Dependability Evaluation of Wireless Sensor Networks

A Thesis

submitted in fulfillment of the requirements for the award of degree

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Jasminder Kaur Sandhu

(Registration No.: 901411005)

under the guidance of

Dr. Anil Kumar Verma, Professor

Dr. Prashant Singh Rana, Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala - 147004

February 2019

Certificate

I, Jasminder Kaur Sandhu, Registration No. 901411005, hereby declare that the work which is being presented in this thesis entitled, **"Dependability Evaluation of Wireless Sensor Networks"** in fulfillment of the requirement for the award of **"Doctor of Philosophy"** submitted at the Computer Science and Engineering Department of Thapar Institute of Engineering and technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Anil Kumar Verma and Dr. Prashant Singh Rana, refers other research works which have been duly listed in the reference section. The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Jasminder Sandhu

(Jasminder Kaur Sandhu)

Registration No. 901411005

This is to certify that the above statements made by the candidate are correct and true to the best of my knowledge.

Verified by:

Anil K. Verma

(Dr. Anil Kumar Verma)

Professor, Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala - 147004

(Dr. Prashant Singh Rana)

Assistant Professor, Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala - 147004

Acknowledgement

Working on this research problem of Dependability Evaluation in Wireless Sensor Networks has been fascinating and extremely rewarding. I would like to thank a number of people who have contributed to the final result in many different ways. To commence with, I pay my obeisance to God, the almighty to have bestowed upon me good health, courage, inspiration, zeal and the light. After God, I express my sincere and deepest gratitude to my supervisor, Dr. Anil Kumar Verma (Professor at Computer Science and Engineering Department), who plowed through several preliminary versions of my text, making critical suggestions and posing challenging questions. His expertise, invaluable guidance, constant encouragement, affectionate attitude, understanding, patience and healthy criticism added considerably to my experience. Without his continual inspiration, it would not have been possible to complete this study. I cannot think of a better supervisor to have. I also express my warmest gratitude to my supervisor Dr. Prashant Singh Rana (Assistant Professor at Computer Science and Engineering Department), for introducing me to the interesting world of Machine Learning and Data Analytics. His guidance helped me throughout my research. The contribution of my supervisors in this research work is beyond their role as an academic supervisor and also constant support on a personal level without which this journey of my research would never be completed. And for this, I am truly grateful. Special thanks to them for their amazing support. I owe a great debt of gratitude to my supervisors.

My deep regards to Prof. Prakash Gopalan, Director, TIET for all the facilities provided to me during my research work. I am thoroughly grateful and highly indebted to Dr. Rafat Siddique (Dean of Research and Sponsored Projects) for his constant support. I am grateful to Dr. Maninder Singh (Head of Computer Science and Engineering Department) for being a source of motivation. I am also indebted to my mentor Dr. Rajneesh Randhawa (Assistant Professor at Department of Computer Science, Punjabi University) for the excellent learning opportunities provided by her. I am thankful to my

friends Ruhi Saxena, Dharmendra Mahato, Sanjeev Rao, Palki Sharma, Meena Pundir, Shiwani Garg, and Amarvir Singh (Sorry, if I missed some of you) for providing a stimulating and fun environment in which to learn and grow.

I feel a deep sense of gratitude to my father Late Mr. M.S.Sandhu who formed part of my vision and taught me the good things that matter in life. The happy memory of my father still provides persistent inspiration for my journey in this life. Words prove a meager media to write down my feelings for my mother Late Mrs. Manjeet Kaur Sandhu, sister Kamaljit Sandhu, brothers Virinder Singh Sandhu, Ravinder Singh Sandhu, and Gagandeep Singh Sandhu, for providing me constant encouragement, divine presence and supporting me spiritually throughout. A very special thanks to my beloved support system Shubhi Sandhu, who provided me a lot of support and encouragement over the years. Though she is not a scientist, she has always supported me with my research work. Additionally, I would like to give a special acknowledgement to my loving family members Rupinder Sandhu, nephews Eshaan, Simar, niece Harleen and Anhad for supporting me, and for having faith in me at every step.

...Virinder Singh Sandhu, "Bro", your stubbornness has inspired me to hold tight to my position no matter how complicated it gets, and your brotherhood has been a welcome release when I get fed up with the philosophy babble.

Abstract

The performance of a network is dependent on the qualitative and quantitative features, which are closely tied to the Quality of Service (QoS). The QoS determines the characteristics of a network required for its effective functioning. The QoS encompasses many aspects of the network such as dependability, scalability, fault recovery, energy efficiency, packet loss ratio. The most important aspect of QoS is dependability and hence dependability evaluation of a network is obligatory to investigate the perilous aspect that affects the faultless functioning of the network. This research work focuses on the reliability and security aspect of dependability. Reliability is defined as the "measure of the continuity of correct service". It is the most quantifiable feature of network design. Security is defined as the "judgment of how likely it is that the network can resist accidental or deliberate intrusions".

The Wireless Sensor Networks (WSNs) are capable of monitoring the dynamically changing environment in a particular timespan. The data collected by this network consists of unexpected and complex patterns. To understand these patterns, Machine Learning plays a vital role. ML algorithms facilitate in discovering important correlations in the collected data and hence provide improved deployment strategy. The main focus of this research work is dedicated to the analysis of various dependability evaluation techniques in WSN. Also, an ML-based framework is proposed to optimize the data flow parameter of the network. The data flow is a vital parameter that affects the QoS of a network.

This dissertation proposes a novel ML-based framework, which predicts the overall reliability of WSN in terms of performance metrics such as, sent packets, received packets, packets forfeit, packet delivery ratio, and throughput. Ten Machine Learning models namely, Cubist, Random Forest, Support Vector Machine (SVM), Neural

Networks, Weka Lazy Model, Conditional Inference Tree, Bayesian Regularized Neural Networks, Bagged Multivariate Adaptive Regression Splines, Bagged Classification and Regression Trees, and Tree Model from Genetic Algorithms are used to predict the Data Flow, Number of Nodes and Protocol Name. Also, an ensemble model is proposed, which yields an optimum result for prediction.

Further, we considered the security aspect taking into account the flooding attack on the WSN. The node deployment has been carried out in two ways: randomized and normalized deployment. Also, an Intrusion Detection System (IDS) is designed to diagnose any suspicious activity in the network traffic flow. This IDS analyzes the traffic patterns both for the randomized and normalized deployment of sensor nodes. Different Machine Learning approaches namely, Linear Tree, Decision Tree, Extreme Learning Machine, Tree Model from Genetic Algorithms, Generalized Additive Model, Model Tree, Projection Pursuit Regression, Bayesian Regularized Neural Network, PartyKit, Generalized Linear Model, and Linear Regression are used for predicting data flow patterns. These models perform differently according to the training-testing partition size.

Also, traffic flow prediction has been carried out with the help of intelligent soft computing techniques such as, Neural Network, Bayesian Regularized Neural Network, Neural Network using Model Averaging, Multi-Layer Perceptron, Multi-Layer Perceptron with Multiple Layers, Quantile Regression Neural Network, and Stacked Autoencoder Deep Neural Network. These methods prove to be very effective and substantially enhance the prediction efficiency.

Keywords:- Wireless Sensor Networks, Dependability, Reliability, Security, Machine Learning.

Contents

List of Figures

List of Tables

Abbreviations

ACO	**A**nt **C**olony **O**ptimization
AODV	**A**d-hoc **O**n-Demand **D**istance **V**ector
APL	**A**verage **P**ath **L**ength
BPSN	**B**attery **P**owered **S**ensor **N**odes
BRNN	**B**ayesian **R**egularized **N**eural **N**etwork
CART	**C**lassification **A**nd **R**egression **T**rees
CH	**C**luster **H**ead
CIP	**C**ritical **I**nfrastructure **P**rotection
CSP	**C**ommunicating **S**equential **P**rocesses
DSR	**D**ynamic **S**ource **R**outing
DF	**D**ata **F**low
D-FNN	**D**ynamic-**F**uzzy **N**eural **N**etwork
EHSN	**E**nergy **H**arvesting **S**ensor **N**odes
ELM	**E**xtreme **L**earning **M**achine
GSPN	**G**eneralized **S**tochastic **P**etri **N**ets
IoT	**I**nternet **o**f **T**hings
IDS	**I**ntrusion **D**etection **S**ystem
Kbps	**K**ilo**b**its **p**er **s**econd
LOGOS	**L**ights **O**ut **G**round **O**perating **S**ystem
MARS	**M**ultivariate **A**daptive **R**egression **S**plines
MFST	**M**achine **L**earning
ML	**M**achine **L**earning

MTBF	**M**ean **T**ime **B**etween **F**ailure
MTTR	**M**ean **T**ime **T**o **R**epair
NFR	**N**on-**F**unctional **R**equirement
NN	**N**umber **o**f **N**odes
NORMDIS	**NORM**al **DIS**tribution
NS-2.35	**N**etwork **S**imulator-**2.35**
PCA	**P**rincipal **C**omponent **A**nalysis
PDR	**P**acket **D**elivery **R**atio
PF	**P**ackets **F**orfeit
PN	**P**rotocol **N**ame
PSO	**P**article **S**warm **O**ptimization
RA	**R**outing **A**gents
RAS	**R**eliability, **A**vailability and **S**erviceability
RO	**R**outing **O**verhead
RP	**R**eceived **P**ackets
RUL	**R**emaining **U**seful **L**ife
RMSE	**R**oot **M**ean **S**quare **E**rror
SDN	**S**oftware-**D**efined **N**etworks
SP	**S**ent **P**ackets
SPN	**S**tochastic **P**etri **N**ets
SOA	**S**ervice-**O**riented **A**rchitecture
SOM	**S**elf-**O**rganized **M**ap
SRN	**S**tochastic **R**eward **N**ets
SSWSN	**S**mall-**S**cale **W**ireless **S**ensor **N**etwork
SVM	**S**upport **V**ector **M**achine
TH	**T**hroughput
UML	**U**nified **M**odeling **L**anguage
QoS	**Q**uality **o**f **S**ervice
WSN	**W**ireless **S**ensor **N**etwork
XML	**E**Xtensible **M**arkup **L**anguage

Chapter 1

Introduction

"A journey of a thousand miles must begin with a single step."

-Lau Tzu

The Wireless Sensor Network (WSN) supports a multitude of applications such as health-care monitoring, military surveillance, and area monitoring [1]. There are many challenges for the successful implementation of WSN, notable being security, communication between nodes, placement of nodes, routing, maintenance. Further, this network must be dependable to a certain extent, for the successful adaptability in a particular environment [2, 3]. Dependability as a parameter has been explored by various researchers.

The preliminary works demonstrate the simulation and modeling, moving to application-oriented research and at present are being influenced by prediction techniques using Machine Learning. Although different researchers have proposed different metrics for dependability. Therefore, there is a dire need to discuss the methods and the metrics for dependability evaluation. This chapter provides a detailed insight into the WSN; Quality of Service (QoS); Dependability and its Characteristics, Negative Traits, and Positive Traits.

1.1 Wireless Sensor Networks

The WSN comprises of numerous sensor nodes set up in the region of interest to realize a pre-defined task assigned by the user or according to a particular application domain [4, 5]. The data is communicated between the source and destination node (sink node or base station) using a technique known as hopping. When intermediate nodes are involved in this communication, it is termed as multiple hops. Whereas, when transmission takes place directly between source and destination in one hop, it is known as single hop (Kindly refer to Figure 1.1). The sensor nodes transmit the data to the Clustering Node or Cluster Head (CH). The CH then forwards it to the final processing node where data aggregation or data agglomeration [6] task is carried out. This processed data is then stored at a local workstation and is utilized by end-users with internet connectivity.

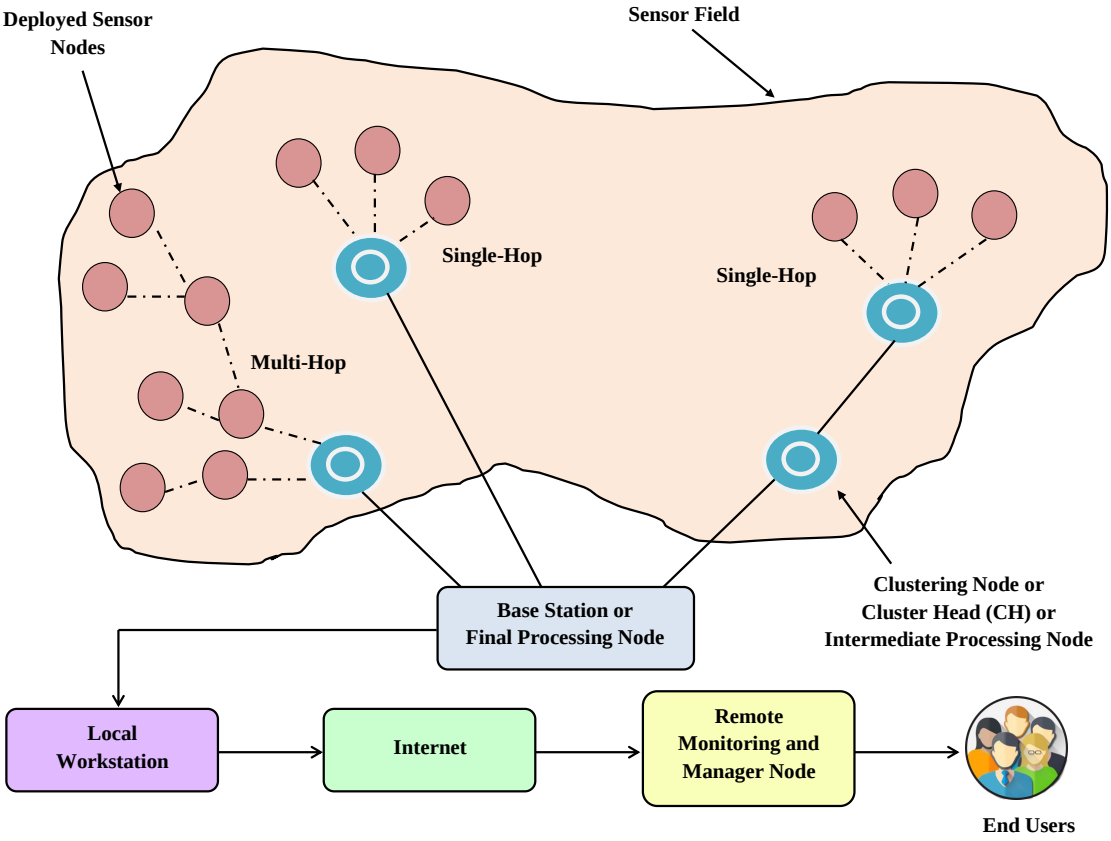


FIGURE 1.1: Key Components of Communication in a Wireless Sensor Network

1.2 Taxonomy on Quality of Service and Dependability Evaluation

The Computing Systems are characterized by five fundamental properties: Functionality, Usability, Performance, Cost, and Quality of Service (QoS) [7]. The QoS ensures quality transmission along with better performance, fewer delays, and minimum packet loss while transmitting data through a communication channel. The QoS is an umbrella term for dependability and includes various network performance parameters that entitle applications to provide requested services to the user [8]. The dependability evaluation of a network is the potential to provide little functionality, those can be reasonably vested. Calibrating trust in a network to deliver the services at that particular time is called the dependability of a network. Achieving dependable data delivery enhances the QoS of the network.

The **QoS** is categorized on the basis of user-described and low-level features as represented in Figure 1.2. As the name suggests, the user-described features can be controlled or optimized by the users, whereas, low-level features cannot be customized by users of the network [9, 10]. Some of these features come under dependability and are explained in the text further (to avoid redundancy). Other features include:

- **Deadline** refers to the data transmission in a limited time-span and is also known as the maximum latency.
- **Periodicity** facilitates the recording of readings provided by sensors at regular time intervals.
- **Priority** designates precedence to the data flow in a WSN, depending on the criticality of data being transmitted.
- **Scalability** refers to the ability of a network to expand by adding more sensor nodes.
- **Fault Recovery** refers to the ability of a network to recover when one or multiple nodes have failed.
- **Energy Efficiency** is a vital parameter for WSN as the sensors are battery-operated and hence have limited energy resources. The optimum usage of these resources is termed energy efficiency.

- **Latency** refers to the delay in the network and needs to be minimized according to user requirements.
- **Throughput** measures data flow rate throughout a network in a particular time span and is measured in Kbps (Kilobits per second).
- **Packet Error Ratio** and **Loss Ratio** are very useful parameters for QoS, as these networks are quite error-prone. The Packet Error Ratio is the ratio of sent packets experiencing an error to the total packets communicated. The Loss Ratio describes the ratio of packets lost to the number of packets transmitted.
- **Responsiveness** deals with the network adaptability during topological changes.
- **Variation in Delay** refers to synchronization between the occurrence of an event and its reporting at the base station (or at the final processing node).

The dependability evaluation [11, 12] of a network is defined as the potential to provide functionalities, those that can be reasonably vested. And, calibrating trust in a network to deliver the services at a particular time is also known as the dependability of a network [13]. The presented work focuses on three facets of dependability evaluation (Kindly refer to Figure 1.3): The Characteristics; Negative Traits, to be restrained while designing dependability (Threats); Positive Traits, for dependability enhancement (Means).

The various characteristics considered are safety, integrity, availability, security, reliability, confidentiality and maintainability. The **dependability** of a network is inferred by its characteristics. The inference is drawn based upon the user requirements. The **characteristics** include:

- **Safety** establishes user and environment friendliness with the network.
- **Confidentiality** ensures authenticity and non-theft of data during transmittal.
- **Integrity** ensures the correctness of data. It promises that the data has not been modified in an improper manner.
- **Maintainability** and **Adaptability** go hand-in-hand. Adaptability ensures that the network is easily adaptable in a particular environment. Maintainability facilitates the capability to improve and adjust.

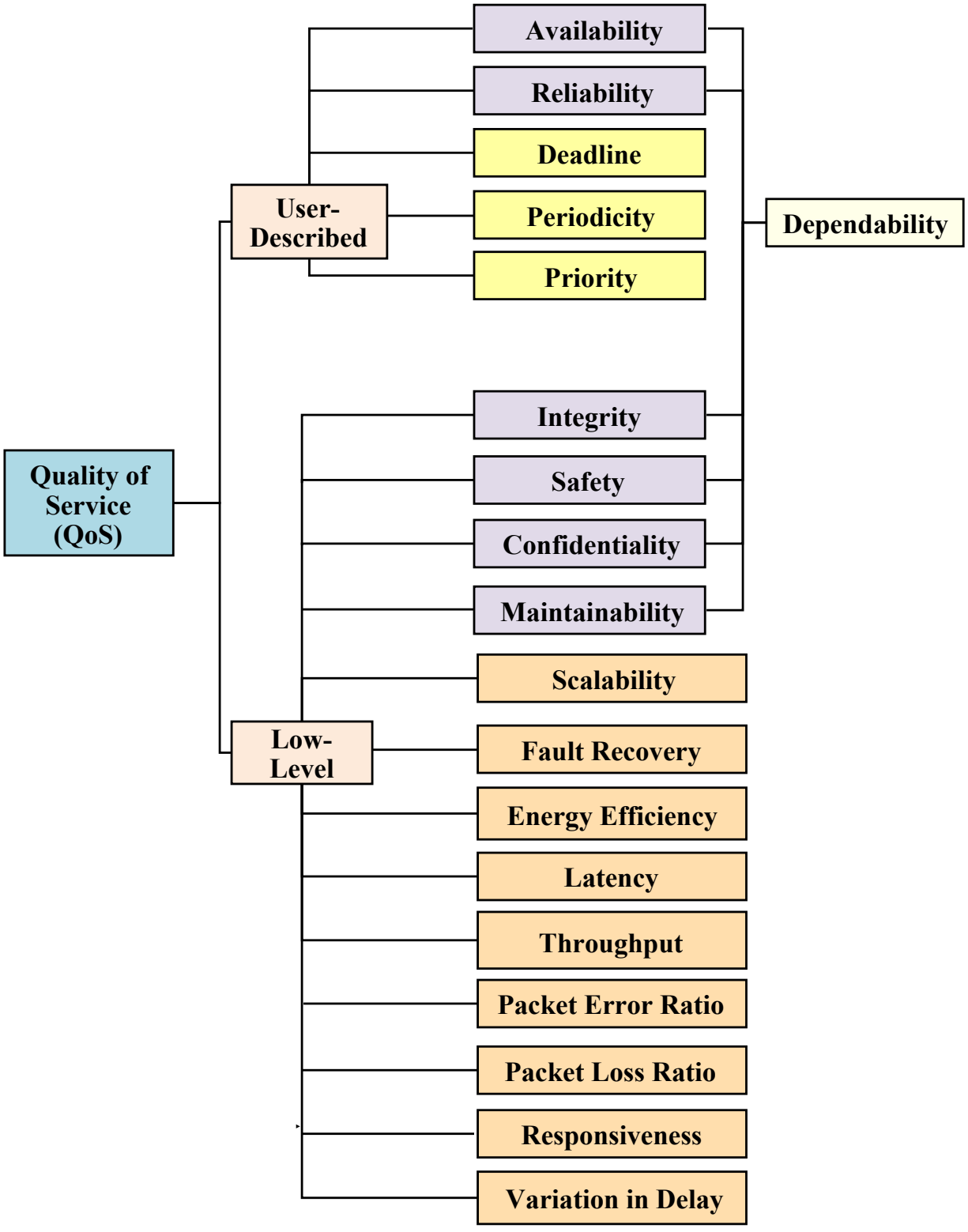


FIGURE 1.2: Quality of Service and Dependability

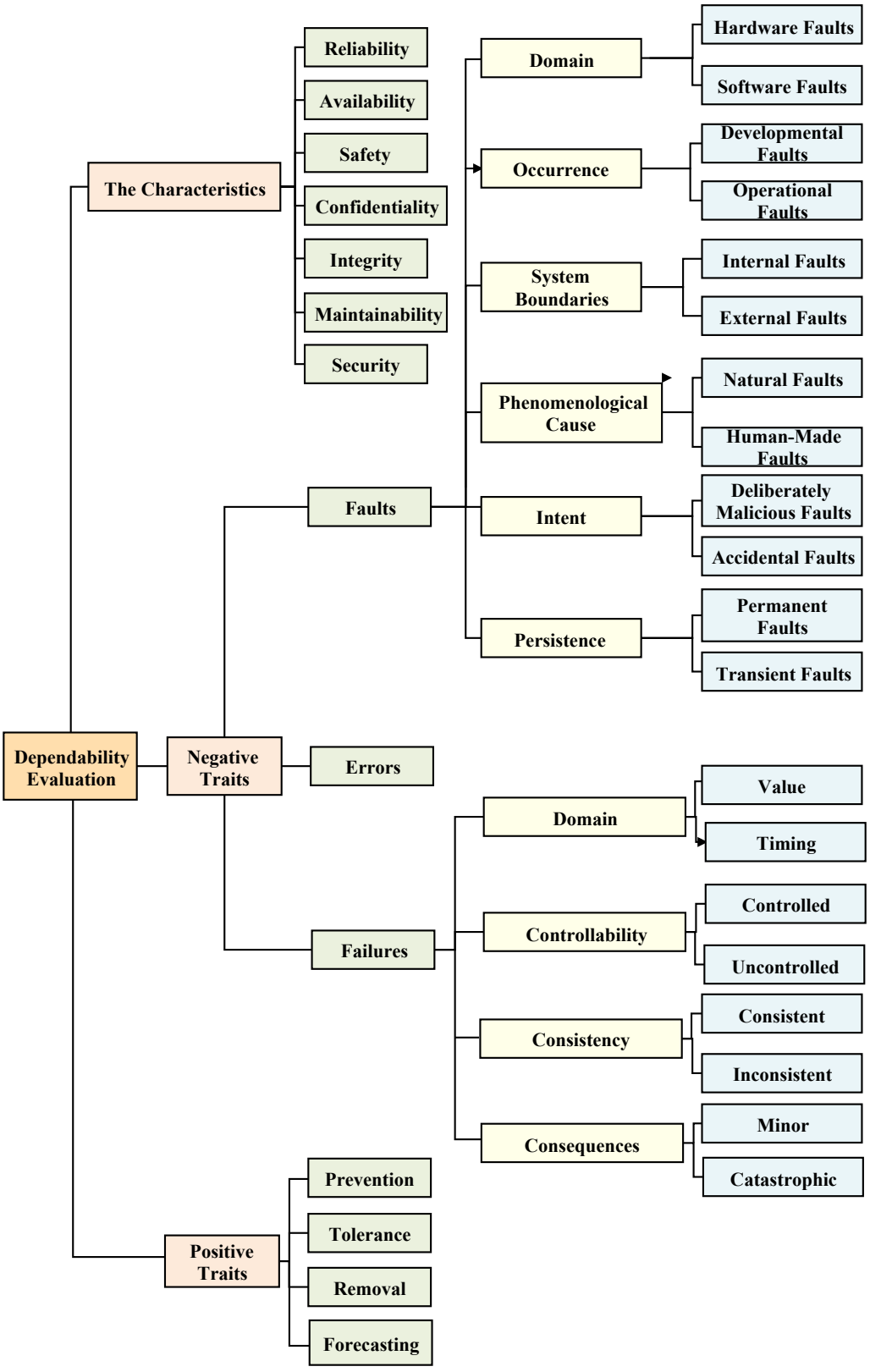


FIGURE 1.3: Dependability Evaluation

- **Availability** represents the readiness to use a specific network.

