensitivity analysis approaches by application to the SWAT model. *Environmental Modelling & Software*, 91:210–222, may 2017. doi: 10.1016/j.envsoft.2017.02.001. URL https://doi.org/10.1016%2Fj.envsoft.2017.02.001.

[233] D. M. Hamby. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental Monitoring and Assessment*, 32(2):135–154, sep 1994. doi: 10.1007/bf00547132. URL https://doi.org/10.1007%2Fbf00547132.

[234] A. Saltelli, S. Tarantola, and K. P.-S. Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, feb 1999. doi: 10.1080/00401706.1999.10485594. URL https://doi.org/10.1080%2F00401706.1999.10485594.

[235] Richard C Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest and Range Experiment Station, Forest Service, United . . . , 1972.

[236] ND Burrows. *Experimental development of a fire management model for jarrah (Eucalyptus marginata) forest*. PhD thesis, PhD Thesis, Department of Forestry, Australian National University, Canberra . . . , 1994.

[237] Elmar Plischke, Giovanni Rabitti, and Emanuele Borgonovo. Computing shapley effects for sensitivity analysis. *arXiv preprint arXiv:2002.12024*, 2020.

[238] AG McArthur. Weather and grassland fire behaviour. commonwealth department of national development. *Forestry and Timber Bureau Leaflet*, 100, 1966.

[239] Francesca Pianosi, Fanny Sarrazin, and Thorsten Wagener. A matlab toolbox for global sensitivity analysis. *Environmental Modelling & Software*, 70:80–85, aug 2015. doi: 10.1016/j.envsoft.2015.04.009. URL https://doi.org/10.1016%2Fj.envsoft.2015.04.009.

[240] Art B Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014. URL https://doi.org/10.1137/130936233.

[241] Yaning Liu, M. Yousuff Hussaini, and Giray Ökten. Global sensitivity analysis for the rothermel model based on high-dimensional model representation. *Canadian Journal of Forest Research*, 45(11):1474–1479, nov 2015. doi: 10.1139/cjfr-2015-0148. URL https://doi.org/10.1139%2Fcjfr-2015-0148.

[242] Giray Ökten and Yaning Liu. Randomized quasi-monte carlo methods in global sensitivity analysis. *Reliability Engineering & System Safety*, 210:107520, jun

2021. doi: 10.1016/j.ress.2021.107520. URL https://doi.org/10.1016%2Fj.ress.2021.107520.

[243] S. Tarantola, N. Giglioli, J. Jesinghaus, and A. Saltelli. Can global sensitivity analysis steer the implementation of models for environmental assessments and decision-making? *Stochastic Environmental Research and Risk Assessment (SERRA)*, 16(1):63–76, feb 2002. doi: 10.1007/s00477-001-0085-x. URL https://doi.org/10.1007%2Fs00477-001-0085-x.

[244] Eric Jacquier, Nicholas G. Polson, and Peter E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 12(4):371–389, oct 1994. doi: 10.1080/07350015.1994.10524553. URL https://doi.org/10.1080%2F07350015.1994.10524553.

[245] Edward P. Herbst and Frank Schorfheide. *Bayesian Estimation of DSGE Models*. Princeton University Press, dec 2015. doi: 10.23943/princeton/9780691161082.001.0001. URL https://doi.org/10.23943%2Fprinceton%2F9780691161082.001.0001.

[246] Jeremy E. Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, aug 2004. doi: 10.1111/j.1467-9868.2004.05304.x. URL https://doi.org/10.1111%2Fj.1467-9868.2004.05304.x.

[247] A. Yegnan, D.G. Williamson, and A.J. Graettinger. Uncertainty analysis in air dispersion modeling. *Environmental Modelling & Software*, 17(7):639–649, nov 2002. doi: 10.1016/s1364-8152(02)00026-9. URL https://doi.org/10.1016%2Fs1364-8152%2802%2900026-9.

[248] George Kuczera and Eric Parent. Monte carlo assessment of parameter uncertainty in conceptual catchment models: the metropolis algorithm. *Journal of Hydrology*, 211(1-4):69–85, nov 1998. doi: 10.1016/s0022-1694(98)00198-x. URL https://doi.org/10.1016%2Fs0022-1694%2898%2900198-x.

[249] Catherine Freissinet, Michel Vauclin, and Marc Erlich. Comparison of first-order analysis and fuzzy set approach for the evaluation of imprecision in a pesticide groundwater pollution screening model. *Journal of Contaminant Hydrology*, 37

(1-2):21–43, apr 1999. doi: 10.1016/s0169-7722(98)00163-6. URL https://doi.org/10.1016%2Fs0169-7722%2898%2900163-6.