- **Reliability** represents the state of a network to provide accurate service on a regular basis. It can be ensured at various levels such as packet-level, event-level, end-to-end and hop-by-hop. The Reliability and Availability terms are the basic measures of network performance and can be expressed using two parameters: MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair). MTBF is the average time between two subsequent failures in the network. And, the average time required by the network to recover from this failure is known as MTTR. The reliability and availability [14] of the network is calculated using Equation 1.1 and Equation 1.2:

$$Reliability = e^{\frac{-Time}{MTBF}} \tag{1.1}$$

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{1.2}$$

  The greater the value of MTBF, the higher is the reliability of the communication network. More is the time required for a network to recover, lesser will be its availability.

- **Security** means safe access to data traversing in a network and independence from external intruder attacks.

The **Negative Traits (Threats)** incorporate error, failure, and faults [15]. These threats hamper the network by not allowing it to function correctly in prescribed time duration. These are described in Figure 1.3. Summarized definitions of threats include:

Fault reveals the flaw in a network, thereby causing an error. Faults cause an error in the network, thereby leading to network failure. The faults are being divided into six main categories, as explained:

- **Domain:** The dimension, in which the fault occurs, including the software or hardware faults are known as domain fault.

- **Occurrence:** The time duration in which the fault originates falls under this category. If the fault occurs during the designing phase, it is termed as a

developmental fault and if it occurs during the functioning of the network, it is known as an operational fault.

- **System Boundaries:** This tells the exact location of a fault, i.e. inside the system (called Internal Faults) or outside the system (called External Faults).
- **Phenomenological Cause:** If the fault occurs without human participation, it is called a natural fault; and if it occurs due to human actions, it is called human-made faults.
- **Intent:** As the name suggests, this fault is based on the intention of human-being. The faults introduced without knowledge of the user is called accidental faults. The faults due to incorrect decisions, taken intentionally are categorized as deliberately malicious faults.
- **Persistence:** This deals with how the fault basically persists in the network. If the fault occurs for a while, it is called transient fault and if it stays permanently in the network, it is a permanent fault.

An **Error** appears at run-time. In this case, the network enters an unsafe state resulting in an incorrect outcome:

Failure takes place when a network enters an illegitimate state from a legitimate state and hence the service becomes unusable. Failure occurs when a network functioning deviates from correct functioning to incorrect functioning. It is classified as:

- **Domain:** It includes value failure and timing failure. When the output value does not match with the specification value, it is called value failure. Whereas the time at which output is received does not comply with the network specifications, it is termed as timing failure.
- **Controllability:** If the cause of failure can be prevented, it is known as a controlled failure, and if not it is known as an uncontrolled failure.
- **Consistency:** When all network users perceive the failure, in the same manner, it is a consistent failure. And, when failure is perceived differently by various users, it is called inconsistent failure.

- **Consequences:** It determines the severity of the failure occurred. If the consequences of failure are worst, it is known as a catastrophic failure. And, if the failure has a minor effect on the functioning of the network, it is classified as a minor failure.

And, the Positive Traits [11] consists of fault prevention, removal, tolerance, and its forecasting. These user requirements vary from one application domain to another and from one user requirement to another, making it difficult to predict dependability. The dependability of a network can be enhanced with the help of **Positive Traits (Means)**, which includes:

- **Prevention** governs the transition from the designing stage to the maintenance stage and starts from preliminary to the terminal stages.
- **Tolerance** is provided by the network so that it can work uninterruptedly and reliably providing efficient services.
- **Removal** deals with the faults, thereby reducing them and further prevents the occurrence of severe faults.
- **Forecasting** eliminates the possibility of fault occurrence in the future. Machine Learning plays an important role in the real-time network forecasting.

In essence, to efficiently predict dependability one should have prior knowledge of what makes a system inoperable at a particular time.

1.3 Machine Learning in Wireless Sensor Networks

Machine Learning is an assortment of tools and pseudo code required for the creation of effective prediction models, as stated by the sensor network designers. Further, the experts in the field of Machine Learning identify it on based on data patterns and themes. The researchers who want to apply Machine Learning to the networking domain need an in-depth understanding of such data patterns involved. As WSN applications deal with dynamic nature, Machine Learning provides a more flexible solution to the researchers

[16]. The main crucial factors to be considered while designing a WSN are the limited storage and power capability of sensor nodes, their dynamic deployment, the management of the vast geographical region and the faulty communication links which cause failure. The various functional challenges of WSN, addressed so far by using Machine Learning techniques [17] include:

1. Routing in WSN
    - Data Routing using Self-Organized Map (SOM)
    - Routing Enhancement using Reinforcement Learning
2. Clustering and Data Agglomeration
    - Large Scale Network Clustering using Neural Network
    - Cluster Head selection using Decision Tree
    - Gaussian Process for Sensor Readings
    - Data Agglomeration using Self-Organizing Map (SOM)
    - Data Agglomeration using Principal Component Analysis (PCA)
    - Collaborative Data Processing using K-Means Algorithm
3. Event Detection and Query Processing
    - Event Recognition through Bayesian Algorithms
    - Forest Fire Detection through Neural Network
    - Query Processing through K-Nearest Neighbors
    - Distributed Event Detection for Disaster Management using Decision Tree
    - Query Optimization using Principal Component Analysis (PCA)
4. Localization and Object Tracking
    - Bayesian Node Localization
    - Robust Location-Aware Activity Recognition
    - Localization based on Neural Networks
    - Localization using Support Vector Machine
    - Decision-Tree based Localization

Also, various non-functional challenges of WSN, addressed by Machine Learning techniques [18] include:

1. Security and Anomaly Intrusion Detection
    - Outlier Detection using Bayesian Belief Network
    - Outlier Detection using K-Nearest Neighbors
    - Outlier Detection using Support Vector Machine
    - Analyzing Attacks with Self-Organizing Map (SOM)
2. QoS, Data Integrity and Fault Detection
    - QoS Estimation using Neural Network
    - Link Quality Estimation Framework
    - Accuracy and Reliability Prediction of Sensor Network
3. Miscellaneous Applications
    - Resource Management using Reinforcement Learning
    - Air Quality Monitoring using Neural Networks
    - Intelligent Lighting Control using Neural Networks

1.4 Research Gaps

The traditional networks are comparatively different from the WSN in various aspects. Therefore, different protocols and tools are required to address these aspects. The main issues in WSN are energy constraint, communicating data from source to destination, security, node deployment, reliability, data agglomeration, fault discovery and integrity of transmitted data. The dynamic behavior of a WSN can be effectively dealt using Machine Learning techniques. So far many issues discussed above have been resolved using Machine Learning based solutions. Still, some issues have not been explored in the light of Machine Learning such as the enhancement of QoS and dependability of the network, resource management, and fault-free data communication.

1.5 Objectives

To fulfill the above identified research gap in literature, the contribution of proposed research is expected to:

1. To study and analyze the existing techniques for dependability evaluation in Wireless Sensor Networks.
2. To devise and develop a model for dependability evaluation in Wireless Sensor Networks.
3. To compare and validate the proposed model with other existing models.

1.6 Thesis Contribution

In this thesis, an attempt has been made to solve the dependability issue of WSN using the Machine Learning approach. More specifically, the main issue addressed is the reliability and security of WSN based on the performance parameters mainly data flow. The main contribution of the thesis are done in several phases and they are as follows:

1. In WSN, reliability is defined as the capability of a network to perform its intended task under certain conditions for a stated timespan. There are many tools for modeling and analyzing the reliability of a network. As the intricacy of various wireless networks is increasing, there is a dire need for state-of-the-art methods in reliability analysis. The term reliability is used as an umbrella term to capture various attributes such as safety, availability, security, and ease of use. The existing methods have many shortcomings which include inadequacy of a novel framework and inefficacy to handle scalable networks. This research work presents a novel framework that predicts the overall reliability of the SSWSNs in terms of performance metrics such as, sent packets, received packets, packets forfeit, packet delivery ratio and throughput. This framework includes various phases starting with scenario generation, construction of a dataset, applying ensemble-based ML techniques to predict the parameters which cannot be calculated. The ensemble

model predicts with an optimum accuracy of 99.9% for data flow, 99.9% for the protocol used and 97.6% for the number of nodes. Finally, to check the robustness of the ensemble model 10-fold cross-validation is used.

2. The WSN is gaining paramount importance due to its application in real-time monitoring of geographical regions (hostile and remote). Its deployment paradigm encompasses a myriad of applications in the terrestrial, underground and underwater environment. At present, there is a paradigm shift taking place from mobile computing to data science. Bridging the two technologies results in the development of various new applications in which security plays a pivotal role. The most imperative aspect of a dependable network is its security. Hence, this research work considers the flooding attack in which, an attacker repeatedly sends packets at higher rates, thereby causing a packet drop and exhaustion of the link capacity resulting in communication failure. To detect this attack, an Intrusion Detection System (IDS) based on the randomized and the normalized deployment of nodes is proposed. Further, Machine Learning techniques are implemented to enhance the dependability of WSN under a flooding attack. The data flow is a significant parameter for governing the flooding effect on the network. It is found that Machine Learning models play a significant role in the prediction of the data flow, which is related to the packet delivery ratio of the network. The experiments on the simulated dataset underline the role of Machine Learning models for data flow prediction on the normalized dataset.

3. Leveraging the benefit of neural networks, this research work focuses on traffic flow prediction in the WSN. Also, Machine Learning models based on the concept of Neural Networks are developed to accurately forecast traffic flow. Various Artificial Intelligence based techniques, namely Neural Network, Bayesian Regularized Neural Networks, Neural Networks using Model Averaging, Multi-Layer Perceptron, Multi-Layer Perceptron with Multiple Layers, Quantile Regression Neural Network and Stacked Autoencoder Deep Neural Network are analyzed. The models are thus designed and tested. Based on the experiments performed on the first-hand data obtained using simulations, it is observed that Quantile Regression Neural Network and Multi-Layer Perceptron with multiple layers models outperform in terms of Accuracy and Root Mean Square Error. The results show that these methods are quite effective and improve prediction efficiency.

The findings of the entire research work are concluded along with the potential scope for future directions. Artificial Intelligence and Machine Learning have reached a critical tipping point and will increasingly augment and virtually extend to every technology-enabled service(s), thing(s), or application(s). This research work paves a way for improving the dependability of a WSN. Needless to mention, the application of this research can be explored for use in - Healthcare, Financial Trading, Social Network Analysis, and many more fields or applications.

1.7 Thesis Organization

The rest of the thesis is structured as follows. Chapter 2 covers the background information. The proposed framework for reliability prediction is presented in detail in Chapter 3. Chapter 4 discusses the security aspect of dependability. Traffic flow prediction using intelligent neural networks based techniques are presented in Chapter 5. Finally, Chapter 6 concludes the thesis and points out the scope of further research. The graphical layout of the thesis is shown below in Figure 1.4:

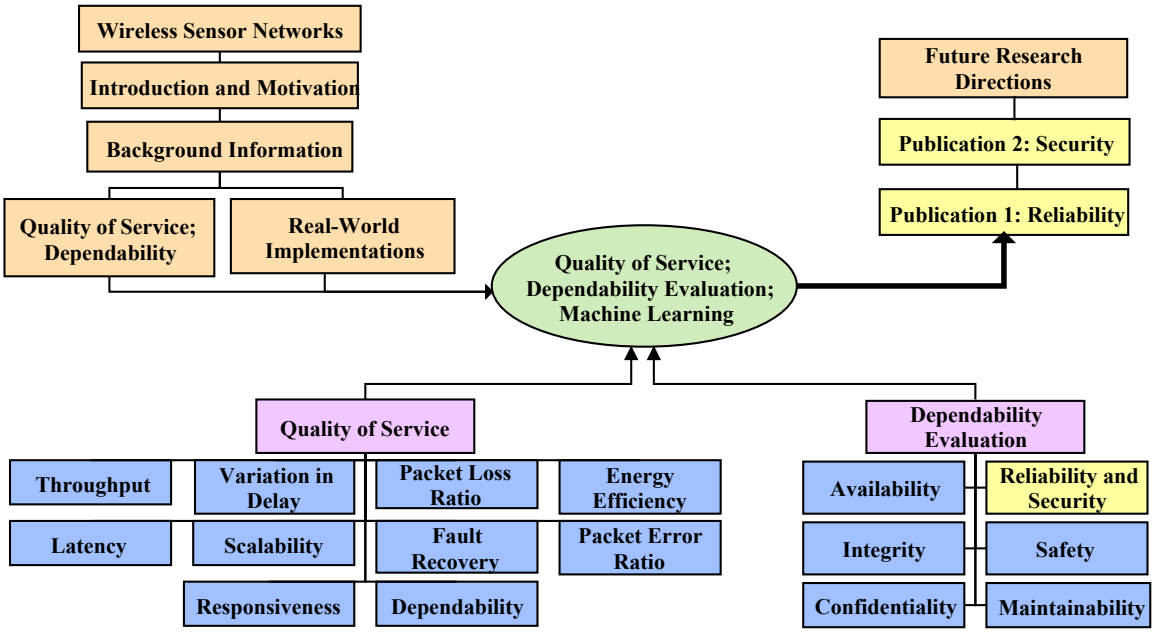


FIGURE 1.4: Thesis Organization

Chapter 2

Background Information

"Stand on the shoulders of giants."

-Bernard of Chartres and Isaac Newton.

The dependability of a network evaluates its efficient functioning. It is a critical parameter that comes under an umbrella term of QoS. This chapter is a qualitative review based on the author's perception of the work related to QoS and dependability evaluation, as reviewed in the literature. The background information selection criteria employed by the authors for deciding on research paper inclusion is represented diagrammatically in Figure 2.1. This work includes the keywords dependability, network analysis, the dependability metrics, WSN, the QoS, safety, security, availability, reliability, Machine Learning.

The relevant papers are collected from various online repositories such as IEEE, ACM, Wiley and include conferences, journals and book chapters. The full text of articles is reviewed to consider the attributes covered and the research gaps in the publications. These further paved a way for futuristic research. Also, comparative analysis of publications is considered from the year 1992 – 2019 in reverse chronological order. The dependability evaluation along with their real-world implementation areas, the research gaps, the evaluation metrics and the tools used are studied and briefly described in Table 2.1. Machine Learning can be applied to various disciplines [19, 20, 21, 22]. The models considered in the course of this work are also presented in reverse chronological order in Table 2.2:

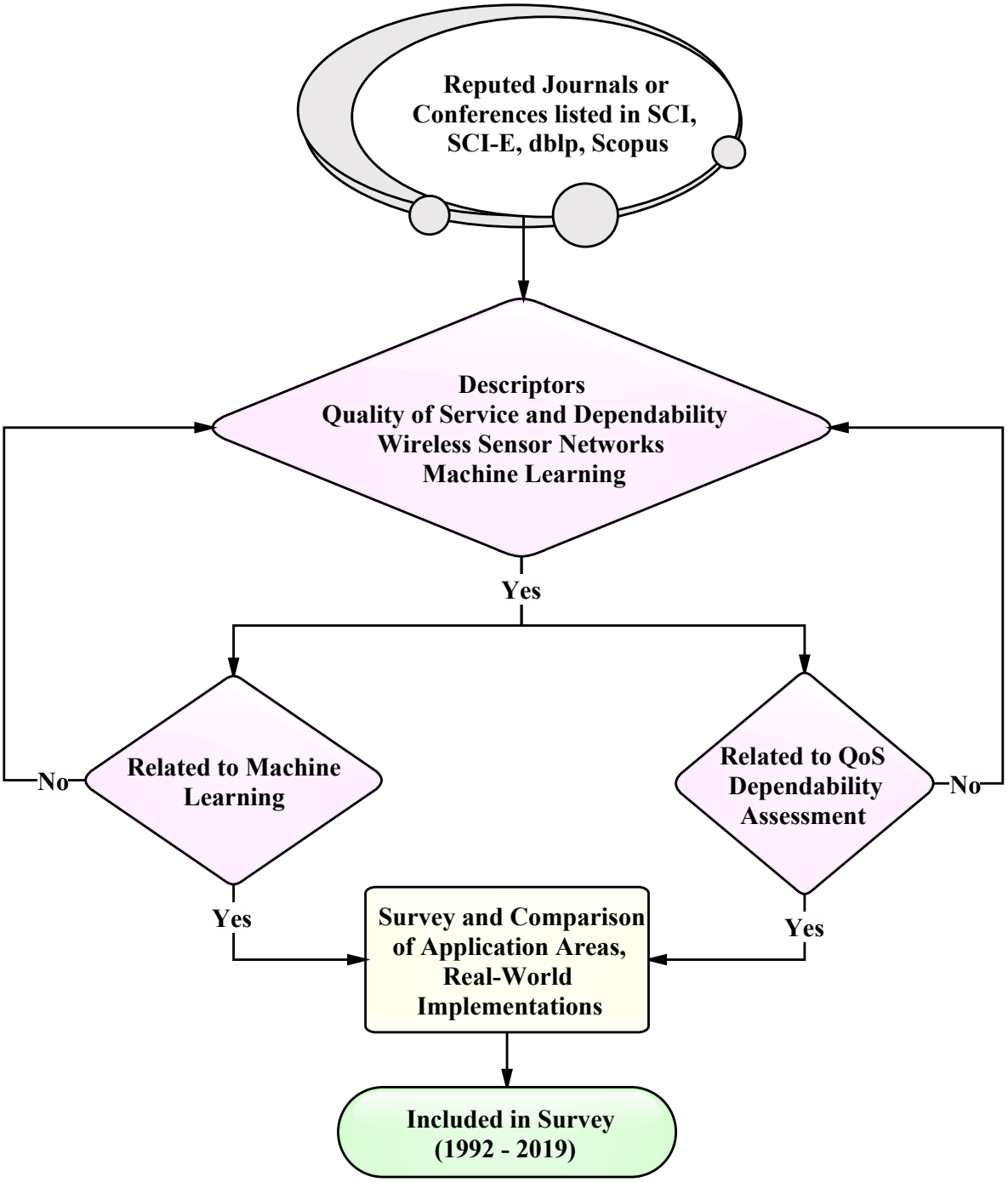


FIGURE 2.1: Background Information Selection Process

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2019	Gilbert E.P.K. [23]	• A trust model based on the time-series is developed. • This model monitors the power generated and consumed in the entire scenario. • Also, this model is validated in the presence of faults which leads to data loss.	• Limited parameters (the data-loss fault and the offset fault)considered for the development of this trust-based model. • Various performance parameters such as accuracy, execution time taken not considered in the course of this work.	• Root Mean Absolute Error • Mean Absolute Error	• Real Testbed named the SensorScope network
2019	Khan T. et al. [24]	• This work proposed a novel approach for trust calculation in large-scale WSNs. • This approach operates inter-cluster and intra-cluster hence, make use of centralized as well as decentralized mechanisms.	• This mechanism applies only to homogeneous WSN and can also be scaled up to address the Denial-of-Service attack on various layers of WSN.	• Communication Overhead • Trust Value	• MATLAB
2019	Akerele M. et al. [25]	• An adaptive scheduling algorithm is designed for Fiber-Wireless Sensor Networks to decrease delays in high priority traffic on the communication channel.	• The concept of optimized reliability need to be explored for higher priority traffic.	• Delay • Reliability	• Omnet++

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2019	Bahi J. et al.[10]	• The main emphasis is on the data loss. • This approach is presented considering the industrial domain.	• The experiment is performed on three topologies only and can be extended to many other deployment paradigms.	• Delay • Number of Errors per Second	• WSN Simulator • R
2018	Zhang T. et al.[26]	• Trust evaluation method for clustered WSN is proposed based on the cloud concept. • The proposed model considers multiple factors such as message, energy.	• This approach is limited to some attacks only such as black hole attack, faked ID attack, sybil attack, selfish behavior attack. • The routing paradigm needs further exploration in order to detect more attacks.	• Trust Grade • Time • Misbehavior Rate • Detection Rate • Computational Complexity • Storage Overhead	• OMNET++
2018	Almeida J. et al. [27]	• This paper discusses dependability in safety-critical wireless networks. • The fault-tolerance mechanism is considered, known as the Medium Guardian Concept.	• The work emphasizes only on the safety perspective, the security attribute is not considered. • Also, the node mobility needs to be considered in the research work.	• Node Faults • Channel Transient Faults • Channel Permanent Faults • Error Detection Latency	• Laboratory Setup for Vehicular Communication considering real-world transmissions

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2018	Sandhu J.K. et al. [28]	• The reliability of WSN is considered based on the basic performance parameters such as packet delivery ratio, throughput, and packets forfeit. • Also, the applicability of Machine Learning techniques in prediction of various parameters of WSN is considered.	• The work considers scalability at small-scale. But the same approach can be extended to large scale WSN deployments also.	• Data Flow • Protocol Name • Number of Nodes • Packet Delivery Ratio • Packets Forfeit	• NS-2.35 • R (Version 3.2.2)
2018	Zhang Z. et al. [29]	• Fault-diagnosis is considered as an aid to enhance the dependability of network. • A survey is provided listing the merits, demerits, goals and limitations of various research papers.	• The limitations of Software Defined Networks (SDN) in terms of data communication and storage need further exploration. • Intelligent routing of data using prediction must be considered and provides a great field for future research.	• Traffic Conditions • Number of Faulty Nodes • Network Lifetime • Safety	• Probabilistic Method • Prediction Techniques • Passive Diagnosis Method • Fuzzy Classification

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2017	Noel A.B. et al. [30]	• Critical Infrastructure Monitoring. • Discusses real-world WSN deployments. • Discusses network design challenges.	• Not all QoS parameters considered. • Health monitoring perspective considered.	• Load • Localization • Network Scalability • Synchronization • Sensor Placement Optimization • Energy Efficiency	• Real Testbeds as well as Laboratory Testbeds • COMSOL • WiSeREmulator • TinyOS
2017	Lee H.C. et al. [31]	• Slope Movement Monitoring • Real-World WSN deployment considered	• The only parameter emphasized is low standby power consumption.	• Slope Movement Monitoring • Power Consumption Analysis • Detect Time (Seconds) • Sleep Time (Minutes)	• SMARTCONE
2017	Yuan D. et al. [32]	• A detailed survey of the WSN metrics has been presented. • The interaction between WSN and Internet of Things (IoT) also discussed.	• The run-time measurement of metrics for WSN optimization is still under exploration.	• Node Centric Metrics • Hop Centric Metrics • End to End Metrics • Network Centric Metrics	• SCALE • SWAT • RadiaLE • TRIDENT
2017	Ueyama J. et al. [33]	• Reliability • Adaptability • River Monitoring System (Case study: Brazil)	• The areas concerning failures in dynamic and critical environment needs further exploration. • The level of safety in an environment needs discussion.	• Monthly Availability Rates (Percentage) • Availability • Safety	• Raspberry PI2 • LooCI and OpenCom Middleware • Real Testbed

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2017	Ojha T. et al. [34]	• Sensor-cloud framework in agricultural domain discussed.	• The framework is not generic and fits well to the agricultural domain only. • The mobility-aware dynamic service management is missing.	• Energy Consumption • Duty • Network Lifetime • Utility • Cost	• NS-3 (Version 3.14)
2016	A.E.Zonouz et al. [35]	• Contributes by combining BPSNs (Battery-Powered Sensor Nodes) and EHSNs (Energy Harvesting Sensor Nodes). A cost-function based routing technique is designed that positively affects the cost, energy, reliability and QoS of the network.	• The reliability enhancement is the main goal of this paper. • Further, optimization techniques need more exploration for improving the overall reliability and QoS.	• Path reliability • Energy Consumption • Residual Energy	• Maple software
2015	W. Elghazel et al. [36]	• Discusses the dependability issue in Wireless Sensor Networks and then elaborates on state of art of prognostic and health management field.	• This work studied Prognostic models for dependability. • No consideration of prognostic methods when data-flow is non-continuous. • No provisioning of good predictions when data is incomplete.	• Remaining Useful Life (RUL) prediction	• Simulation based

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2015	Muhammad Adeel Mahmood et al. [37]	• Reviews reliability schemes based on redundancy and retransmission methods.	• Reliability schemes discussed are based upon Retransmission, Redundancy, Packet and Event. • No mechanism provided for Redundancy-based Event Reliability. • Lack of research on the use of a hybrid mechanism involving both retransmission and redundancy to provide reliability in WSN.	• Reliability • Energy consumption • Complexity • Latency • Scalability • Routing • Deployment strategy	• GlomoSim • NS-2 • TOSSIM
2015	Arslan Munir et al. [38]	• Models and analyzes fault detection and fault tolerance in WSNs. Markov models are developed for reliability and MTTF (Mean Time to Failure) computation.	• Most of the event detection approaches fail in distinguishing between events and faults if faults are located at the event boundary. • No additional security feature is provided for Heterogeneous Hierarchical Multi-core Embedded Wireless Sensor Network.	• Errors Detected • Mean Time between Errors • False Positives • Mean Time to Failure • Sensor Failure Probability	• NS-2 • MATLAB • SHARPE
2014	Guangjie Han et al. [39]	• Lists many trust based best practices for a robust trust model.	• Trust Models in Ordinary WSN and Cluster-Based WSN are discussed. • Multi-Criteria trust model incorporating multiple factors such as data trust, node trust, link trust, behavior trust not discussed.	• Prediction • Data Trust • Node Trust • Link Trust • Behavior Trust	• Simulation based

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2014	Antonio Damaso et al. [40]	• Routing protocol-based model has been proposed for calculating reliability. • Battery level is the key factor. • Three unique sections have been incorporated in this paper: (1) Reliability model for evaluating power and routing algorithms; (2) automated reliability model generation; (3) tools to support these models.	• Two main steps: (1) consideration to reliability of additional routing protocols; (2) to increase reliability, enhance the current tools available. • All the routing algorithms discussed in the paper are for centralized network and homogeneous nodes. • Routing in decentralized network topology not discussed. • The heterogeneous nodes in WSN not explored.	• Reliability • Power Consumption • Energy Consumption	• Tooling: Editor, Translator and Evaluator • CPN Tools • Mercury
2014	Joseph E. Mbowe, George S. Oreku [41]	• Three important factors namely, Reliability, Availability and Serviceability (RAS) has been used to increase the reliability, thereby, improving the QoS of the overall WSNs.	• Performance is decreased considerably because of the traditional metrics such as delay, throughput. QoS of overall system degrades as a result. • QoS metrics incorporating mechanism dealing with delay, jitter has not been discussed,	• Mean Time between Failure • Mean Time to Repair • Delay • Throughput • Jitter	• Numerical Evaluation

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2013	B.Silva et al. [42]	• Presented an integrated environment, namely, ASTRO, this contemplates: (i) RBDs, SPNs and CT Markov Chain for dependability evaluation; (ii) a method based on Life-Cycle Assessment (LCA) for quantification of sustainability impact.	• ASTRO has a generic environment which can be used to evaluate general systems. • Certain analytical techniques for SPN models are missing. • Functionalities related to IT infrastructures are absent from the system. • Presented tool is not suitable for heterogeneous and dynamic networks.	• Lifetime Exergy • Total Cost of Ownership • Availability • Mean Time to Failure • Mean Time to Repair	• ASTRO • Mercury
2013	S.Bernardi et al. [43]	• Overview of the traditional dependability techniques has been presented. Main focused techniques is the Fault Tree Analysis and Petri Net Analysis.	• The CPN is another tool which can be implemented for dependability modeling, which is missing from the paper.	• Fault Rate • Repair Rate • Mean Time to Failure • Mean Time to Repair	• UML tools • StarsStudio-Astra • DEEM (Dependability Modeling and Evaluation of Multiple Phased Systems)

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2013	Waseem Ahmed, Yong Wei Wu [44]	• Discusses models for reliability calculations and also deals with unpredicted issues using fault tolerance.	• Experiments are carried out based upon: (1) old existing data of some similar application; (2) simulation outcomes; (3) testing of some similar real environment. • Reliability analysis must be carried out continuously because of unpredictable nature of the internet. • No discussion about CPU Load, User Load and Network Traffic which affects reliability.	• Trust • Communication Time • Processing Time	• PRISM
2012	Sazia Parvin et al. [45]	• Survivability evaluation model has been presented in two circumstances: (1) under an attack, and (2) under key compromise.	• No computation carried out with regard to security analysis of the proposed scheme. The detailed analytical method not available. • No mechanism for single node software rejuvenation and reconfiguration used. The self-healing concept not explained. • No consideration given to the extended version of this work. This work deals with the overall network level cooperation in case of time-critical operation using rejuvenation and reconfiguration of the software.	• Rejuvenation Rates • Availability • Survivability • Recovery Rates • Revocation Rates	• SHARPE • IBM Software Rejuvenation Tool

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2012	Yongxian Song et al. [46]	• Bionic reconfigurable WSN nodes have been implemented to facilitate self-healing concept. • For reliability measurement, markov model and probability techniques have been used.	• This paper deals only with hardware failure by implementing bionic reconfigurable Wireless Sensor Network nodes. • No provisioning for Software and Network failure. • No consideration for dynamic constraints and heterogeneous nodes.	• Sink Node Reliability • Number of retransmissions • Energy consumption • Communication time	• Anadigm Designer 2
2012	Ivanovitch Silva et al. [14]	• Industrial environment is considered for dependability measurements. • Also, fault-tree based technique has been used to model WSN.	• Only permanent faults and network topology faults have been considered. • No premeditation about Transient failures and common-cause failures.	• Failure Rate • Repair Rate • Mean Time to Failure • Birnbaum's Measure • Criticality Importance • Unavailability	• SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator)
2011	Jaime Chen et al. [47]	• QoS and dependability evaluation parameters are discussed.	• For Critical Infrastructure Protection (CIP), certain level of QoS needs to be ensured depending upon applications requirement. • No consideration to a comprehensive way of furnishing QoS universally, for satisfying needs of multiple users.	• Energy Efficiency • Scalability	• Survey concludes with the finding that real-platform is preferred as compared to simulations.

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2011	Yves Langeron et al. [48]	• Aimed at conserving energy thereby extending the dependability.	• SPN used to model independency of components, heuristic algorithm used to schedule component activity. • No mechanism provided for node mobility.	• Failure Rate • Coverage • QoS • Average Availability of the intended Safety function over a given mission of time	• Petri-Nets • Moca.RP
2010	Chang et al. [49]	• Algorithmically reliability has been computed in distributed systems using the recursive merge and the binary decision diagram.	• Reliability is enhanced using proposed algorithm in a distributed environment. • No consideration given to analysis of failure frequency. • No analysis provided for sensitivity parameter.	• Hit Ratio • Execution Time (Seconds) • Number of Sub-graphs • Distributed Program Reliability • Terminal-Pair Network Reliability	• C/C++ language implemented on a primary-type SUN UltraSPARC workstation
2010	James P.G. Sterbenz [50] et al.	• Architectural framework for resilience and survivability has been suggested.	• Survivability, dependability, security, performability are related terms. • $D^2R^2 + DR$ principle for resilient networks designed. • No attention provided towards understanding and defining resilience metrics, remediation mechanisms.	• Mean Time to Failure • Mean Time to Repair • Mean Time Between Failure • Operational State (Normal Operation, Partially Degraded, Severely Degraded) • Service Parameters (Acceptable, Impaired, Unacceptable)	• Survey on projects: ANSA, T1, CMU-CERT, SUMOWIN, FIND, FIRE, GENI, PoMo, ResumeNet, and GpENI

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2010	Zhengyi Le et al. [51]	• Exponential and Weibull models used for reliability analysis. • Cost Vs Reliability computations have been carried out with the help of simulation.	• An additional design of component-based network reliability is formulated by modeling of user needs before investigating the design reliability. • No analysis provided for availability, serviceability in a dependable environment.	• Hardware Cost • Lifetime of Sensor • Reliability • Time	• Sunspot Wireless Sensor Nodes • MATLAB
2010	Dario Bruneo et al. [52]	• Dependability parameters, namely, reliability and producibility have been studied characterized by active-sleep cycles.	• No consideration for complex network topology. • No evaluation about the communication link failure. • No discussion about the dependability in case of wireless links which are not reliable.	• Reliability • Producibility • Time (Seconds) • Mean Time to Failure • Overall Number of Nodes	• WebSPN • Monte Carlo Simulations
2009	Cardellini et al. [53]	• Proposed an approach based on modeling of a Service-Oriented Architecture(SOA), which are basically self-adaptable and fulfill the dependability features.	• Only one adaptation mechanism based on architecture selection paradigm is discussed. • No consideration for other adaptation mechanisms such as the spatial redundancy concept to increase flexibility. • Consideration given to only reliability and availability attributes which could be extended to a broad category of QoS requirements.	• QoS • Service Time • Reputation • Cost • Dependability	• Developed a prototype tool • ActiveBPEL • ApacheODE

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2006	Lihua Xu et al. [54]	• The dependability requirements are transformed into their corresponding software architecture constructs.	• The proposed pattern is too generic to facilitate the enterprise and product-line architectures. • No discussion to more real-world applications for modeling the NFRs. • No discussion about transforming the architectures to programs and providing automated tool support.	• Non-Functional Requirements (NFRs)	• Aspect Oriented Programming • XML • Java • COSMOS • UML Argus-I toolset
2006	Lance Doherty, Dana A. Teasdale [55]	• TDMA-network with centralized monitoring policy is used to achieve maximal amount of received packets, while considering low-power feature emphasizing enhanced reliability.	• A node when reset, is disconnected from the network leading to packet losses. This practical scenario is not considered in this work. • Maintenance of nodes also leads to packet loss. • Not applicable to decentralized scenario. • Only mesh topology considered whereas, other topology such as hypercube can be a better option. • For topology optimization, methods which give better results such as gradient method or dimension exchange method are not considered.	• Number of Packets Lost • Loss Ratio • Time (Days) • CRC Errors • Lifetime of Network	• Real Testbed with 50 nodes using AA batteries

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2006	Amirhosein Taherkordi et al. [56]	• An event-based middleware service has been designed satisfying two important dependability factors, namely, reliability and availability.	• No discussion about middleware upgradation. This can be done using some traditional services such as caching. • Quality factors such as safety and security not investigated. • No discussion about calculating availability and reliability on nodes present in a cluster.	• System Availability • Number of interested events delivered to sink node compared to events detected by sensors • Loss Rate	• JIST Simulator
2006	Michael G. Hinchey et al. [57]	• Designed a model which mechanically tantamount requirements into comparable formal model. This model is natural language based and is used for further transformations such as code generation.	• Communicating Sequential Processes (CSP) not optimized for efficiency. • Model not applied to multiple significant examples hence making it fully non-functional robust model to be implemented on large scale sensor networks.	• Ultra-High Dependability Systems • Completeness and Consistency of Requirement • Bug-Free • Cost • Delay	• CSP • EzyCSP • LOGOS (Lights-Out Ground Operating System) • TinyOS • Java, NesC, C++ • ANTS
2004	Claudia Betous Almeida, Karama Kanoun [58]	• Presents a modeling approach which provides generic process for analyzing the system, based upon GSPNs (Generalized Stochastic Petri Nets) and the refinement process.	• The approach presented is confined to a generic structure being used in instrumentation and control system embedded in power plants. • No generic approach is presented • No discussion about functional level analysis by which the system overhead and complexity can be managed.	• Liveness • Boundness • Safeness • Faults • Error Detection • Efficiency • Latency • Annual Unavailability	• SURF-2 • GSPN

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
2002	Mohamed Kaaniche et al. [59]	• Dependable systems are produced with the discussed developmental model.	• Dependability-explicit development model includes: (1) a process for creating a system, (2) dependability prediction process, and (3) other processes. • Emphasis is on activities to be carried out and the main objectives to be achieved. • No emphasis upon the methodology to be inculcated.	• Fault Prevention • Fault Tolerance • Fault Removal • Fault Forecasting	• Real-World Application discussed: GNSS2/ Galileo System
2000	Walter J. Gutjahr [60]	• Optimal state transition probabilities computed using markov model for software tests. • Also, optimization has been carried out	• Dependability attributes evaluated are: Risk, Safety, and Reliability. • The method described in this paper is conceptually complex • It cannot provide solution to very large Markov usage models. • No computation detail about performance measures.	• Risk • Safety • Reliability • Cost function • Uncertainty in behavior	• Mathematical Lemma and Proof • Rare Event Simulation Technique

to be contd. on next page

TABLE 2.1: QoS, Dependability Evaluation and their Real-World Implementations (contd.)

Year	Authors	Work Description	Data-traces and Gaps	Evaluation Metrics	Tools Used
1998	A.Platis et al. [61]	• Estimate reliability, availability, maintainability, and different time variables.	• The model is suitable for only independent modules of the system. • No provision for dynamic nodes.	• Overhead Line Failure Rate • Load • Maintenance Rate • Instantaneous Availability • Reliability • Instantaneous Expected Load Curtailed • Interruption Frequency • Energy not Supplied on a Time Interval	• Monte Carlo Simulation • Non-Homogeneous Markov Chains
1995	M.Malhotra and K.S.Trivedi [62]	• Presented Generalized SPN (GSPN) and Stochastic Reward Nets (SRN) for dependability modeling and analysis of distributed computing systems.	• The proposed model works only for Uniform and Non-Uniform Memory Access multiprocessor systems where, the components are homogeneous. • Model is not applicable to heterogeneous systems and for message passing architecture.	• Reward Rates • Failure Rate of Component • Storage and Time • System Reliability • Complexity	• GSPN, SRN modeling
1992	Lopez-Benitez [63]	• Reliability and availability has been computed with the help of SPNs. MFST has also been used.	• Reliability models proposed in this paper. • No discussion about performance models. • No discussion about transmission errors and software-related errors.	• Reliability • Global Repairs	• Stochastic Petri Nets Package (SPNP) • Markov Modeling

TABLE 2.2: Application of Machine Learning in different domains

Year	Model	Authors	Application Area	Contribution
2019	Q-Learning, Neural Network	Alarifi A. et al. [64]	Wireless Sensor Networks	The network performance is considerably enhanced for inter and intra cluster communication in a Wireless Sensor Networks. Also, Neural Network approach is applied to enhance the network lifetime.
2018	Extreme Learning Machine	Phoemphon S. [65]	Wireless Sensor Networks	The Wireless Sensor Network localization problem is discussed based on soft-computing paradigm. This work uses the Particle Swarm Optimization (PSO), Fuzzy Logic and the Extreme Learning Machine techniques to resolve the localization problem of these networks.
2017	Cubist	Lior Turgeman et al. [66]	Health-Care	The main advantage of the Cubist model is accountability of the predictions it provides. This provides in-depth knowledge of the logic rules required for construction. In this work, LOS (Length of Stay) has been computed using static input information.
2017	Random Forest	Julian Hagenauer, Marco Helbich [67]	Travel Mode Choice	The variable importance is an important factor while using various Machine Learning models. Random Forest is an ensemble classifier and generally performs better if there is convincing diverseness among the models.
2017	Random Forest	Luis M. Candanedo et al. [68]	Energy Use	The relationship between various variables has been considered. The Random Forest model is constructed using multiple regression trees. These trees are constructed using random sample values. This inturn improves the prediction.
2017	Support Vector Machine	Ke Hu et al. [69]	Air Pollution Estimation	SVM is a supervised model used for regression problems. It uses non-linear mapping. The training data from the dataset is mapped to higher dimensional features. A hyperplane can be defined for the same. A web-based application is also developed for this problem.
2017	Support Vector Machine	Michael E. Cholette et al. [70]	Computer-Aided Engineering	This work is useful when excessive numerical simulations are involved. Simulation results are fetched and an SVM model is thereafter applied.

to be contd. on next page

TABLE 2.2: Application of Machine Learning in different domains (contd.)

Year	Model	Authors	Application Area	Contribution
2017	Support Vector Machine	Kai Cheng et al. [71]	Global Senstivity Analysis (GSA)	A SVM-based mixed kernel function is proposed. This yields low computational cost. This approach is validated using analytical functions.
2017	Neural Network	Bin Weng et al. [72]	Stock Market	The online data from stock market, traditional time-series data and the online resources are utilized to prepare a dataset. The inference is drawn using Machine Learning models. This serve the purpose of decision-making using intelligent techniques.
2017	Neural Network	Santosh Singh Rathore, Sandeep Kumar [73]	Software Faults	A detailed insight to ensembling has been presented. And, an ensemble has been prepared to deal with various software faults. The ensemble being the most effective way which do not allow the outcome to reach worst-case.
2017	Bagged CART	Mohammad Ehsanul Karim et al. [74]	Statistical Learning Approaches	The Bagged CART model yields the outcome with lower biasness. Thus, leading to less variability and more accuracy than the logistic regression. Thus, Bagged CART model significantly improves the CART model.
2017	Tree Model from Genetic Algorithm	Lingjian Yang et al. [75]	Mathematical Programming	These models supports higher non-linearity, have excellent interpretation and also shows an improvement on Mean Absolute Errors.
2017	Conditional Inference Tree	Abbas Ali Rezaee, Seyyed Ehsan Golparvar [76]	Competing Motivators	In this work, a Random Forest of Conditional Inference Tree is constructed. This model outperforms logistic regression models as validated.
2017	Conditional Inference Tree	Jidong J. Yang, Bashan Zuo [77]	Performance of Smart Sensors	The study proves that error increase during bad weather. For the validation of the work, models used are Conditional Inference Tree, Regression Models.
2017	Bagged CART (Classification and Regression Tree)	Brandon Heung et al. [78]	Soil Mapping	This paper deals with an ensemble of CART model along with the Bagging model. It reduces the variance and is computationally effective model.

to be contd. on next page

TABLE 2.2: Application of Machine Learning in different domains (contd.)

Year	Model	Authors	Application Area	Contribution
2017	Bayesian Regularized Neural Network	Isabel Pocas et al. [79]	Grapevine Water Status	The Bayesian Regularized Neural Network (BRNN) model trains the neural network which are similar to data and robust as well. It avoids overfitting and overtraining issues.
2013	Bayesian Regularized Neural Network	Jonathan L. Ticknor [80]	Financial Sector	The biggest advantage of this work is that it enables expansion of network without creation of the overfitting issue. It is a generic technique which is applicable to stock exchanges.
2013	Weka Lazy Modifier	Sabina Smusz et al. [81]	Bioactive Compounds	The Weka package consists of many classifiers. The results has been evaluated using the heat maps. High-level results can be obtained by using meta-classifiers. Eleven Weka models has been used in the course of this work.
2012	Bagged MARS (Multivariate Adaptive Regression Splines)	Hui-Yi Lin et al. [82]	Single Nucleotide Polymorphisms (SNP)	In this work, Random Forest and Bagged MARS models have been combined to study the predictive sub-module of SNP and to analyse the interactions between them. These models proved quite useful in exploring the SNP interactions.
2009	Support Vector Machine	Chih-Chia Yao [83]	Image Restoration	Linear, Non-Linear, Reduced, Extractive SVM has been discussed. The massive storage for kernel matrix is dealt with. Statistics are used to extract the support vectors. The computational time and performance is enhanced.
2002	Bagged MARS	Simone Borra, Agostino Di Ciaccio [84]	Nonparametric Regression	Bagging and Boosting improves the accuracy as proved in the course of this work. MARS reduces the error coefficient.

The identified challenges governing dependability evaluation in WSN are discussed in this section. These challenges have paved a way for future researchers working in this particular domain. Event Reliability based upon redundant packet transmittal is an unexplored area of research. The reliability characteristic is considerably enhanced using this redundant transmission. The main constraint in WSN is the energy consumption which must further be optimized to invigorate the reliability of data transmittal. The hybrid mechanisms based on redundant transmittal and re-transmittal schemes need to be further explored to conserve energy. The problem of faults and events which are stationed at the event boundary needs to be addressed. Most of the techniques discussed in the literature are inefficient in this case. As surveyed, the reliability characteristics can be enhanced using the multi-criteria trust models which need further evaluation. These trust models are based on conglomerate factors such as behavior, link, node and data.

Also, the dependability evaluation carried out so far in literature makes use of static data. Dynamic behavior needs to be introduced to deal with the challenging current real-world application areas. This inculcates the need for additional novel techniques for adding dynamic behavior to the network. The scheduling of component-based activity in WSN is still an NP-complete problem. Such problems can be dealt with using appropriate heuristic approaches. Further, this research area can be explored from the prospect of Machine Learning.

The WSNs are popularly placed in a hostile environment such as defence. Event identification becomes very challenging in such application domains. Also, accuracy of the event identified must be higher. So far, this has been carried out by using sleep-and-wake up cycles on sensors so that a single message is transmitted only once (avoiding redundancy).