[250] Andrew D. Richardson and David Y. Hollinger. Statistical modeling of ecosystem respiration using eddy covariance data: Maximum likelihood parameter estimation, and monte carlo simulation of model and parameter uncertainty, applied to three simple models. *Agricultural and Forest Meteorology*, 131(3-4):191–208, aug 2005. doi: 10.1016/j.agrformet.2005.05.008. URL https://doi.org/10.1016%2Fj.agrformet.2005.05.008.

[251] Andreas Bachmann and Britta Allgöwer. Uncertainty propagation in wildland fire behaviour modelling. *International Journal of Geographical Information Science*, 16(2):115–127, mar 2002. doi: 10.1080/13658810110099080. URL https://doi.org/10.1080%2F13658810110099080.

[252] R. Cibin, K. P. Sudheer, and I. Chaubey. Sensitivity and identifiability of stream flow generation parameters of the SWAT model. *Hydrological Processes*, 24(9): 1133–1148, apr 2010. doi: 10.1002/hyp.7568. URL https://doi.org/10.1002%2Fhyp.7568.

[253] K. Holvoet, A. van Griensven, P. Seuntjens, and P.A. Vanrolleghem. Sensitivity analysis for hydrology and pesticide supply towards the river in SWAT. *Physics and Chemistry of the Earth, Parts A/B/C*, 30(8-10):518–526, jan 2005. doi: 10.1016/j.pce.2005.07.006. URL https://doi.org/10.1016%2Fj.pce.2005.07.006.

[254] Kati L. White and Indrajeet Chaubey. SENSITIVITY ANALYSIS, CALIBRATION, AND VALIDATIONS FOR a MULTISITE AND MULTIVARIABLE SWAT MODEL. *Journal of the American Water Resources Association*, 41(5): 1077–1089, oct 2005. doi: 10.1111/j.1752-1688.2005.tb03786.x. URL https://doi.org/10.1111%2Fj.1752-1688.2005.tb03786.x.

[255] Xiao Li, George Hadjisophocleous, and Xiao qian Sun. Sensitivity and uncertainty analysis of a fire spread model with correlated inputs. *Procedia Engineering*, 211: 403–414, 2018. doi: 10.1016/j.proeng.2017.12.029. URL https://doi.org/10.1016%2Fj.proeng.2017.12.029.

[256] Thorsten Wagener and Joshua Kollat. Numerical and visual evaluation of hydrological and environmental models using the monte carlo analysis toolbox. *Environmental Modelling & Software*, 22(7):1021–1033, jul 2007. doi: 10.1016/j.envsoft.2006.06.017. URL https://doi.org/10.1016%2Fj.envsoft.2006.06.017.

[257] Anthony Frank D'Augustine. *MATLODE: A MATLAB ODE solver and sensitivity analysis toolbox*. PhD thesis, Virginia Tech, 2018.

[258] Pamphile T. Roy, Sophie Ricci, Romain Dupuis, Robin Campet, Jean-Christophe Jouhaud, and Cyril Fournier. BATMAN: Statistical analysis for expensive computer codes made easy. *The Journal of Open Source Software*, 3(21):493, jan 2018. doi: 10.21105/joss.00493. URL https://doi.org/10.21105%2Fjoss.00493.

[259] Anne Dutfoy, Ivan Dutka-Malen, Alberto Pasanisi, Régis Lebrun, Fabien Mangeant, Jayant Sen Gupta, Maurice Pendola, and Thierry Yalamas. Openturns, an open source initiative to treat uncertainties, risks' n statistics in a structured industrial approach. 2009.

[260] Stefano Tarantola and William Becker. SIMLAB software for uncertainty and sensitivity analysis. In *Handbook of Uncertainty Quantification*, pages 1–21. Springer International Publishing, 2015. doi: 10.1007/978-3-319-11259-6_61-1. URL https://doi.org/10.1007%2F978-3-319-11259-6_61-1.

[261] Iooss Bertrand, Janon Alexandre, and Pujol Gilles. *Senstivity Analysis*, 2020. R package version 1.18.0, https://CRAN.R-project.org/package=sensitivity.

[262] Bryan Stanfill, Henrike Mielenz, David Clifford, and Peter Thorburn. Simple approach to emulating complex computer models for global sensitivity analysis. *Environmental Modelling & Software*, 74:140–155, dec 2015. doi: 10.1016/j.envsoft.2015.09.011. URL https://doi.org/10.1016%2Fj.envsoft.2015.09.011.

[263] B.A Keating, P.S Carberry, G.L Hammer, M.E Probert, M.J Robertson, D Holzworth, N.I Huth, J.N.G Hargreaves, H Meinke, Z Hochman, G McLean, K Verburg, V Snow, J.P Dimes, M Silburn, E Wang, S Brown, K.L Bristow, S Asseng, S Chapman, R.L McCown, D.M Freebairn, and C.J Smith. An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, 18(3-4):267–288, jan 2003. doi: 10.1016/s1161-0301(02)00108-9. URL https://doi.org/10.1016%2Fs1161-0301%2802%2900108-9.

[264] Razi Sheikholeslami, Saman Razavi, Hoshin V. Gupta, William Becker, and Amin Haghnegahdar. Global sensitivity analysis for high-dimensional problems: How to objectively group factors and measure robustness and convergence while reducing computational cost. *Environmental Modelling & Software*, 111:282–299, jan 2019. doi: 10.1016/j.envsoft.2018.09.002. URL https://doi.org/10.1016%2Fj.envsoft.2018.09.002.

[265] Francesca Pianosi and Thorsten Wagener. Distribution-based sensitivity analysis from a generic input-output sample. *Environmental Modelling & Software*, 108: 197–207, oct 2018. doi: 10.1016/j.envsoft.2018.07.019. URL https://doi.org/10.1016%2Fj.envsoft.2018.07.019.

[266] E. Borgonovo, X. Lu, E. Plischke, O. Rakovec, and M. C. Hill. Making the most out of a hydrological model data set: Sensitivity analyses to open the model black-box. *Water Resources Research*, 53(9):7933–7950, sep 2017. doi: 10.1002/2017wr020767. URL https://doi.org/10.1002%2F2017wr020767.

[267] O. Rakovec, M. C. Hill, M. P. Clark, A. H. Weerts, A. J. Teuling, and R. Uijlenhoet. Distributed evaluation of local sensitivity analysis (DELSA), with application to hydrologic models. *Water Resources Research*, 50(1):409–426, jan 2014. doi: 10.1002/2013wr014063. URL https://doi.org/10.1002%2F2013wr014063.

[268] MICHAEL ELDRED, ANTHONY GIUNTA, BART VAN BLOEMEN WAANDERS, JR STEVEN WOJTKIEWICZ, WILLIAM HART, and MARIO ALLEVA. DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis version 3.0 reference manual. Technical report, apr 2001. URL https://doi.org/10.2172%2F798492.

[269] Ujjwal KC, Saurabh Garg, and James Hilton. An efficient framework for ensemble of natural disaster simulations as a service. *Geoscience Frontiers*, 11(5):1859–1873, sep 2020. doi: 10.1016/j.gsf.2020.02.002. URL https://doi.org/10.1016%2Fj.gsf.2020.02.002.

[270] Francesca Campolongo and Andrea Saltelli. Sensitivity analysis of an environmental model: an application of different analysis methods. *Reliability Engineering &*

System Safety, 57(1):49–69, jul 1997. doi: 10.1016/s0951-8320(97)00021-5. URL https://doi.org/10.1016%2Fs0951-8320%2897%2900021-5.

[271] F. Pianosi, T. Wagener, J. Rougier, J. Freer, and J. Hall. Sensitivity analysis of environmental models: A systematic review with practical workflow. In *Vulnerability, Uncertainty, and Risk*. American Society of Civil Engineers, jun 2014. doi: 10.1061/9780784413609.030. URL https://doi.org/10.1061%2F9780784413609.030.

[272] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17, apr 1996. doi: 10.1016/0951-8320(96)00002-6. URL https://doi.org/10.1016%2F0951-8320%2896%2900002-6.

[273] Andrea Saltelli and Ricardo Bolado. An alternative way to compute fourier amplitude sensitivity test (FAST). *Computational Statistics & Data Analysis*, 26(4):445–460, feb 1998. doi: 10.1016/s0167-9473(97)00043-1. URL https://doi.org/10.1016%2Fs0167-9473%2897%2900043-1.

[274] George Qian and Adam Mahdi. Sensitivity analysis methods in the biomedical sciences. *Mathematical Biosciences*, 323:108306, may 2020. doi: 10.1016/j.mbs.2020.108306. URL https://doi.org/10.1016%2Fj.mbs.2020.108306.