A generic framework must be devised which incorporates the dependability characteristics, thereby enhancing the overall QoS. This framework must satisfy the requisites of all users. The existing architectures must be mapped to codes and an automatic tool must be devised supporting multiple applications. The dependability evaluation schemes must deal with heterogeneity and dynamicity of WSNs. QoS is considered as Non-Functional Requirements (NFRs) of a system. More real-world applications need to be dealt with while modeling QoS and dependability in the network. An exciting future work would be exploring Machine Learning for dependability evaluation on various characteristics such as reliability, availability, security, safety.

As surveyed, the trend in research is migrating from simulation-based modeling to prediction models. These prediction models make use of the Machine Learning techniques such as Random Forest, Decision Tree, Linear Model, Support Vector Machine, Tree Model from Genetic Algorithm, and Bayesian Regularized Neural Networks. The current trend also focuses on the concept of fuzzy logic and provides more a adaptable solution to dependability problems.

Further, the buzzwords for research in the direction of Machine Learning include Artificial Intelligence, Data Science, Soft Computing, Intelligent Neural Networks, Fuzzy Logic, and Genetic Algorithm such as ACO (Ant Colony Optimization, PSO (Particle Swarm Optimization), Deep Learning, Multi-Objective Optimization, Multi-Parametric Analysis, Collaborative Learning Optimization.

The research timeline in Figure 2.2 elaborates on the trend of dependability in communication networks since 1992. This timeline shows that the trend in programming languages has also mitigated from Stochastic Petri Nets (SPN) to Prototype models, which has further moved to network simulator and the real testbeds. These approaches were not much intelligent, dynamic and adaptable. This led to the advancement in techniques such as soft computing, Machine Learning and also paved a way for programming languages such as R, Python, MATLAB (which has an inbuilt module for soft computation).

In the future, a hybrid technique of prediction based simulation or modeling and also the intelligent computation methods can be explored for better optimization of networks or for more dependable networks.

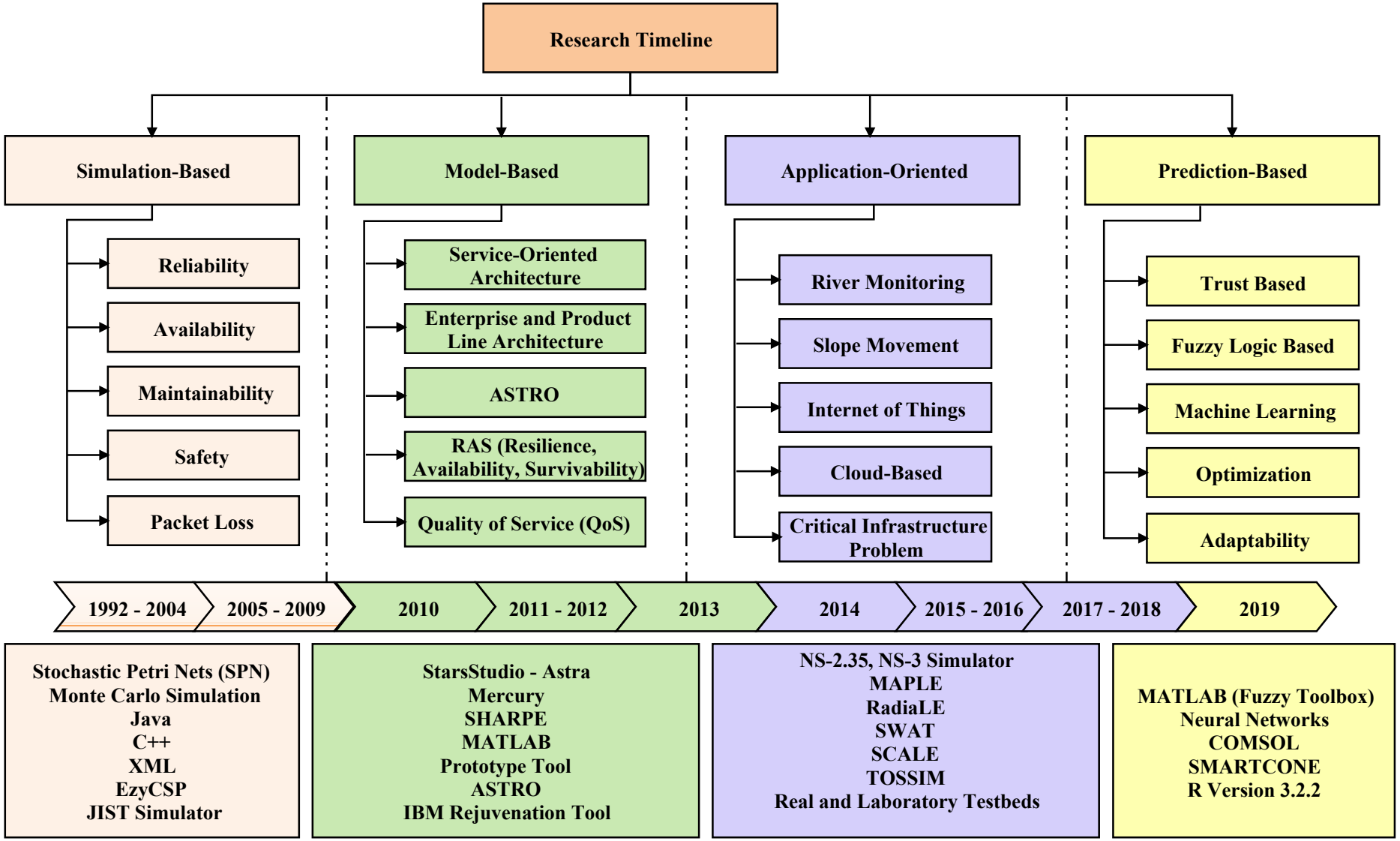


FIGURE 2.2: Research Timeline

Chapter 3

Reliable Network Prediction of Wireless Sensor Networks

"The greatest challenge to any thinker is stating the problem in a way that will allow a solution."

-Bertrand Russell

The term reliability[1] is defined as the capability of a network to perform its intended task under certain conditions for a stated time span. It can also be defined as, "a measure of the continuity of correct service". With reference to Figure 3.1, detailed anatomy of reliability is presented along with the basic definitions of the underlying concepts. There are three basic terms governing reliability which are threats, attributes, and means. Threats popularly known as impairments, are undesirable in reliable communication and encompass the popular Fault $\rightarrow$ Error $\rightarrow$ Failure chain. The fault is defect in a network. An Error shows the deviation between expected and actual outcomes. Whereas, failure is the inability of a network to perform its intended task. Attributes are the properties expected from a network and, include availability, maintainability, and testability. Availability is defined as, "the ability of a network to be in a state to perform a required function at a given instant of time within a given time interval; assuming that the external

[1]The contents of the chapter are peer-reviewed and published in:
Jasminder Kaur Sandhu, Anil Kumar Verma, Prashant Singh Rana, "A Novel Framework for Reliable Network Prediction of Small Scale Wireless Sensor Networks (SSWSNs)", Fundamenta Informaticae, SCI-Indexed, Impact Factor 1.204

resources if required, are provided". Maintainability is the ability of a network to undergo repairs and evolutions making it more adaptable to the current demand of users. Testability describes the degree to which a network facilitates the establishment of the test criteria. Means facilitates the development of a dependable network, which includes fault avoidance (prevents the occurrence of faults), fault tolerance (make network functional in case of presence of certain faults), fault detection (recognizing faults) and fault restoration (reconfigure the network in presence of faults).

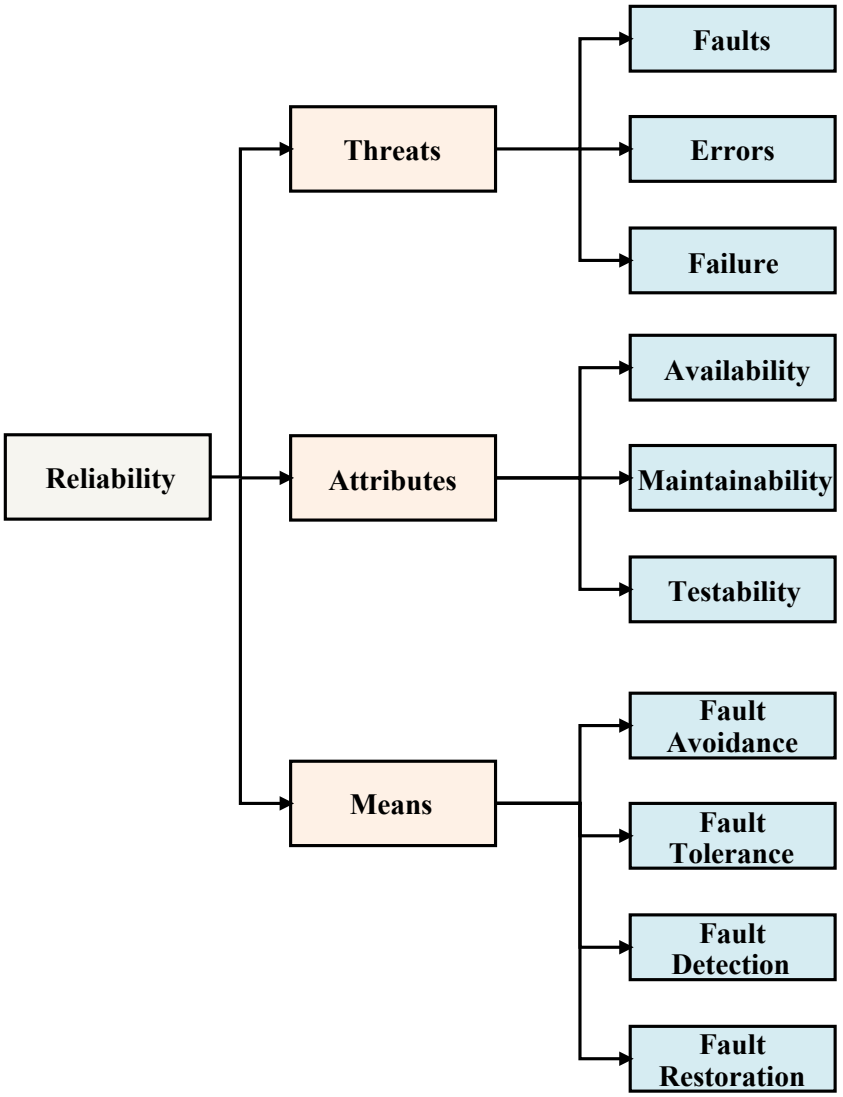


FIGURE 3.1: The Anatomy of Reliability

The task of predicting reliability while providing services with a much higher confidence level is a complicated task. Reliability requirements vary according to a particular application and for a particular user. Reliability contributes eloquently to design optimization and achieving measurable performance objectives. Thus, to make a

particular application reliable, it is crucial to know what hinders a network from providing correct service at the required time instance.

In SSWSNs, the vital objective is to maximize the network lifetime. It is dependent on many factors such as data transmission and collection, the node density, the protocol being used and lastly upon reliability of the network. The reliability is the most quantifiable feature of network design which focusses on faults. Faults are linked directly to the packet drop or the packet delivery ratio in a network. The network design can be considered from two different perspectives, as illustrated in Figure 3.2. The analysis of design concerning reliability focuses on the Fault $\rightarrow$ Error $\rightarrow$ Failure chain. These attributes can be easily modeled and hence design can be optimized using a Machine Learning concept [18]. However, the security perspective cannot be accurately modeled due to the fact that intentional malicious attacks on the network do not follow any predictable pattern.

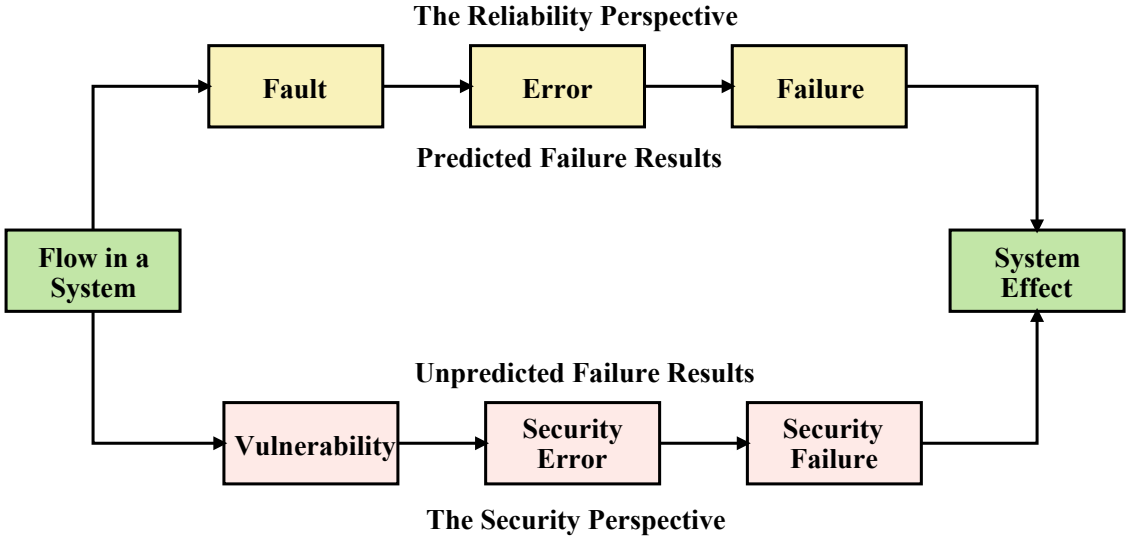


FIGURE 3.2: The Network Design Perspective

This chapter focuses on the reliability perspective of network design. The reliability can be predicted easily based upon the network lifetime [85], which in turn is dependent on the Data Flow (DF), Protocol Name (PN) and the Number of Nodes (NN). The data flow optimization in a network can also help in reducing the number of faults. So far no Machine Learning technique has been devised which calculates reliability considering these core network parameters (DF, PN, and NN). Further, for reliable communication in the futuristic networks, the three parameters taken care of are the DF, PN and the NN.

The DF parameter regulates the flow of data through a network and hence governs the performance of that network. The PN helps in undermining the specific area of application in which the SSWSNs is employed. The NN parameter reflects the density of the nodes in a network (particularly important for SSWSNs).

Further, this chapter presents an overview of SSWSNs and reliability in network prediction. The figure 3.3 presents a diagrammatic view of the work carried out in this chapter. The layer 3 is the lowest layer and deals with the SSWSNs structure. It contains the machine layer for sensor node placement, network layer provides communication between the sender and receiver, data processing and visualization. The middle layer 2 provides the implementation of the SSWSNs in various application areas. The top-most layer 1 tackles various challenges incurred in this network. Some of the challenges have been considered in this work, namely DF, PN, and the NN.

This chapter presents the first work of this type in the existing body of literature. The contributions of this work to the existing trend are multi-fold as summarized below:

1. A novel framework has been proposed to measure SSWSNs reliability.
2. This framework takes care of known parameters as well as application-dependent parameters such as data flow, protocol name and the number of nodes for a reliable SSWSNs.
3. The prevalent prediction techniques such as Cubist, Random Forest, Weka Lazy Modifier, Bagged CART, Neural Network, Support Vector Machine, Conditional Inference Tree are applied to evaluate this framework.
4. An ensemble model is also devised as an optimal prediction model for DF, PN and the NN.

3.1 Proposed Framework for Reliability Analysis

Mechatronics integrates the mechanical, control, electronics, and computer domain. This makes it a multidisciplinary area. Also, this integration has led to the increase in complexity of design. The integrated methods for designing are continuously

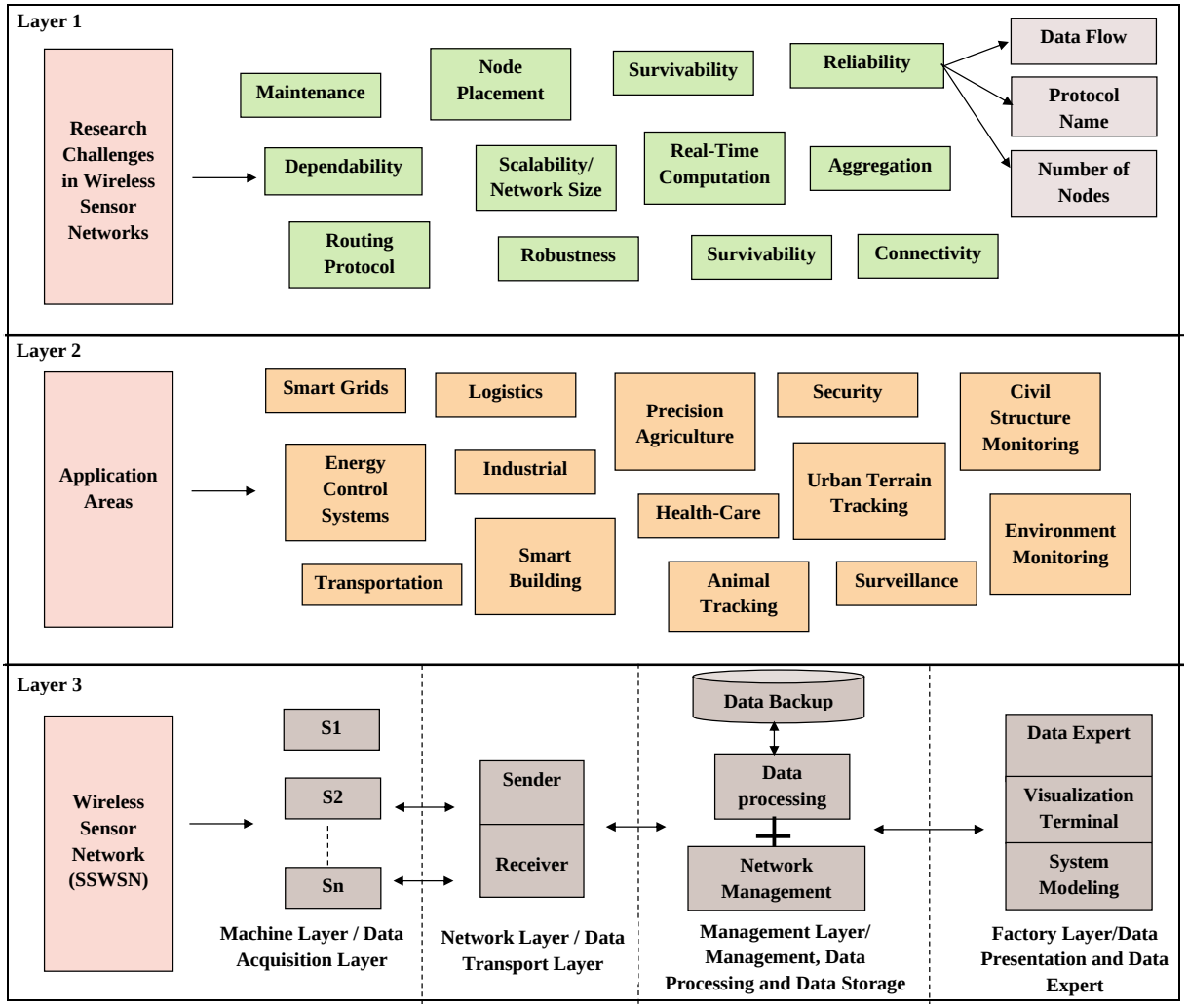


FIGURE 3.3: The Layered Approach to Problem Design

developing, but the assessment of dependability in these designs does not follow the recent trend. As a result, number of designs developed lack quality features. The solution to this issue to wrap the entire design with the dependability layer or more specifically with the reliability oriented designing. The proposed reliability analysis framework is inspired by Mechatronics Dependability Evaluation Framework, the V-Model [Refer Figure 3.4] and is adapted for the SSWSNs [86]. The proposed reliability framework adapted for a SSWSN has been depicted in Figure 3.5. This framework would provide realistic results as it will be based on the real-world data.

Components, Connectors, and Configurations form the basis of a framework for any communication network [86, 87]. Components are computational units of the framework such as the design optimization module and the result analysis module. Connectors are

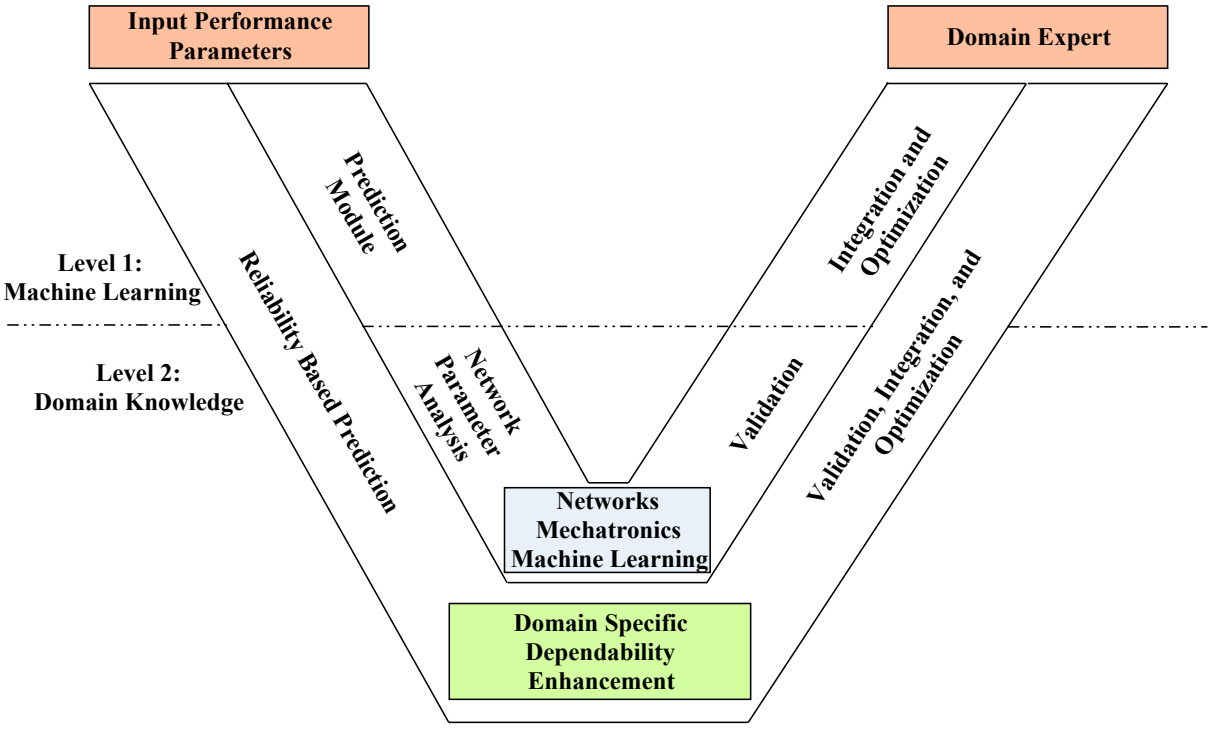


FIGURE 3.4: V-Model for Wireless Sensor Networks

the entities that enable interaction between various components such as the prerequisites of the framework labeled as the performance parameters are supplied as input to the network simulation modeling component. Configuration is the basic topography of the framework consisting of components and connectors.

The reliability framework provides an assessment about the performance level of a network in terms of reliability, taking into account the prerequisites such as the number of packets sent, received, dropped, packet delivery ratio and throughput. The reliability here means the trustworthiness of transmission and is predicted based upon the flow characteristics [11]. The analysis depends solely on the prerequisites which fit best as per the knowledge of the network reliability or network performance experts. The prerequisites [88, 89] which can also be termed as the performance parameters of SSWSNs are defined as follows:

- **Transmitted Packets (SP)**: The total number of packets disseminated from the sender to receiver in a network scenario.
- **Received Packets (RP)**: The total number of packets collected at the receiver end in a network scenario.