[275] Bertrand Iooss and Paul Lemaître. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems*, pages 101–122. Springer US, 2015. doi: 10.1007/978-1-4899-7547-8_5. URL https://doi.org/10.1007%2F978-1-4899-7547-8_5.

[276] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145(2):280–297, may 2002. doi: 10.1016/s0010-4655(02)00280-1. URL https://doi.org/10.1016%2Fs0010-4655%2802%2900280-1.

[277] H.J. Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967. doi: 10.1109/proc.1967.5962. URL https://doi.org/10.1109%2Fproc.1967.5962.

[278] Tasmania Fire Service. *STATE FIRE COMMISSION ANNUAL REPORT*. Oct 2019.

[279] Water Tasmanian Department of Primary Industries, Parks, Environment. Tasmanian Vegetation Monitoring, and Mapping Program. Tasveg 3.0. 2013.

[280] AG McARTHUR. Fire behavior in eucalypt forest. canberra, department of development. *Forestry and Timber Bureau*, 1967.

[281] JB Marsden-Smedley and WR Catchpole. Fire behaviour modelling in tasmanian buttongrass moorlands .II. fire behaviour. *International Journal of Wildland Fire*, 5(4):215, 1995. doi: 10.1071/wf9950215. URL https://doi.org/10.1071%2Fwf9950215.

[282] Mark D. Hill. Amdahl's law in the multicore era. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture*. IEEE, feb 2008. doi: 10.1109/hpca.2008.4658638. URL https://doi.org/10.1109%2Fhpca.2008.4658638.

[283] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling & Software*, 22(10):1509–1518, oct 2007. doi: 10.1016/j.envsoft.2006.10.004. URL https://doi.org/10.1016%2Fj.envsoft.2006.10.004.

[284] Lisa Richards, Nigel Brew, and Lizzie Smith. 2019–20 australian bushfires—frequently asked questions: a quick guide. *Research Paper Series 2019*, 20:12, 2020.

[285] J Ruffault, Vincent Moron, RM Trigo, and T Curt. Daily synoptic conditions associated with large fire occurrence in mediterranean france: evidence for a wind-driven fire regime. *International Journal of Climatology*, 37(1):524–533, 2017.

[286] Ana Russo, Célia M Gouveia, Patrícia Páscoa, Carlos C DaCamara, Pedro M Sousa, and Ricardo M Trigo. Assessing the role of drought events on wildfires in the iberian peninsula. *Agricultural and Forest Meteorology*, 237:50–59, 2017.

[287] Ricardo M Trigo, Pedro M Sousa, Mário G Pereira, Domingo Rasilla, and Célia M Gouveia. Modelling wildfire activity in iberia with different atmospheric circulation weather types. *International Journal of Climatology*, 36(7):2761–2778, 2016.

[288] A Cardil, DM Molina, and LN Kobziar. Extreme temperature days and their potential impacts on southern europe. *Natural Hazards and Earth System Sciences*, 14(11):3005–3014, 2014.

[289] Christopher D O'Connor, David E Calkin, and Matthew P Thompson. An empirical machine learning method for predicting potential fire control locations for pre-fire planning and operational fire management. *International journal of wildland fire*, 26(7):587–597, 2017.

[290] Miguel M Pinto, Carlos C DaCamara, Isabel F Trigo, Ricardo M Trigo, and K Feridun Turkman. Fire danger rating over mediterranean europe based on fire radiative power derived from meteosat. *Natural Hazards and Earth System Sciences*, 18(2):515, 2018.

[291] Emilio Chuvieco, Inmaculada Aguado, Javier Salas, Mariano García, Marta Yebra, and Patricia Oliva. Satellite remote sensing contributions to wildland fire science and management. *Current Forestry Reports*, 6(2):81–96, apr 2020. doi: 10.1007/s40725-020-00116-5. URL https://doi.org/10.1007%2Fs40725-020-00116-5.

[292] Raphael A. Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[293] Paul J. Smith and Donald A. Berry. Statistics: A bayesian perspective. *The American Statistician*, 51(3):293, aug 1997. doi: 10.2307/2684909. URL https://doi.org/10.2307%2F2684909.

[294] Ujjwal KC, Saurabh Garg, James Hilton, and Jagannath Aryal. Fire simulation data set for tasmania, 2021. URL https://data.csiro.au/collections/collection/CIcsiro:49133v1/DItrue.