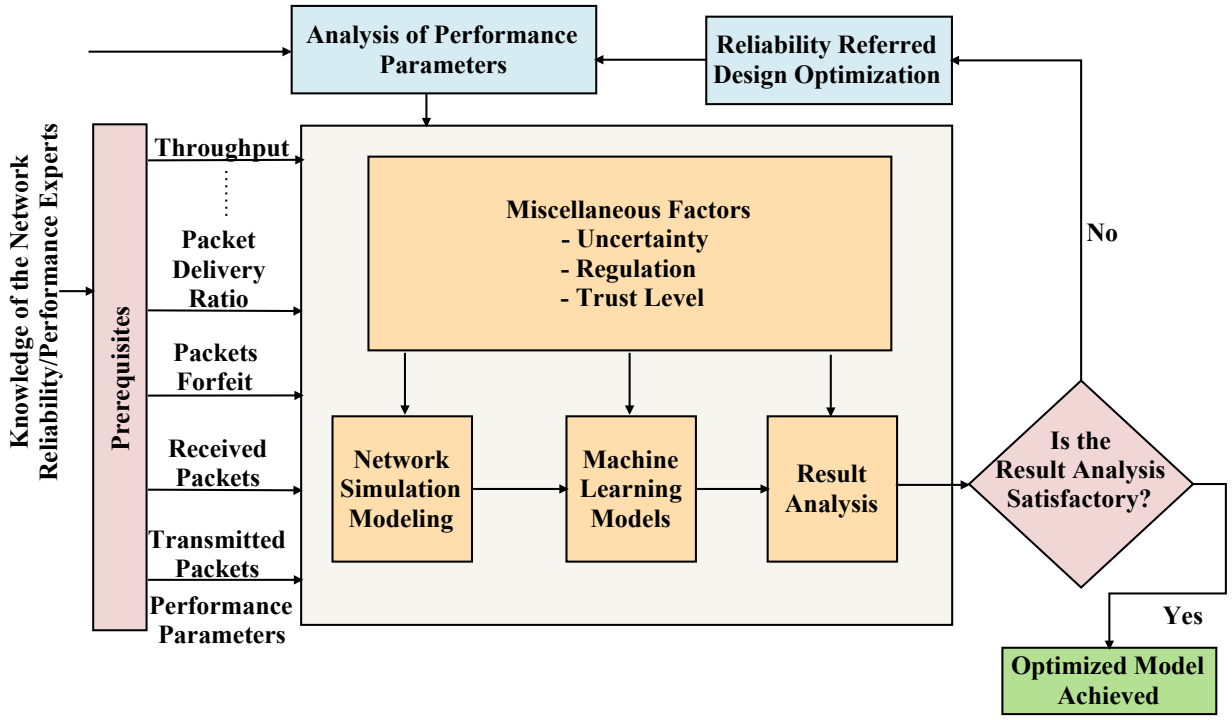


FIGURE 3.5: The Reliability Framework

- **Packets Forfeit (PF)**: Summing up the packets forfeited during transmission from source to destination.
- **Agents for Routing (RA)**: These are responsible for communicating the packet with the neighboring node.
- **Routing Overhead (RO)**: It tells the total routing packets required for communication in SSWSNs.
- **Packet Delivery Ratio (PDR)**: Ratio of packets delivered to packets sent and is expressed in terms of percentage.
- **Average Path Length (APL)**: The shortest route between the pair of nodes is calculated, which is then added. The resultant is divided by the count of these pairs and is termed APL.
- **Throughput (TH)**: It is a measure of how fast we can actually send data through a network and is expressed in Kbps (Kilobits per second). It is preferred over bandwidth because it is a practically measured parameter, whereas, the bandwidth is more of a theoretical parameter.

- **Data Flow (DF)**: It is defined as the rate at which data flow through a network and is represented in Mbps (Megabits per second).

- **Protocol Name (PN)**: It determines the type of routing protocol used in a network. The network protocols are classified as proactive, reactive, hybrid. This work considers two reactive protocols namely, AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing).

- **Number of Nodes (NN)**: The total sensor nodes present in a network determine the number of nodes. The deployment of nodes can be sparse or dense according to the region of interest.

Keeping in view these prerequisites, a dataset is created in the immediately next modeling process. This dataset is designed with the help of simulations and the prediction results are thus considered for optimized reliability. The evaluation process consists of three phases:

- **Network Simulation Modeling**: In this step, the network scenario is constructed using a network simulator NS-2.35. The details of network simulation using NS-2.35 can be referred from Annexure-A. With variable data flow rates, a dataset is constructed using routing protocols AODV and DSR. This dataset is given as input in the next step explained below. The correlation between all the features of the dataset is discussed in Figure 3.6.

- **Machine Learning Models**: Based on the above-created dataset, various Machine Learning models are applied such as Random Forest, Neural Network. The results thus obtained include RMSE (Root Mean Square Error), correlation, accuracy which are important terms in the analysis of network efficiency. The main reason for using these models is that they predict the DF rate, PN and NN. Also, an ensemble is designed to optimize the result of the existing models used. The details of R programming used for implementing Machine Learning models can be referred from Annexure-B.

- **Result Analysis**: The results obtained above are analyzed (in terms of Correlation, Coefficient of Determination, Root Mean Square Error , Time Taken, and Accuracy), processed and are thus checked to ensure whether the network requirement is fulfilled or not. If the network requirement is fulfilled, it is then

checked for reliability attribute; else, the design is further optimized using the NS-2.35 modeling.

Certain **Miscellaneous Factors** govern this evaluation process such as the degree of uncertainty involved in these predictions, the regulations imposed by law-making authorities, the trust level (data trust, node trust, hybrid trust).

More factors of interest can be added to this category, depending upon the application for which this particular framework is used. This makes the framework flexible and generic according to a particular application domain. The assumption of constant buffer capacity has been considered as a miscellaneous factor throughout this work.

3.1.1 Network Simulation Modeling

The dataset is constructed using an NS-2.35 simulator. In constructed wireless scenarios, omnidirectional communication takes place between the sensor nodes using routing protocols AODV and DSR, which are reactive. The simulation setup has been constructed for only SSWSNs. The sensor nodes are randomly deployed in the network. The communication takes place between sensor nodes, manager nodes and the base station using the Relative Identification and Direction-Based Sensor Routing (RIDSR) scheme inspired from [90]. The sensor nodes are deployed in an area covering 1000 $\times$ 1000 meters with a simulation time of 20 ms. The simulation parameters set for generating these scenarios are discussed below:

TABLE 3.1: NS-2.35 Simulation Parameters

S.No.	Parameter	Value
1	Channel Type	Wireless
2	Radio-propagation model	Two-Ray Ground
3	Interface queue type	DropTail/CMUPriQueue
4	Antenna Model	Omnidirectional
5	Max packet in Interface Queue	150
6	Number of Nodes	5-50
7	Data Flow	0.1-10 Mbps
8	Routing protocols	AODV/DSR
9	X dimension of topography	1000 m
10	Y dimension of topography	1000 m
11	Simulation Time	20 ms

3.1.2 Machine Learning Methods

The following subsection describes the dataset and Machine Learning methods used for decision-making. The dataset is created by using NS-2.35 simulator [91, 92] with various performance parameters. A partial dataset is presented in this section.

3.1.2.1 Description of Dataset

The dataset consists of 10000 records with 11 performance parameter values. The modeled structure convey in-depth knowledge of the reliability prediction framework. For prediction of DF, PN and the NN, these performance parameters are kept as the target variables and rest of the variables in the dataset acts as input to the Machine Learning model. The Table 4.3 describe features of dataset also termed as performance parameters constructed for this work:

The simulated values specified above are calculated from the trace file produced during the NS-2.35 simulation. These trace files are analyzed with the help of awk and perl scripts. The dataset hence generated, contain integer, real values and the values to be predicted. The dataset is completely balanced with no missing and redundant entries. The dataset used in this work is available as a supplement at *http://bit.ly/SSWSN-Reliability*.

TABLE 3.2: Description of the dataset

SN	Feature	Information	Description
1	SP	Transmitted Packets	Simulated Value
2	RP	Received Packets	Simulated Value
3	PF	Packets Forfeit	Simulated Value
4	RA	Agents for Routing	Simulated Value
5	RO	Routing Overhead	Simulated Value
6	PDR	Packet Delivery Ratio	Simulated Value
7	APL	Average Path Length	Simulated Value
8	TH	Throughput	Simulated Value
9	DF	Data Flow	Range = 0.1 - 10 Mbps
10	PN	Protocol Name	0 = AODV, 1 = DSR
11	NN	Number of Nodes	Range = 5-50

For better readability, kindly refer to Section 3.1.

TABLE 3.3: Sample Dataset

SP	RP	PF	RA	RO	PDR	APL	TH	DF	PN	NN
750	745	5	0	0	99.33	1	153.6	0.3	0	5
1001	973	28	0	0	97.20	1	205	0.4	0	5
1751	1655	96	0	0	94.51	1	358.6	0.7	0	5
2000	1825	175	0	0	91.25	1	409.6	0.8	0	5
2250	1825	425	0	0	81.11	1	460.8	0.9	0	5
9250	1825	7425	0	0	19.72	1	1894.4	3.7	0	5
9501	1825	7676	0	0	19.20	1	1945.8	3.8	0	5
9750	1825	7925	0	0	18.71	1	1996.8	3.9	0	5
6501	484	6017	974	14.98	7.44	3.01	1530.47	2.6	0	40
6750	484	6266	974	14.42	7.17	3.01	1581.47	2.7	0	40
7000	484	6516	974	13.91	6.91	3.01	1632.67	2.8	0	40
7251	484	6767	975	13.44	6.67	3.01	1684.28	2.9	0	40
7501	484	7017	974	12.98	6.45	3.01	1735.27	3	0	40
7501	583	6918	1483	19.77	7.77	3.54	1950.11	3	1	45
7750	598	7152	1503	19.39	7.71	3.51	1999.26	3.1	1	45
15251	667	14584	1336	8.76	4.37	3	3396.61	6.1	0	25
15501	667	14834	1336	8.62	4.30	3	3447.81	6.2	0	25
251	250	1	0	0	99.60	1	103.83	0.1	1	5
501	500	1	0	0	99.80	1	206.23	0.2	1	5
750	750	0	0	0	100.00	1	308.43	0.3	1	5
1001	1000	1	0	0	99.90	1	411.03	0.4	1	5
1251	1250	1	0	0	99.92	1	513.43	0.5	1	5
1501	1500	1	0	0	99.93	1	615.83	0.6	1	5
1751	1750	1	0	0	99.94	1	718.23	0.7	1	5
2000	1999	1	0	0	99.95	1	820.22	0.8	1	5
2250	2005	245	0	0	89.11	1	872.65	0.9	1	5
2501	2004	497	0	0	80.13	1	923.65	1	1	5
2750	2004	746	0	0	72.87	1	974.44	1.1	1	5
3001	2006	995	0	0	66.84	1	1026.46	1.2	1	5
3250	2006	1244	0	0	61.72	1	1077.45	1.3	1	5
3500	2004	1496	0	0	57.26	1	1127.83	1.4	1	5
3751	2004	1747	0	0	53.43	1	1179.44	1.5	1	5
4000	2006	1994	0	0	50.15	1	1231.05	1.6	1	5
4251	2004	2247	0	0	47.14	1	1281.84	1.7	1	5
251	249	2	267	106.37	99.20	2.07	157.08	0.1	1	10
501	498	3	517	103.19	99.40	2.04	310.68	0.2	1	10
750	748	2	767	102.27	99.73	2.03	463.87	0.3	1	10
1001	992	9	1016	101.50	99.10	2.02	616.45	0.4	1	10
1751	990	761	1018	58.14	56.54	2.03	769.64	0.7	1	10
2000	989	1011	1017	50.85	49.45	2.03	820.22	0.8	1	10
2250	987	1263	1020	45.33	43.87	2.03	872.45	0.9	1	10
2501	986	1515	1022	40.86	39.42	2.04	923.24	1	1	10
2750	988	1762	1021	37.13	35.93	2.03	974.03	1.1	1	10
3001	984	2017	1019	33.96	32.79	2.04	1024.61	1.2	1	10
3250	987	2263	1021	31.42	30.37	2.03	1077.04	1.3	1	10
3751	987	2764	1020	27.19	26.31	2.03	1179.03	1.5	1	10
4500	986	3514	1021	22.69	21.91	2.04	1332.84	1.8	1	10
4751	988	3763	1021	21.49	20.80	2.03	1384.24	1.9	1	10
5000	987	4013	1022	20.44	19.74	2.04	1435.24	2	1	10

The prediction problem is of regression type (DF) and classification type (PN and NN both). The PN prediction lies in the category of binary classification type problem as only AODV and DSR is used in the course of this work. AODV stands for Ad-Hoc On-Demand Distance Vector. DSR stands for Dynamic Source Routing. These are reactive protocols which enables route creation only on-demand. The NN prediction is a multiclass classification type problem since its value varies from 5 to 50. The network is scalable and more sensor nodes can be added to the scenario. Table 4.4 demonstrates the dataset used in the course of this work with the shuffled arrangement of records.

Correlation studies the statistical relationship between attributes of the dataset having dependence. The range of the correlation coefficient varies between -1 to +1. Every correlation has two qualities: strength and direction. The direction of a correlation is either positive or negative. In a negative correlation, the variables have an inverse relationship. Whereas, in the case of positive correlation, the variables have a direct relationship. The strength of correlation is dependent upon the value of its coefficient. The importance of correlation is to predict future behavior on the basis of relationship analysis between the different variables of the dataset.

A matrix is created, showing correlations among pairs of variables contained in the dataset. This method for measuring the linear dependence between any two-variable pair in the dataset is known as the Pearson Correlation. This matrix is known as the Correlation Matrix and is represented by CM.

A CM provides the summarization of the features of the dataset for advanced analysis. The main goal of the CM is to analyze the data patterns. The CM is designed with the help of R programming language.

$$\mathbf{CM} = \begin{array}{c} \\ \textbf{PDR} \\ \textbf{DF} \\ \textbf{PN} \\ \textbf{NN} \\ \textbf{Reliability} \end{array} \begin{array}{c} \begin{array}{ccccc} \textbf{PDR} & \textbf{DF} & \textbf{PN} & \textbf{NN} & \textbf{Reliability} \end{array} \\ \begin{pmatrix} 1 & -0.6081 & 0.0252 & -0.2293 & 1 \\ -0.6081 & 1 & 0 & 0 & -0.6081 \\ 0.0252 & 0 & 1 & 0 & 0.0252 \\ -0.2293 & 0 & 0 & 1 & -0.2293 \\ 1 & -0.6081 & 0.0252 & -0.2293 & 1 \end{pmatrix} \end{array}$$

The CM suggests that maximum correlation exists between parameter PDR and Reliability. Also, CM contain redundant values. Hence, the upper triangular representation of the CM has been presented in Figure 3.6. The negative correlations are in blue and positive correlations in red color.

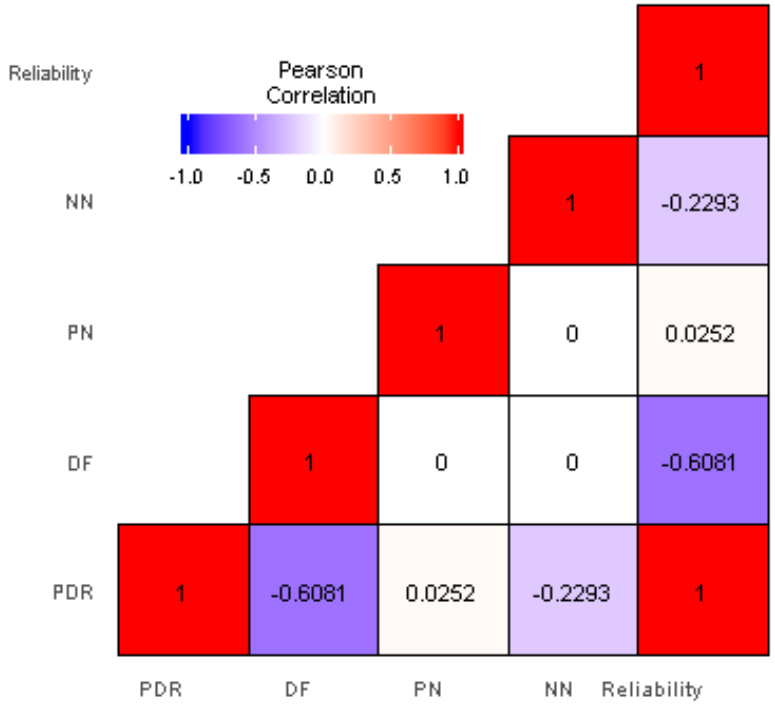


FIGURE 3.6: The correlation between dataset features

3.1.2.2 Machine Learning Models

Various Machine Learning models are used in the course of this work for predicting three network parameters namely, Data Flow(DF), Protocol Name(PN) and Number of Nodes(NN).

The models include: Cubist, Random Forest, SVM, Neural Networks, Weka Lazy Model, Conditional Inference Tree, Bayesian Regularized Neural Networks, Bagged MARS, Bagged CART, Tree Model from Genetic Algorithms. The details of these models along with their tuning parameters are available in Table 4.5):

1. Cubist (cubist): Cubist generates rule–based models for prediction of data. It consists of different cases and every case has two parameters target attribute and dependent variable. The cubist model aims to analyze the target value based on the attribute values. Cubist usually constructs a model which consists of several rules and therefore every rule is associated with other rules. If all the rules are satisfied then they form a linear expression.

2. Random Forest (randomForest): Random inputs are given to produce a forest of trees. It can be applied to both regression and classification type problems and hence is of dual usage. It also provides information about feature importance. It is an ensemble-based learning method. The main task performed by this model is to merge multiple decision trees into a single tree through which predictions can be closer to the actual value.

3. Support Vector Machine (SVM): It is again of dual usage. Decision planes form an essential component of this model. Decision boundaries are defined based upon these decision planes. It uses two types of variables, namely categorical and continuous. As the name suggests, the categorical variable uses two categories of 0 and 1 and the continuous variable takes multiple values as is the case with regression.

TABLE 3.4: Machine Learning Techniques

Technique	**Method Used**	**Package Included**	**Tuning Parameters**
Cubist[93]	cubist	Cubist	committees=50
Random Forest[94]	rf	randomForest	mtry=500, sampling=bagging
SVM[95]	svm	e1071	nu=10, epsilon=0.5
Neural Networks[96]	neuralnet	neuralnet	hlayers=10, MaxNWts=10000, maxit=100
Weka Lazy Model[97]	IBk	glm	control = *Weka_control*(), options = NULL
Conditional Inference Tree[98]	ctree	party	None
Bayesian Regularized Neural Networks[99]	brnn	brnn	None
Bagged MARS[100]	bagEarth	earth	B=50
Bagged CART[100]	bagging	ipred	coob=TRUE
Tree Models from Genetic Algorithms[101]	evtree	evtree	minbucket = 10, maxdepth = 2

4. Neural Network (NN): It deals with non-linear and complex relationships between a response variable and the predictor. It uses different algorithms such as backpropagation, multi-layer perceptron.

5. Weka Lazy Model (WLM): R interfaces to Weka lazy learners. IBk provides a k-nearest neighbor classifier.

6. Conditional Inference Tree (CIT): It includes two steps of execution searching for a split point and variable selection. It requires a conditional inference framework and is based on recursive partitioning.

7. Bayesian Regularized Neural Networks (brnn): The Bayesian Regularization for Neural Networks (BRNN) is widely used in prediction problems for multiple fields of various applications and more recently, for genome-enabled prediction.

8. Bagged MARS(earth): It is of dual usage and is used for classification and regression type problems. It works by reduction technique proceeding backward while selecting features.

9. Bagged CART (bagging): It follows a tree-like arrangement consisting of nodes and leaves. The training data is partitioned at each node using the conditional if-then construct. The non-linear relationships can be easily studied using this model.

10. Tree Model from Genetic Algorithm (evtree): This model uses an evolutionary scheme for constructing trees. It maintains a simple structure and provides better performance than rpart, ctree. Hence, it yields optimized results.

3.1.2.3 Result Analysis

The detailed result analysis of prediction of DF, PN and the NN is discussed in Section 3.4.

3.2 Mathematical Formulation of Reliability

The reliability is classified as packet, event, packet and event reliability. The evaluation metric for packet reliability is the packet reliability faults and can be measured by the

number of packets dropped or corrupted. This finally leads to node failure, communication failure, and security failure, as shown in Figure 3.7:

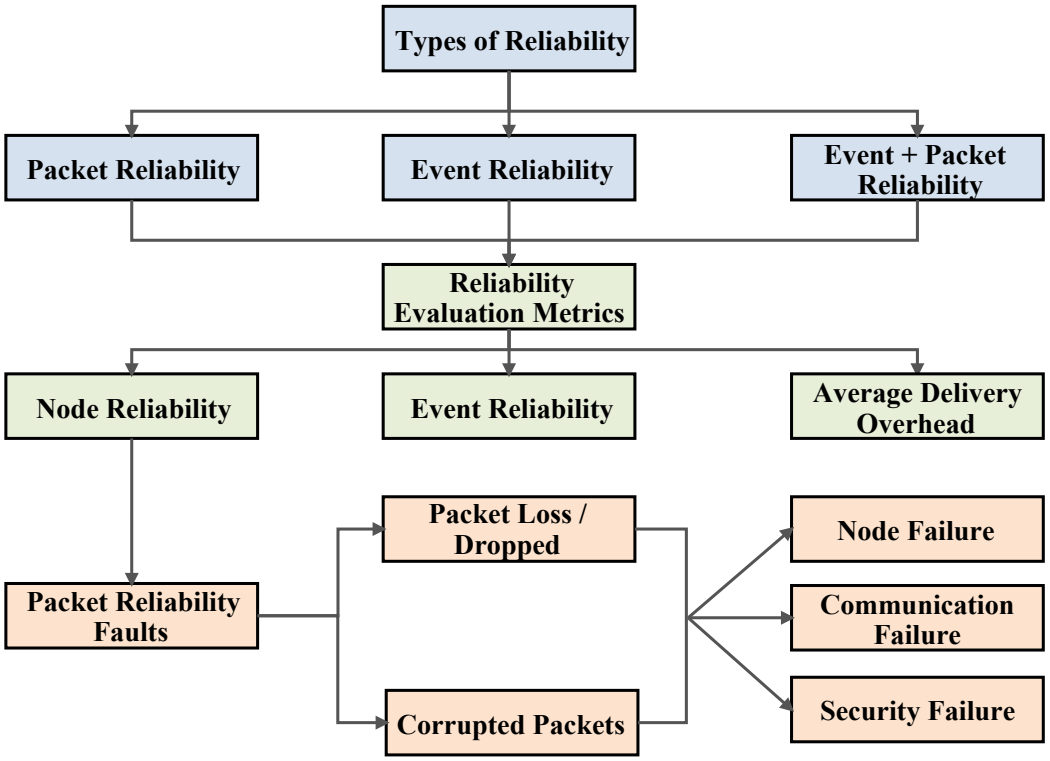


FIGURE 3.7: The Reliability Evaluation Paradigm

For further reading on reliability, faults, error, and failure, refer [102, 103]. This work addresses the node reliability, which is a reliability evaluation metric. It is defined as the ratio of packets forfeit in the network to the number of packets generated by the node [104]. Hence, to enable reliable communication and reduce the number of faults, DF needs to be regulated as well as taking into account the PN and NN. Now, assuming F to be the time to failure of a network. The probability of network survival until some time say *tym* is termed as the reliability *R(tym)* of the network. So, *R(tym) = P(F > tym) = 1 - D(tym)*, where D is the distribution function of the network lifetime, F.

Hence, deriving reliability function according to [105], it yields reliability as:

$$R(tym) = 1 - \frac{N_f(tym)}{N_0} \tag{3.1}$$

The total number of data sent on the network is represented by N_0 and is a constant value. $N_f(tym)$ represents the failure or the number of packets forfeited in the network. The

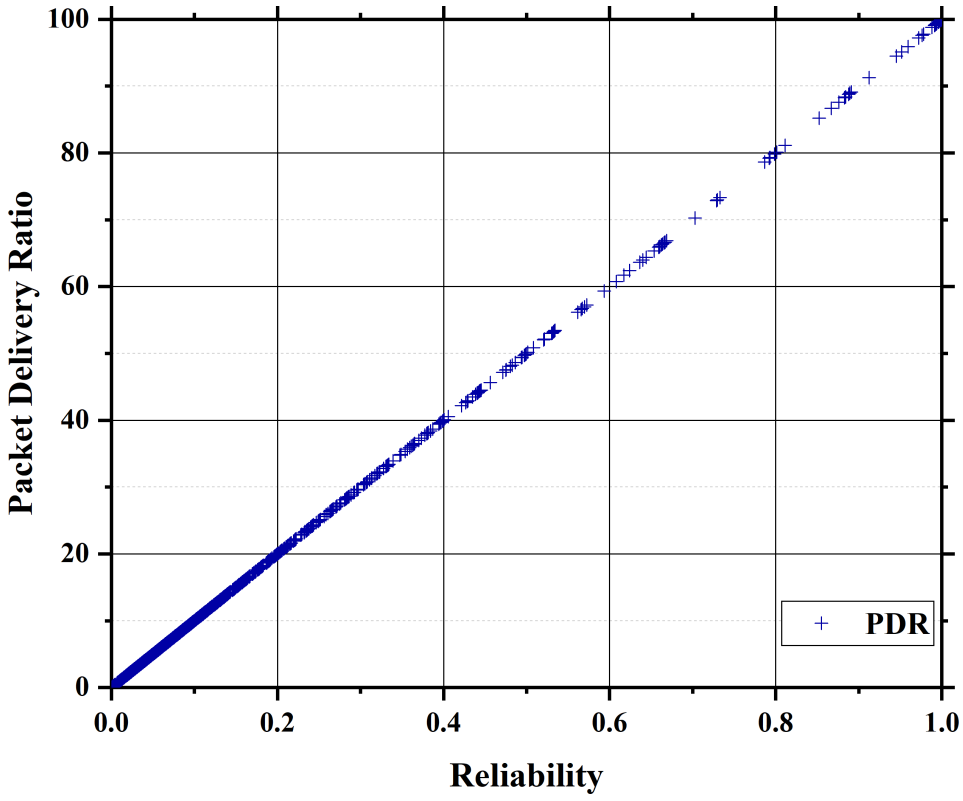


FIGURE 3.8: Reliability Vs Packet Delivery Ratio (PDR)

value of reliability lies between 0 and 1. Closer the value to 1, better is the reliability. The Reliability Vs PDR (Packet Delivery Ratio) plot in Figure 3.8 shows that with an increase in PDR, the reliability of network enhances:

Also, the effect of DF on the reliability of a network can be understood from Figure 3.9. This shows that if the DF increases, the reliability of the network decreases considering the assumption of constant buffer size, which further adds to the miscellaneous factors in the proposed framework.

3.3 Performance Evaluation

The workflow of our framework is demonstrated in Figure 3.10. Firstly, the performance parameters of the network are quested. Secondly, based upon these parameters a dataset is constructed. Thirdly, the missing, unused and redundant values are removed from the dataset, this is called Data Cleansing phase. These features are dropped because they cannot be calculated for an unknown network. In the next phase, various Machine

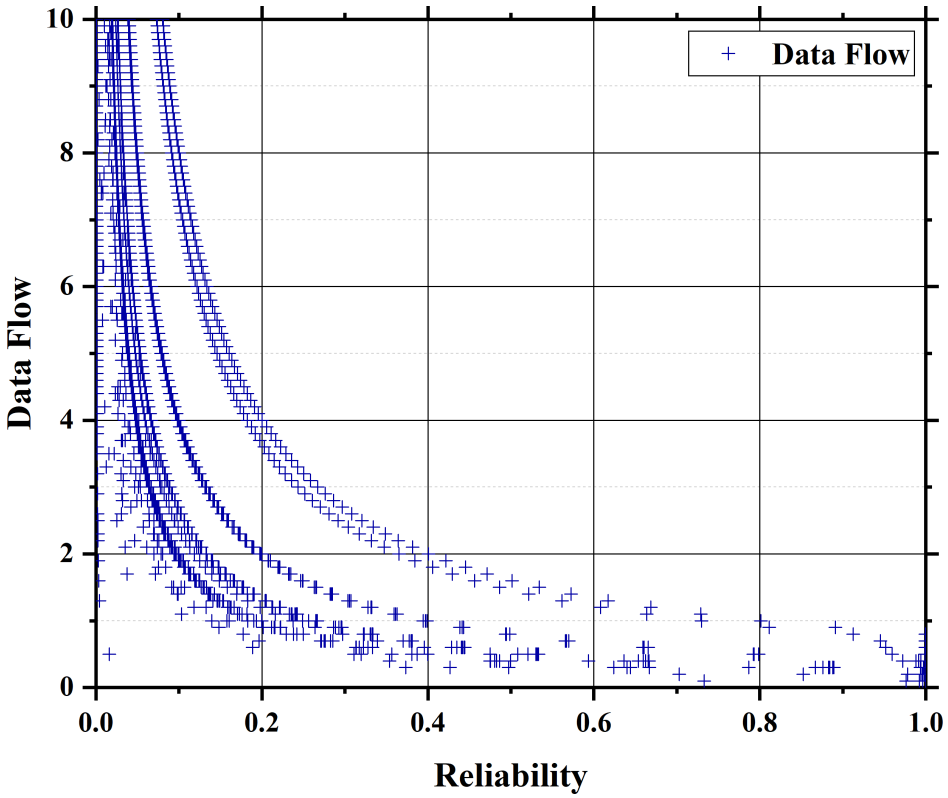


FIGURE 3.9: Reliability Vs Data Flow

Learning techniques are applied and an ensemble model is constructed to calculate the optimized accuracy of the model. Lastly, the results are analyzed in the light of reliability of a network.

In modeling our network, various Machine Learning techniques are used to predict the output variables DF, PN, and NN. Various other performance metrics elaborated in Table 4.3 form the basic input variables. All the Machine Learning models implemented use the formula as described below:

$$DF \sim f(SP, RP, PF, PDR, TH) \tag{3.2}$$

$$PN \sim g(SP, RP, PF, PDR, TH) \tag{3.3}$$

$$NN \sim h(SP, RP, PF, PDR, TH) \tag{3.4}$$

In the above equations, function $f()$, $g()$, $h()$ specifies the input variables to predict the DF, PN and NN respectively.

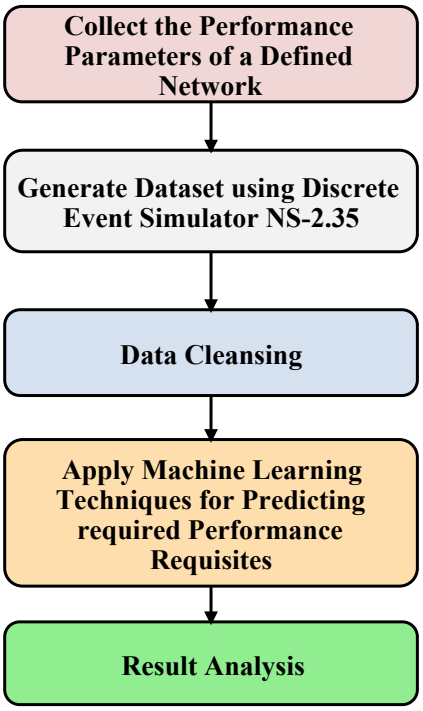


FIGURE 3.10: Work Flow of the Proposed Framework

3.3.1 Evaluation Parameters for Data Flow Prediction

The DF parameter falls under the category of regression type problem. Various metrics that can be calculated for a regression type problem are the correlation, coefficient of determination, Root Mean Square Error and accuracy. The following sub-sections explain these metrics:

3.3.1.1 Correlation (r)

Correlation provides information about how actual and predicted values are linked. It can be calculated as:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{3.5}$$

where, a depicts actual value, b depicts predicted value, $\bar{a}$ represents actual values mean, $\bar{b}$ represents predicted values mean and n is the number of instances. The value of correlation lies in [0,1]. More the value tends towards 1, the better is the correlation.

3.3.1.2 Coefficient of Determination (R^2)

The coefficient of determination (R^2) is the primary outcome of regression evaluation and its value lies in the interval $0 < R^2 < 1$. More the value tends towards 1, the better is the regression model. If the value is zero that means the regression model is a failure. Mathematically, it is the square of correlation. It can be computed as:

$$R^2 = r * r \tag{3.6}$$

3.3.1.3 Root Mean Square Error

Root Mean Square Error (RMSE) tells us how much error is present between the actual and predicted values. In short, it is used for numerical analysis of predicted results and can be computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \tag{3.7}$$

where, a is the actual target, p is predicted target and n are the total number of instances.

3.3.1.4 Accuracy

The accuracy represents how close predicted value is to the actual value, within acceptable error limits. It predicts which model best suits a problem. It is computed as:

$$Accuracy = \frac{100}{n}\sum_{i=1}^{n} q_i \tag{3.8}$$

$$q_i = \begin{cases} 1 & if abs(p_i - a_i) \leq err \\ 0 & otherwise \end{cases}$$

where, a depicts actual target, p depicts predicted target, err represents acceptable error and n is number of instances.

3.3.2 Evaluation Parameters used for Protocol Prediction

The protocol used in the network falls under the binary class type problem as our work uses two protocols namely AODV and DSR. Various metrics that can be calculated for a binary class type problem are H, Gini coefficient, AUC, AUCH, Kolmogorov-Smirnoff, MER, MWL, specificity, sensitivity, ROC and accuracy. The following sub-sections describe these metrics:

3.3.2.1 H

A rational substitute to measure the performance of a classification problem is the H-measure. This measure represents the Area Under the ROC Curve and also yields a center plot using a Beta prior. It is calculated with the help of hmeasure package available in R.

3.3.2.2 Gini Coefficient

The disparity of distribution is calculated using the Gini coefficient and its value lies between 0 and 1. It can be computed as:

$$Gini = 2AUC - 1 \tag{3.9}$$

3.3.2.3 AUC and AUCH

The AUC is a measure of the Area Under the ROC Curve and its value lies between 0.5 and 1. It depicts the quality of models used for classification problems. It is computed as:

$$AUC = \frac{Gini + 1}{2} \tag{3.10}$$

AUCH represents the area under the Convex Hull of the Receiver Operating Characteristic (ROCH) curve.

3.3.2.4 Kolmogorov-Smirnoff

An alternative metric that seeks to jointly consider specificity and sensitivity is the Kolmogorov-Smirnoff (KS) statistic. It corresponds to the maximum value their sum takes, as the threshold is varied. This also attains an intuitive graphical interpretation, as the maximum vertical distance between ROC and the diagonal.

3.3.2.5 MER and MWL

MER metrics represent the minimum error rate. The threshold value here acts as a free parameter. MWL metrics represent the Minimum Cost Weighted Error Rate. It is related to the KS statistics. The cost guides the threshold value for this measure.

3.3.2.6 Specificity and Sensitivity

Specificity is the True-Negative(TN) rate of a classifier and is measured when sensitivity is held fixed at 95 percent. Sensitivity is the True-Positive(TP) rate of a classifier and is measured when specificity is held fixed at 95 percent. These can be computed as:

$$Specificity = \frac{TN}{FP + TN} \tag{3.11}$$

where, FP represents False-Positive rate.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.12}$$

where, FN represents False-Negative rate.

3.3.2.7 ROC Curve

It is an influential graphical tool for envisioning the performance of a learning algorithm over variant decision criteria [106]. It is used to study the behavior of algorithms, identify optimal behavior regions, perform model selection, and for conducting the comparative analysis of learning algorithms. This function is included in the pROC package available in R [107]. In a ROC plot, the x-axis represents False-Positive rate or Specificity and the y-axis represents the True-Positive rate or the Sensitivity.

3.3.2.8 Accuracy

True-Positive (TP) depicts a number of predictions that are positive, the actual value being positive. Similar is the case with True-Negative (TN). The accuracy is computed as:

$$Accuracy = \frac{TP + TN}{Total\ Data} * 100 \tag{3.13}$$

3.3.3 Evaluation Parameters for Predicting the Number of Nodes

The NN prediction falls under the multi-class classification type problem. The accuracy metric is computed for this problem. The following sub-section explains this metric:

3.3.3.1 Accuracy

True-Positive (TP) depicts a number of predictions that are positive, the actual value being positive. Similar is the case with True-Negative (TN). For measuring accuracy a multi-class type problem is converted into binary-class type problem. Accuracy in such cases can be computed by using Equation 3.13.

3.3.3.2 K-Fold Cross-Validation

The K-fold cross-validation is used to predict the robustness of the Machine Learning models used. Randomized partitioning of dataset yields p samples of the same size. One

such sub-sample is used as test data for validation purposes and rest p-1 sub-samples will be used for training purposes. This is called cross-validation and is iterated p number of times, hence yielding p results. These p results when averaged produces a single value. So, all data elements are involved in both training and testing. The validation of each data element is permitted exactly once thereby, increasing the precision of our results. In this work, the robustness of the best predictive model is obtained using 10 consecutive executions which are known as 10-fold cross-validation.

3.4 Results and Discussions

This section provides an insight into various Machine Learning techniques applied to the dataset segregated into train and test data. This dataset comprises of performance parameters governing a network which includes the basic requisites to the reliability framework. These models execute on their specifications (refer to Table 4.5) and are evaluated based upon RMSE, correlation, R^2, accuracy, AUC, Gini, AUCH, KS, ROC, MER, MWL, specificity, sensitivity for both regression and classification type problems. Lastly, K-fold validation is used to check the robustness of techniques applied for prediction.

3.4.1 Data Flow

In this section, the predicted outcome of the best five Machine Learning methods is presented. The models are executed by setting their tuning parameters (refer to Table 4.5). All the prediction methods use data which is divided into training and testing data fixed at 70% and 30% respectively. Two types of ensemble techniques have also been applied with equal and variant weights. The results thus computed for a single execution are shown below:

The performance in terms of accuracy for DF rate is shown in Table 3.6. This is the problem of regression type. The acceptable error range is set less than and equal to one. The accuracy has been calculated for 10 consecutive runs of the same model.

TABLE 3.5: Performance Comparison of Machine Learning Models for Data Flow

SN	Model Name	Correlation	R^2	RMSE	Accuracy
1	Cubist	1.00	1.00	0.00	100.00
2	Random Forest	1.00	1.00	0.07	99.40
3	WLM	1.00	1.00	0.15	99.10
4	CIT	1.00	0.99	0.13	96.20
5	BRNN	0.98	0.96	0.31	92.20
6	Ensm-EW	**0.99**	**0.98**	**0.01**	**99.60**
7	Ensm-VW	**1.00**	**1.00**	**0.00**	**99.90**

R^2 = Coefficient of Determination; Ensm-EW = Ensemble Model with equal weights; Ensm-VW = Ensemble Model with variable weights.

Also, to optimize the results an ensemble model has been devised. This model is the combination of the top three models with equal weights and with variant weights. The accuracy of the ensemble model thus designed is close to the accuracy of the best predictive model. Hence, it is termed as an optimal solution. The results show that the best model for predicting the DF is the Cubist model, followed by Random Forest and then Weka Lazy Classifier model. In the case of optimization, an ensemble with variant weights yield the best results. Figure 3.11 comparatively demonstrates the accuracy plot of all the methods applied for DF prediction:

TABLE 3.6: 10-fold cross validation for prediction of Data Flow

Runs	Cubist	Random Forest	WLM	CIT	BRNN	Ensm-EW	Ensm-VW
1	100	99.4	99.1	96.2	92.2	99.6	100
2	99.7	99.6	99	96	92.1	99.5	99.3
3	99.9	99.1	99.5	96.5	92.6	99.1	99.2
4	99.8	99.3	99.1	96.3	92	99.6	99.5
5	99.9	99.7	99.3	96.5	91.9	99.3	99.8
6	99.6	99.4	99.2	96.2	92.4	99.2	100
7	99.8	99.6	98.6	96.4	92.2	99.5	99.9
8	99.7	99.3	98.9	96.7	92.7	99.8	99.8
9	100	99.2	99.1	96.6	92.9	99.7	99.6
10	99.9	99.5	98.8	96.3	92.3	99.6	100

Ensm-EW = Ensemble Model with equal weights; Ensm-VW = Ensemble Model with variable weights.

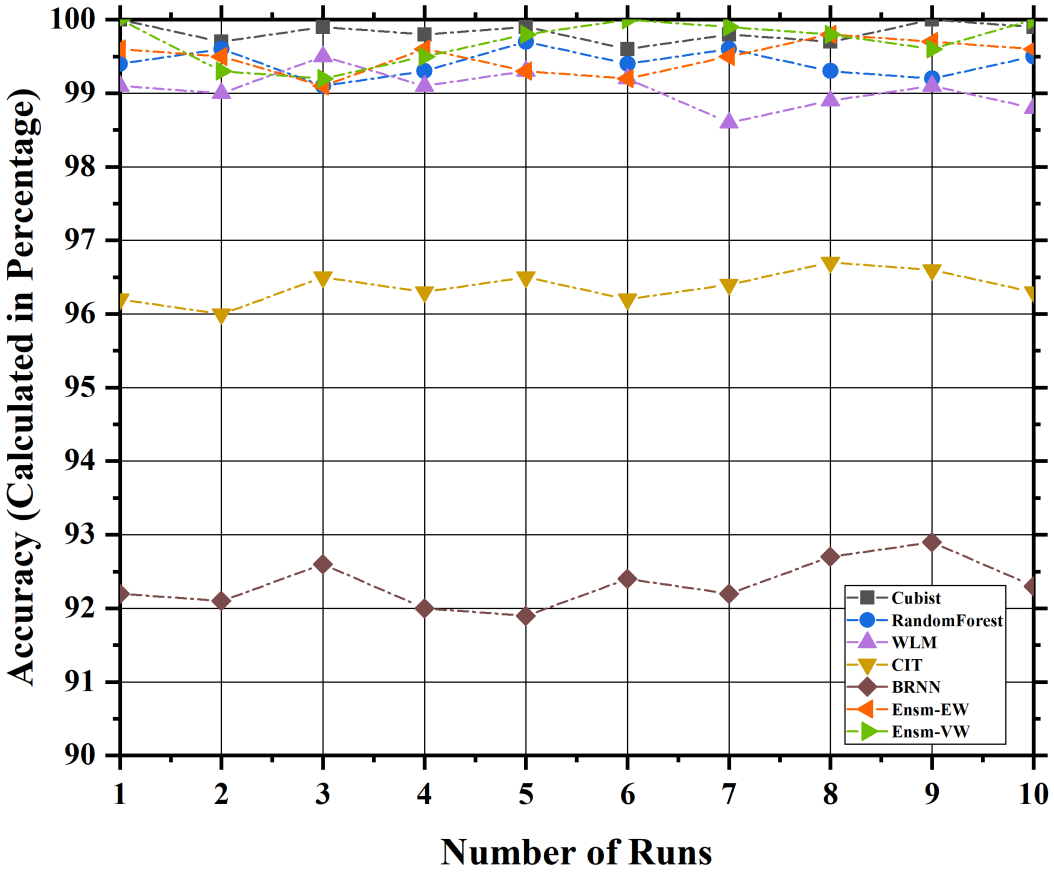


FIGURE 3.11: 10-fold validation for the prediction of Data Flow

3.4.1.1 Comparative Analysis of Data Flow

In this section, the results obtained are considered in the light of cross-validation [108]. The Table 3.7 shows the accuracy of applied models in case of single execution and cross-validation. These results validates the stability of the prediction models. The results of single execution are compared with cross-validation results. Also, the results show that accuracy after cross-validation is either equal or greater than the actual accuracy obtained in single execution. To support the results a box-plot for cross-validated results has been included as Figure 3.12. This shows the minimum, maximum and average value in the form of box. Hence, it illustrates that the actual results are not much varied from the cross-validated results.

TABLE 3.7: Comparative Result Analysis

Model Name	Without Cross-Validation	With Cross-Validation (Accuracy)			
	Accuracy	Average	Standard Deviation	Minimum Value	Maximum Value
Cubist	**100**	99.83	0.13	99.60	**100**
Random Forest	99.40	99.41	0.19	99.10	**99.7**
WLM	99.10	99.06	0.25	98.60	**99.5**
CIT	96.20	96.37	0.21	96.00	**96.7**
BRNN	92.20	92.33	0.32	91.90	**92.9**
Ensemble-EW	99.60	99.49	0.22	99.10	**99.8**
Ensemble-VW	99.90	99.71	0.30	99.20	**100**

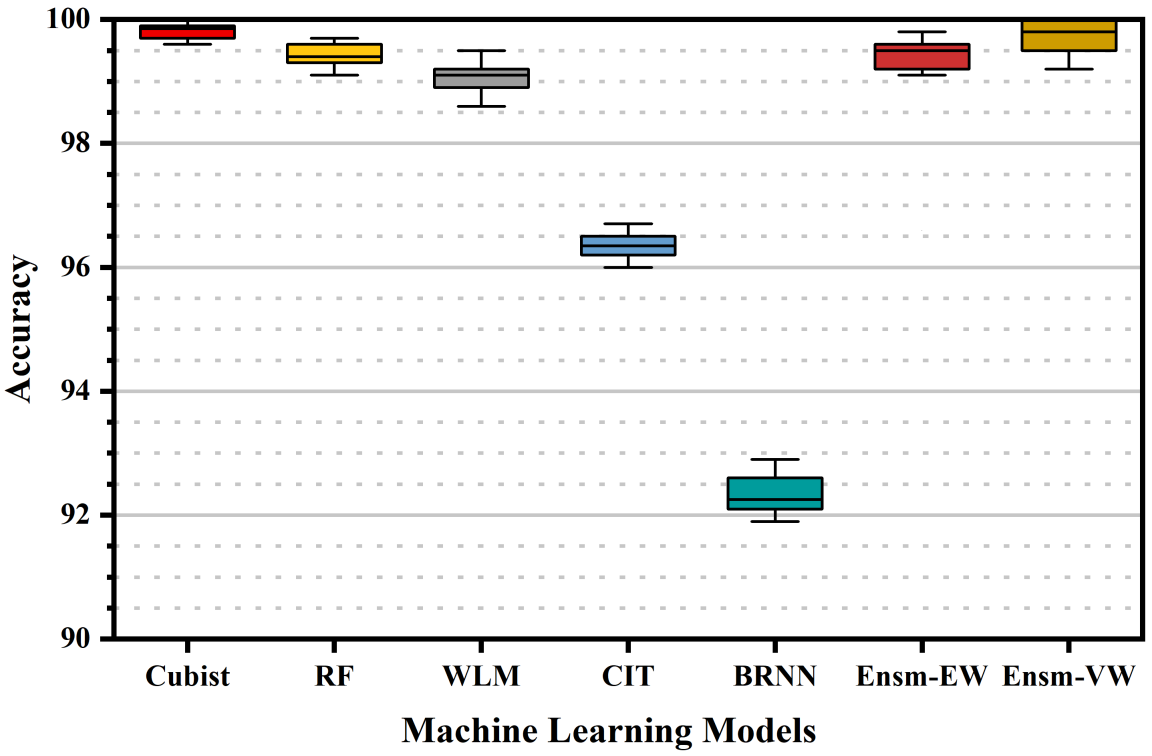


FIGURE 3.12: Box-Plot for Cross-Validation of Accuracy

Similarly, the same has been observed in the case of RMSE for Data Flow, not much deviation has been noticed [Refer to Figure 3.13].

3.4.2 Protocol Used

In this section, analysis has been carried out on the dataset based upon various Machine Learning models. These models operate upon various tuning parameters (refer to Table 4.5). All the prediction methods use data which is divided into training and testing data fixed at 70% and 30% respectively. An ensemble model has been prepared which yield

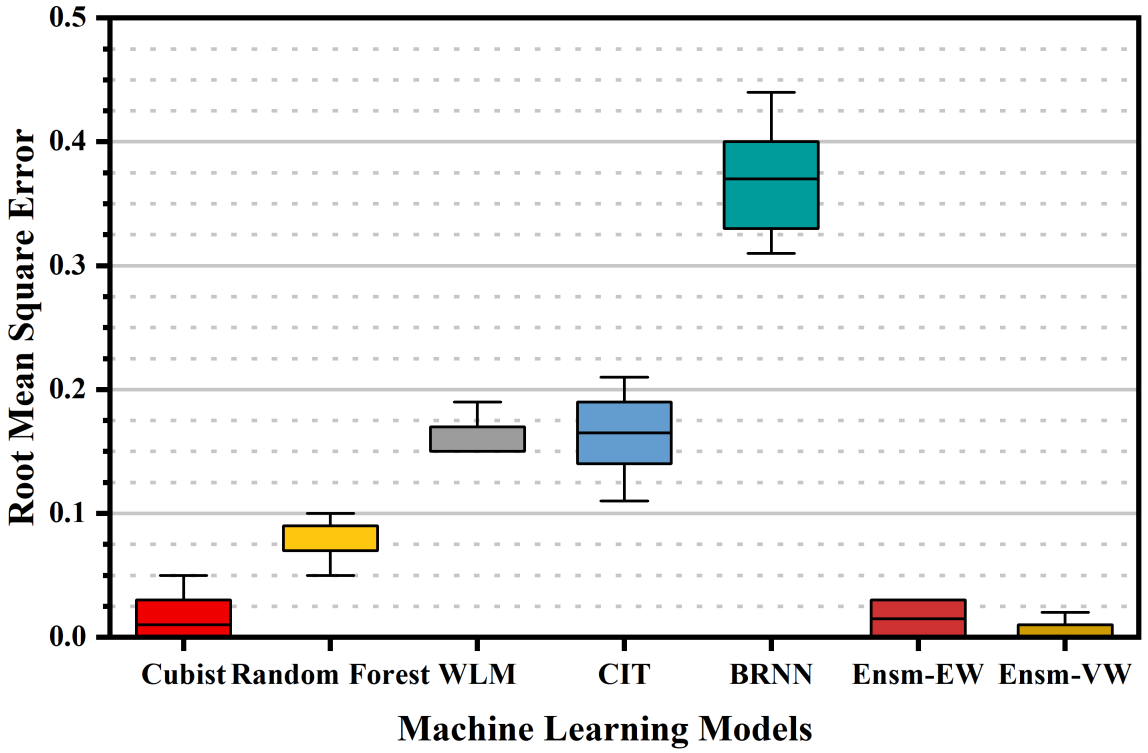


FIGURE 3.13: Box-Plot for Cross-Validation of RMSE

results comparative to the best predictive model. The results thus computed for a single execution are shown below:

TABLE 3.8: Performance comparison of Machine Learning Models for Protocol Used

SN	Model Name	H	Gini	AUC	AUCH	KS	MER	MWL	Spec	Sens	Acc
1	Random Forest	0.97	0.98	0.99	0.99	0.98	0.01	0.01	1.00	1.00	99.17
2	Bagged CART	0.95	0.98	0.99	0.99	0.97	0.02	0.01	1.00	1.00	98.83
3	Neural Network	0.94	0.98	0.99	0.99	0.97	0.06	0.02	0.96	0.99	98.67
4	Conditional Inference Tree	0.73	0.92	0.96	0.96	0.93	0.90	0.70	0.63	0.87	94.33
5	Tree Model From Genetic Algorithm	0.66	0.78	0.89	0.89	0.78	0.11	0.11	0.44	0.53	90.00
6	Ensemble Model	0.85	**0.92**	0.96	0.96	0.93	0.22	0.17	**0.81**	**0.88**	**99.33**

Sens = Sensitivity; Spec = Specificity; Acc = Accuracy.

The performance in terms of accuracy for the PN is shown in Table 3.9. The accuracy has been calculated for 10 consecutive runs of the same model. Also, to optimize the results,

an ensemble model has been devised. This model is the combination of the top three models. The accuracy thus obtained is close to the accuracy of the best predictive model. Hence, it is termed as an optimal solution. This is the problem of binary classification type.

TABLE 3.9: 10-fold cross validation for prediction of protocol used

Runs	Random Forest	BaggedCART	Neural Network	CIT	TMGA	Ensemble
1	99.2	98.8	98.7	94.3	90.0	99.3
2	99.3	98.2	98.4	94.1	90.4	99.7
3	99.0	98.6	98.7	94.6	90.2	99.4
4	99.1	98.9	98.1	94.9	90.7	99.9
5	99.4	98.4	98.8	94.3	90.1	99.5
6	99.5	98.7	98.2	94.2	90.6	99.6
7	99.7	98.1	98.6	94.7	90.4	99.3
8	99.2	98.8	98.9	94.8	90.0	99.7
9	99.3	98.5	98.8	94.1	90.3	99.1
10	99.6	98.3	98.2	94.5	90.1	99.7

The results show that the best model for predicting the PN is the Random Forest model, followed by Bagged CART and then the Neural Network model. In the case of optimization, ensemble model yield the best result. Figure 3.14 comparatively demonstrates the accuracy plot of all the methods applied to the PN, which is a binary class classification problem type in our work.

3.4.2.1 ROC Curve

When observing the plot, if the plot follows the straight line from the lower left to upper right, then the classifier cannot differentiate between negative and positive data. If the curve tends to bend to the upper left, then the model can differentiate the actual positive and negative data. On the contrary, if the curve tends to bend to the lower right, then it is just a completely wrong prediction model. Hence, the models used can clearly differentiate between the actual positive and negative data. There are certain overlapping lines showing for Random Forest, Neural Network, Ensemble Model. Whereas, Bagged CART, Conditional Inference Tree, Tree Model from Genetic Algorithm are represented as non-overlapping lines in Figure 3.15.

Also, there is another linked term AUC. The AUC score will be between 0 and 1 [109]. The higher the value of AUC, usually the better the model is.

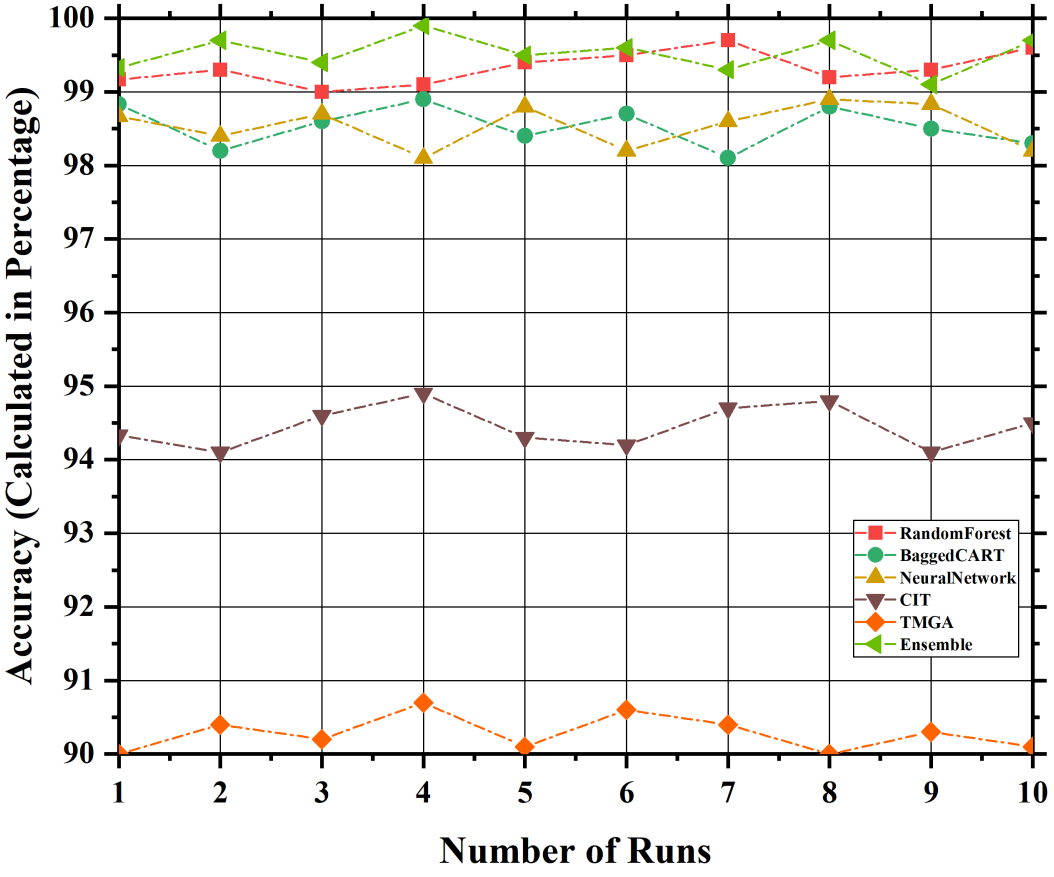


FIGURE 3.14: 10-fold validation for the prediction of protocol used

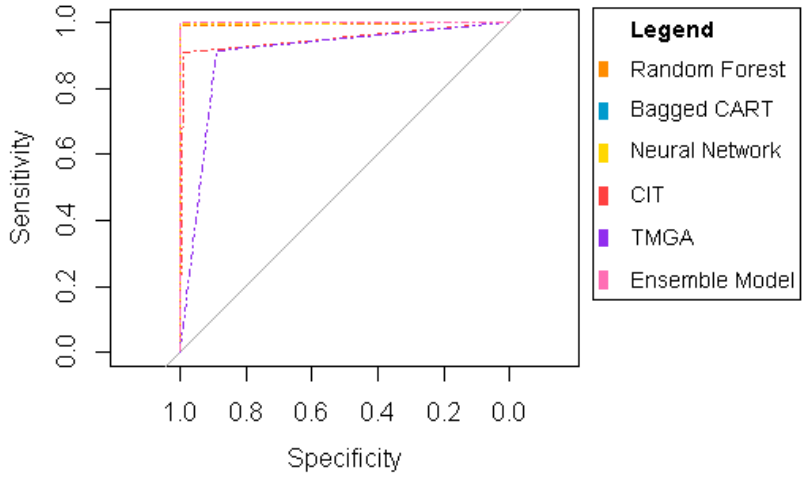


FIGURE 3.15: ROC for Protocol Used

3.4.2.2 Comparative Analysis of Protocol Used

In this section, the Machine Learning models are better evaluated using the cross-validation technique [108]. Table 3.10 shows the accuracy of applied models in the case of single execution and cross-validation. These results estimate the performance of the best predictive model. The results show that the maximum accuracy of prediction after cross-validation is greater than the actual accuracy obtained in a single execution.

TABLE 3.10: Comparative Result Analysis

Model Name	Without Cross-Validation	With Cross-Validation (Accuracy)			
	Accuracy	Average	Standard Deviation	Minimum Value	Maximum Value
Random Forest	99.17	99.33	0.22	99.00	**99.7**
Bagged CART	98.83	98.53	0.28	98.10	**98.9**
Neural Network	98.67	98.54	0.29	98.10	**98.9**
CIT	94.33	94.45	0.29	94.10	**94.9**
TMGA	90.00	90.28	0.24	90.00	**90.7**
Ensemble	99.33	99.52	0.24	99.10	**99.9**

To support the results a box-plot for cross-validated results has been included as Figure 3.16. This shows the minimum, maximum and average value in the form of a box. Hence, it illustrates that the actual results are not much varied from the cross-validated results.

Similarly, the cross-validation of AUC tells which of the used models predicts the classes in the best possible way [Refer to Figure 3.17]. This suggests that there is not much variation in the value of AUC when different dataset partitions are considered for validation. Further, the boxplot shows the minimum, maximum and average value of AUC in different prediction models.

Further, sensitivity or recall is the proportion of actual positive cases that are correctly identified. And, Specificity is the proportion of actual negative cases that are correctly identified. The preferred value of these is close to one. The cross-validation of these model evaluation parameters is shown in Figure 3.18 and Figure 3.19.

3.4.3 Number of Nodes

In this section, analysis has been carried out on the dataset based upon various Machine Learning models. These models operate upon various tuning parameters (refer to Table

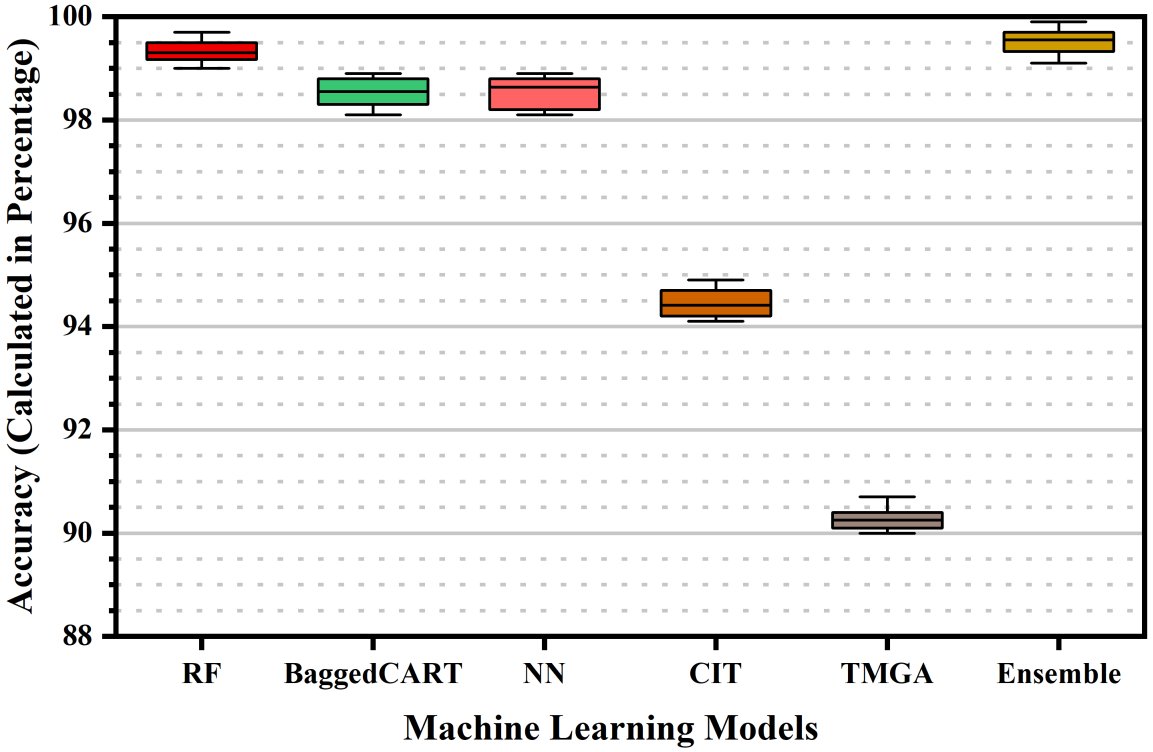


FIGURE 3.16: Box-Plot for Cross-Validation of Accuracy

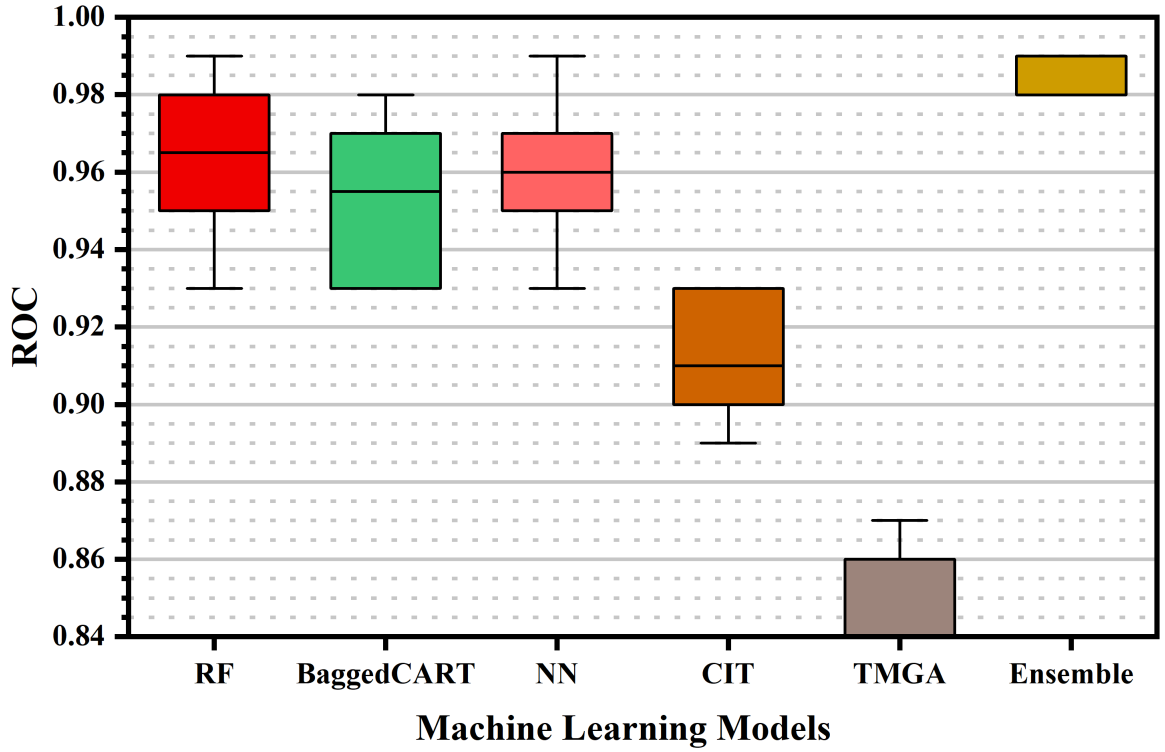


FIGURE 3.17: Box-Plot for Cross-Validation of ROC

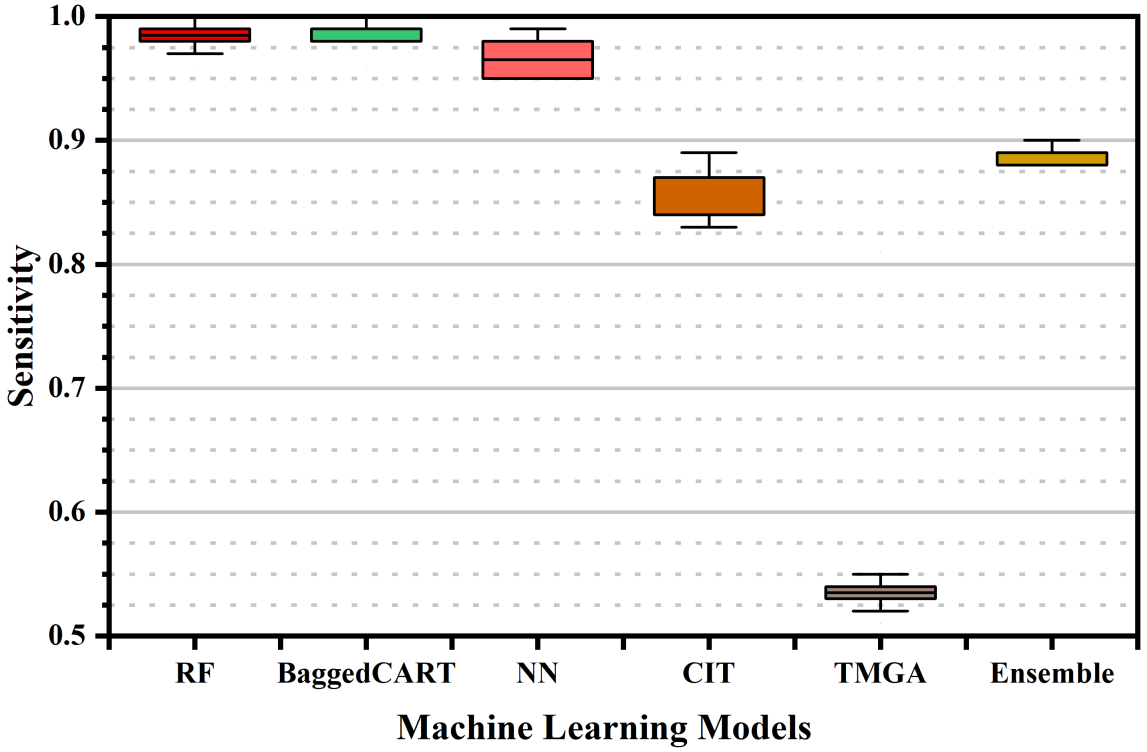


FIGURE 3.18: Box-Plot for Cross-Validation of Sensitivity

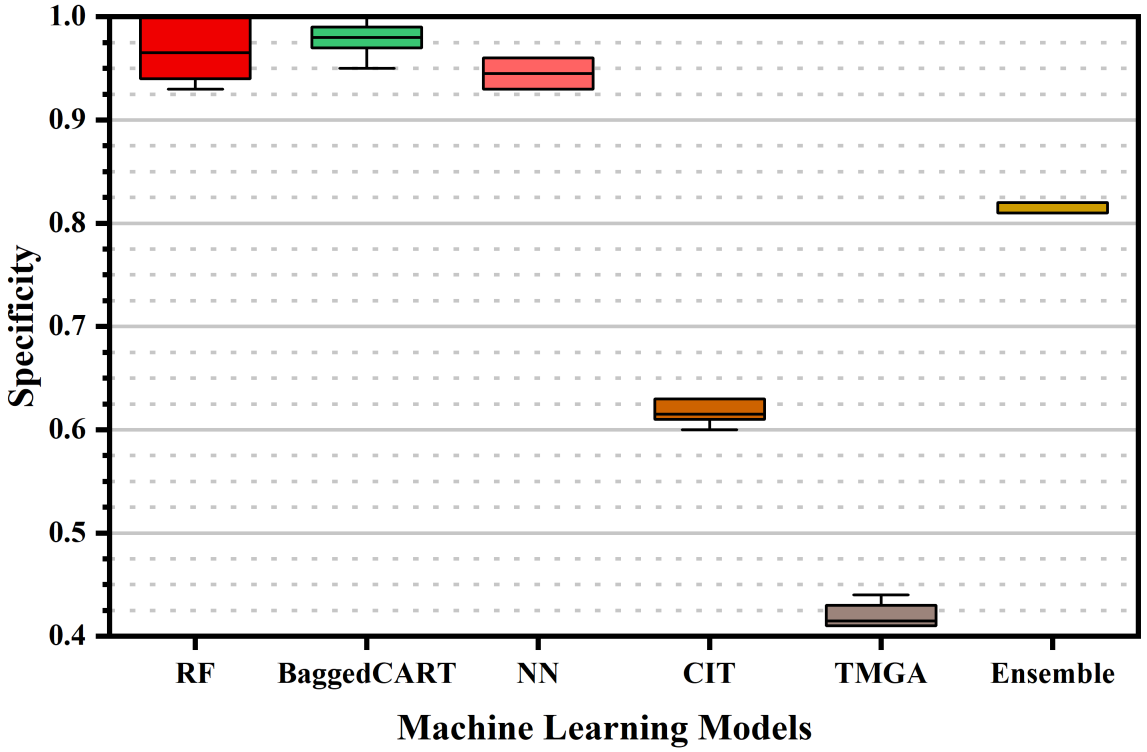


FIGURE 3.19: Box-Plot for Cross-Validation of Specificity

4.5). All the prediction methods use data that is divided into training and testing data fixed at 70% and 30% respectively. An ensemble model has been prepared which yield results comparable to the best predictive model. The results thus computed for a single execution are shown below:

TABLE 3.11: Performance comparison of Machine Learning Models for Number of Nodes

SN	Model Name	Accuracy
1	SVM	98.50
2	Conditional Inference Tree	97.50
3	Random Forest	95.17
4	Bagged MARS	67.67
5	Tree Model From Genetic Algorithm	65.17
6	Ensemble	**97.60**

The performance in terms of accuracy for the NN is shown in Table 3.12. The accuracy has been calculated for 10 consecutive runs of the same model. Also, to optimize the results, an ensemble model has been devised. This model is a combination of the top three models. The accuracy thus obtained is close to the accuracy of the best predictive model. Hence, it is termed as an optimal solution. This is a multi-class classification problem type.

TABLE 3.12: 10-fold Cross validation for prediction of number of nodes

Runs	SVM	CIT	Random Forest	Bagged MARS	TMGA	Ensemble
1	98.5	97.5	95.2	67.7	65.2	97.6
2	98.9	97.2	95.5	67.9	65.4	97.9
3	98.6	97.4	95.7	67.6	65.5	97.5
4	98.3	97.6	95.3	67.8	65.7	97.2
5	98.7	97.9	95.6	67.5	65.4	97.4
6	98.2	97.5	95.2	67.7	65.9	97.6
7	98.5	97.3	95.3	67.4	65.6	97.9
8	98.9	97.7	95.5	67.3	65.4	97.3
9	98.4	97.4	95.8	67.5	65.2	97.7
10	98.8	97.8	95.9	67.8	65.3	97.4

The results show that the best model for predicting the NN is the Support Vector Machine (SVM), followed by the Conditional Inference Tree Model and then the Random Forest model. In the case of optimization, an ensemble model is being used. Figure 3.20 comparatively demonstrates the accuracy plot of all the prediction methods applied to the NN, which is a multi-class classification problem type.

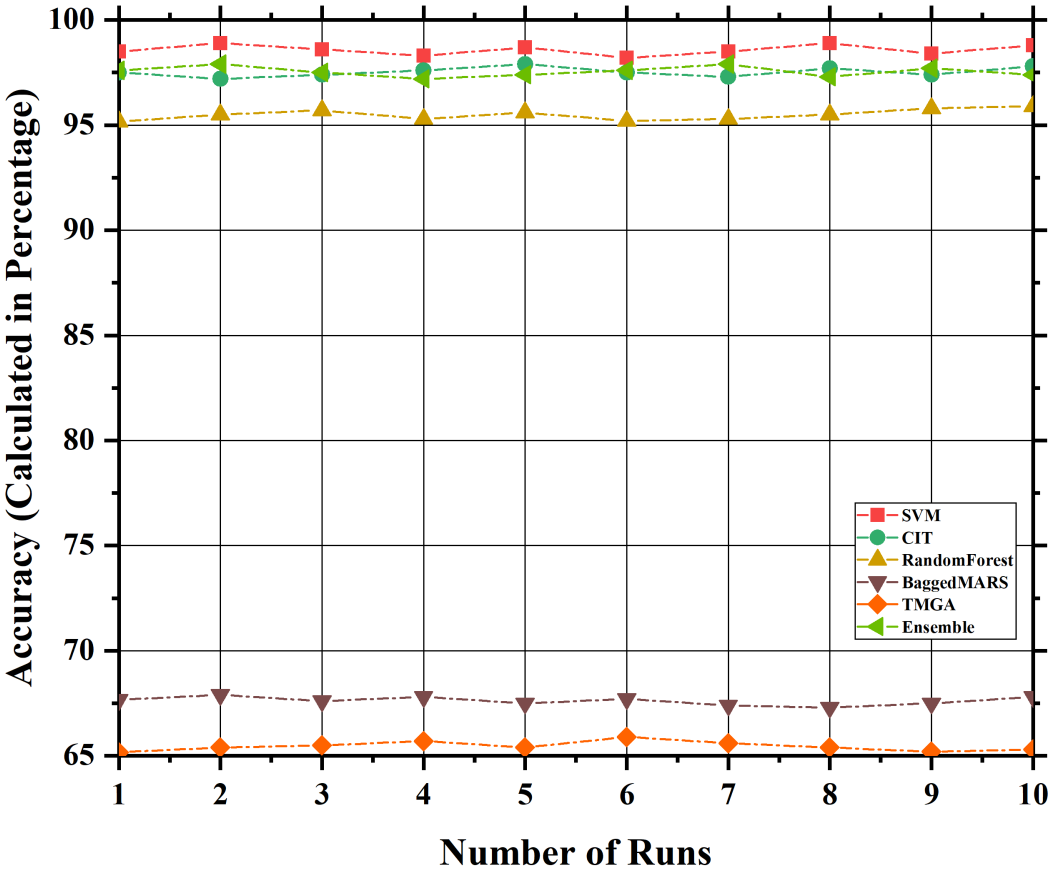


FIGURE 3.20: 10-fold validation for the prediction of number of nodes

3.4.3.1 Comparative Analysis of Number of Nodes

In this section, the prediction models are better evaluated in terms of accuracy using the cross-validation technique [108]. Table 3.13 shows the accuracy of applied models in the case of single execution and cross-validation.

In this 10-fold cross-validation, the cross-validation procedure is repeated ten times, yielding many random partitions of the original sample. These results are again averaged (or otherwise combined) to produce a single estimation. Further, these results show that the maximum accuracy of prediction after cross-validation is greater than the actual accuracy obtained in a single execution.

To support the results a box-plot for cross-validated results has been included as Figure 3.21. This shows the minimum, maximum and average value in the form of a box. Hence, illustrates that the actual results are not much varied from the cross-validated results, as depicted by the standard deviation.

TABLE 3.13: Comparative Result Analysis

Model Name	Without Cross-Validation	With Cross-Validation (Accuracy)			
	Accuracy	Average	Standard Deviation	Minimum Value	Maximum Value
SVM	98.50	98.58	0.24	98.20	**98.9**
CIT	97.50	97.53	0.22	97.20	**97.9**
Random Forest	95.17	95.50	0.25	95.17	**95.9**
Bagged MARS	67.67	67.62	0.19	67.30	**67.9**
TMGA	65.17	65.46	0.23	65.17	**65.9**
Ensemble	97.60	97.55	0.24	97.20	**97.9**

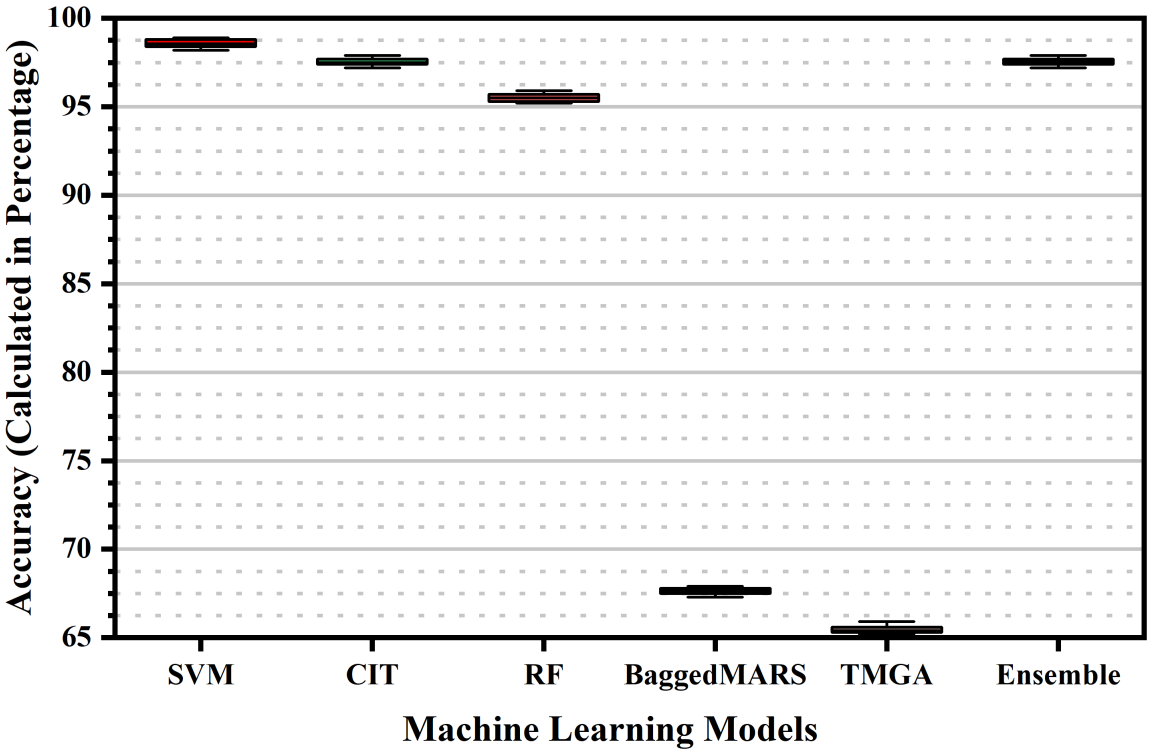


FIGURE 3.21: Box-Plot for Cross-Validation of Accuracy

3.5 Summary

In this chapter, a novel reliability framework that includes various phases of implementation starting with scenario generation using a discrete-event simulator, dataset creation, applying Machine Learning techniques and finally the result analysis is presented. Here, Machine Learning methods do not include any additional information from other models or alternative network information. All the models are evaluated on

RMSE, correlation, R^2, AUC, Gini and accuracy. Ten Machine Learning models are used to conclude that an ensemble model yields optimum results in predicting data flow, the protocol used, and the number of nodes. The ensemble model predicts with an accuracy of 99.9% for data flow, 99.9% for the protocol used and 97.6% for the number of nodes. Finally, the 10-fold cross-validation procedure has been used to check the robustness of an ensemble model. The future work includes a study of the effect of various other routing protocols and the addition of more performance metrics to produce even better results. This work focusses on small-scale networks but we believe that this can be extended to large-scale networks. The dataset used in this work is available as a supplement at *http://bit.ly/SSWSN-Reliability*.

Chapter 4

Dependability Enhancement under Flooding Attack: A Machine Learning Perspective

"The more original a discovery, the more obvious it seems afterwards."

-Arthur Koestler

The Wireless Sensor Network(WSN) hold the promise of expediting the process of data processing in large network deployed for real-time applications[1] . The assorted applications of this network are based upon the sub-tasks described in Equation 4.1:-

$$Multitude\,of\,Applications \sim max(f(R_C, P_{CPU}, S_T, A_T)) \tag{4.1}$$

In this equation, a multitude of applications of WSN is the function of *f(x)* where $x \; \varepsilon \; \{R_C, P_{CPU}, S_T, A_T\}$. R_C denotes the radio communication; P_{CPU} is the processing task carried out by the CPU, S_T is sensing task and A_T is the actuation (controlling) task. These sub-tasks should be maximized to attain a dependable WSN [110]. The wireless

[1]The contents of the chapter are peer-reviewed and published in:
Jasminder Kaur Sandhu, Anil Kumar Verma, Prashant Singh Rana, "Enhancing dependability of wireless sensor network under flooding attack: A Machine Learning perspective", International Journal of Ad-Hoc and Ubiquitous Computing, SCI-Indexed, Impact Factor 0.705

communication between sensor nodes paves a way for an attack by the adversary. Therefore, an effective mechanism is required for secure communication of data packets from the source to destination node [111, 112].

When the sensor nodes are densely deployed, there is a greater threat of flooding attack [113, 114]. Flooding attack [110] is a type of denial-of-service attack [115, 116, 117] which makes network unavailable by flooding it with voluminous traffic [118]. During data transmission, every node has a buffer. Once this buffer is full, any further connection request cannot be handled. The future attempt to transmit packets by such node leads to packet drop. When this packet drop crosses a certain threshold limit, it is interpreted as the occurrence of a flooding attack. This attack can decrease the throughput of the network by 84 percent [119]. This attack also leads to excess bandwidth consumption and service availability issues [120, 121, 122]. The mitigation of flooding attack will lead to increased availability and will further enhance the dependability of that network [123, 124].

In this chapter, a Rule-Based Intrusion Detection System (IDS) is proposed using the colloquial watchdog approach with an appended buffer to detect a flooding attack. This approach is based on the trust factor [125, 126]. The mitigation mechanism includes a novel normalized approach based on the NORMDIS scheme [127] in which the node coordinates are optimized taking into consideration the mathematical distribution and transmission ranges of various nodes. The results are analyzed for 10 to 50 nodes in a network.

Further, the dependability of a network can be determined based on the basic performance parameters such as network lifetime, energy efficiency, throughput, Packet Delivery Ratio (PDR), Data Flow (DF). This work considers DF and PDR as an important dependability enhancement features. The contribution of this work is three-fold: i) to propose an IDS for detecting flooding effect on the network, ii) to optimize the dependability of the network and iii) to implement and perform prediction of DF which in turn reduces the impact of flooding on the network. The novelty of this work is the application of Machine Learning models for understanding the behavior of WSN and thus opens new avenues for the research fraternity.

4.1 Assumptions and Graph Model

4.1.1 System Assumptions

The WSN contains multiple sensor nodes deployed in a specific area to be monitored [128]. Assuming that this deployment of nodes is made at any random location based on transmission range, to allow the best possible coverage. This transmission range based communication scenario helps in the creation of an un-normalized dataset, where nodes are deployed in a random fashion. It is based on the observation that the node which wants to communicate with another node must lie in the threshold transmission range of that communicating node.

Let A be the area in which N nodes are to be deployed. So, the node density is defined as $D = N/A$ [129]. The Poisson distribution can be used to determine the probability that there are m nodes deployed in the area, as given by Equation 4.2 [130]: –

$$P(m) = \frac{(DR)^m}{m!} e^{-DR} \tag{4.2}$$

4.1.2 Graph Model

A sensor network is described by an optimal tuple $G = (V, E, F_V, F_E)$ where V is the number of nodes, E represents the communication links $(E = V \times V)$. F_V represents a set of functions used to portray properties of each node V (such as energy reserve, transmission range). F_E stipulates properties for each communication link (such as link capacity). In this work, scenarios are constructed with the number of nodes denoted by $i = [10, 20, 30, 40, 50]$ and node $i \varepsilon N$ is stationary, if and only if, its physical deployment does not change with time. A network is said to be stationary if all the nodes are static. The range assignment of this topology is denoted by the function:

$$RA : N \rightarrow (0, r_{max}] \tag{4.3}$$

The equation 4.3 represents the transmission range, where r_{max} depicts the maximum transmission range of nodes in the generated scenario. This range is dependent on the features or more commonly called the specification of the radio transceivers equipping the nodes. Assuming that all the transceivers possess the same features; hence, r_{max} has a single value for all nodes in the network.

According to Rappaport [2002] power P_i required by the node i to correctly transmit data to node j must satisfy the inequality in Equation 4.4:-

$$\frac{P_i}{\delta_{i,j}^{\alpha}} \geq \beta \tag{4.4}$$

where $\alpha \geq 2$ is the distance power gradient, $\beta \geq 1$ is the transmission quality, and $\delta_{i,j}$ is the Euclidean distance between the nodes. While the value of β is usually set to 1, the value of α depends on environmental conditions. In the ideal case, the value of $\alpha = 2$; however, α is typically 4 in various realistic situations. A value of α is defined between $[2,6]$ and is commonly accepted. This formula holds true for free space environments with no obstruction in the line of sight and does not consider possible occurrence of reflection, diffraction and scattering caused by building, terrain, etc. The transmission range concept is illustrated in Figure 4.1:

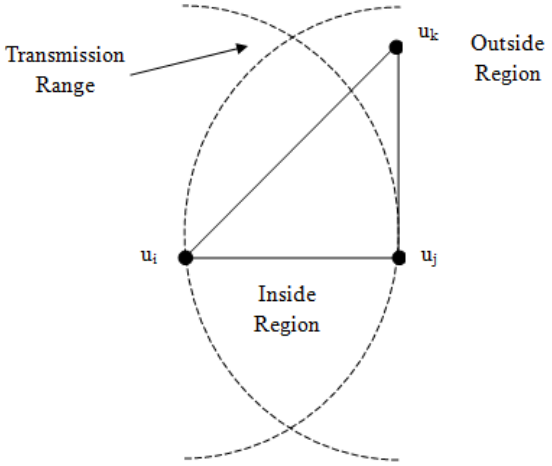


FIGURE 4.1: Transmission Range

In free space, the radio signals propagate along a straight line path which exists between the sender and receiver known as the Line-Of-Sight (LOS). In these transmission scenario, free space loss is experienced even if, no matter exists between sender and receiver which is given by the inverse square law represented in Equation 4.5: –

$$ReceivedPower(P_r) \propto \frac{1}{d^2} \tag{4.5}$$

where, d represents the distance between sender and receiver. The model of Rodoplu and Meng suggests that the power needed for transmission and reception of a signal is represented in Equation 4.6: –

$$sP = u(d) = ad^{\alpha} + bd + c \tag{4.6}$$

where, a depends on the physical environment such as unit size of a signal, d is the distance between two nodes, α represents the signal attention and is adjusted depending on the model used and c is the energy consumption. Also, α = 2 for the free space and α = 4 for the urban environment.

4.2 Detection of the Flooding Attack

4.2.1 Intrusion Detection System

The Intrusion Detection System checks behavior of network passively to find out which node is working abnormally [131, 132**?**]. This unit is installed on the sink node (also known as the base station) or the deployed sensor nodes or on both the ends. It has a life cycle of three phases as illustrated in Figure 4.2:

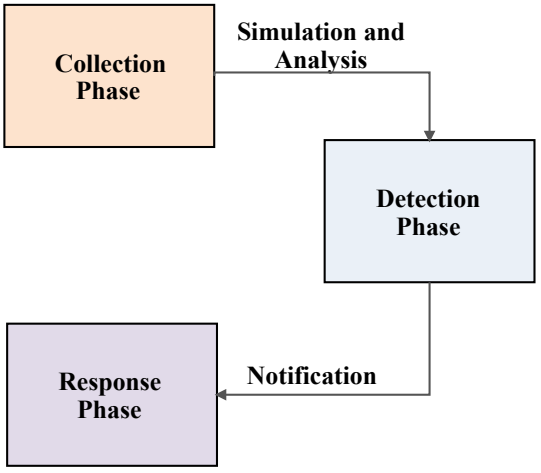


FIGURE 4.2: General Structure of IDS

The network data is gathered in the collection phase. This module is responsible for gathering the data about local events as well as the neighboring events. It monitors traffic patterns and resource usage. The data is simulated using a network simulator (NS-2.35) and analyzed based on performance parameters [Refer to Table 3]. This forms the un-normalized dataset. The detection phase finds intrusion by applying a suitable detection policy. This phase detects the abnormal functioning of the network in terms of packet drop and packet delivery ratio. The patterns monitored in the collection phase are further analyzed for normal or malicious behavior. Any deviation from the normal functioning of a network is termed as an attack. A notification is then sent to the response phase. In case of such unexpected activities, an alarm is generated by the response phase.

4.2.2 Classification of Intrusion Detection System

The detailed analysis of literature classifies the IDS [133, 134] in three categories, as illustrated below:

1. **Rule-Based IDS:** The rules depending upon specific attacks are articulated in Rule-Based IDS model. So, sub-phases for these systems include data acquisition, rule implementation, and intrusion detection.

2. **Cluster-Based IDS:** This approach improves the security of clusters for the sensor network. It primarily follows two approaches: a model based on authentication to resist external attacks; and model based on an energy-saving mechanism which focuses on misbehavior, both in the member nodes and the cluster-head nodes.

3. **Hybrid IDS:** Both the cluster-based and rule-based IDS combines to form hybrid IDS. The main goal is to achieve high security and lower energy consumption.

Table 4.1 illustrates the categories of IDS existing in the literature and also, the particular type of attack mitigated by them:

TABLE 4.1: Comparison of IDS Models

IDS Model	Network Architecture	Handled Attacks	Energy Consumption	Advantage	Disadvantage
Rule-Based	Distributed	DoS attacks, Sinkhole, Flooding, Blackhole, Selective forwarding.	Low	Detects all those attacks having specific rules, signatures.	Cannot detect new attacks.
Cluster-based IDS	Hierarchical	Misdirection attack.	Low	Guaranteed data-delivery.	Increased traffic.
Hybrid IDS	Hierarchical	Selective forwarding, Sinkhole, Hello-flood and Wormhole attacks.	Medium	Can detect both existing and new attacks.	Requires more computation and resources.

4.2.3 Proposed Rule Based IDS for Detection of Flooding Attack

In this section, an IDS is proposed based on a set of rules for monitoring the effect of flooding attack. The DF governs the influence of a flooding effect on WSN. So, to predict an optimum DF, multiple Machine Learning models are used.

4.2.3.1 Assumptions of Proposed System and Rules Applied

The main rationale behind the proposed system is based on the nature of the communication in network[135, 136], the deployment strategy of sensor nodes[137, 138, 139], and the intrusion detection scheme used to detect an attack in the network [140, 141]. The proposed system is constructed using Rule-Based IDS with the following assumptions:

1. The wireless links are of broadcasting nature.
2. The sensors are densely deployed randomly.
3. Each sensor has its own watchdog to monitor its one-hop neighboring nodes behavior.

The proposed IDS is constructed using the following rules:

1. The sender node transmits a packet and checks whether the intermediate node has the capacity to forward the packet or not. If the intermediate node drops the packet,

the failure count is increased. And, when this failure count exceeds a pre-defined threshold limit, an alarm is raised to indicate the occurrence of an attack.

2. If the trust level or the confidence level exceeds certain tolerated limits, the normalization technique is applied to optimize the network scenario.

4.2.3.2 Proposed Intrusion Detection System for Detection of Flooding Attack

The proposed IDS is divided into four components as demonstrated in Figure 4.3:

- Native reply module
- Track-down unit (cooperative and native)
- Interaction module
- Native packet monitoring

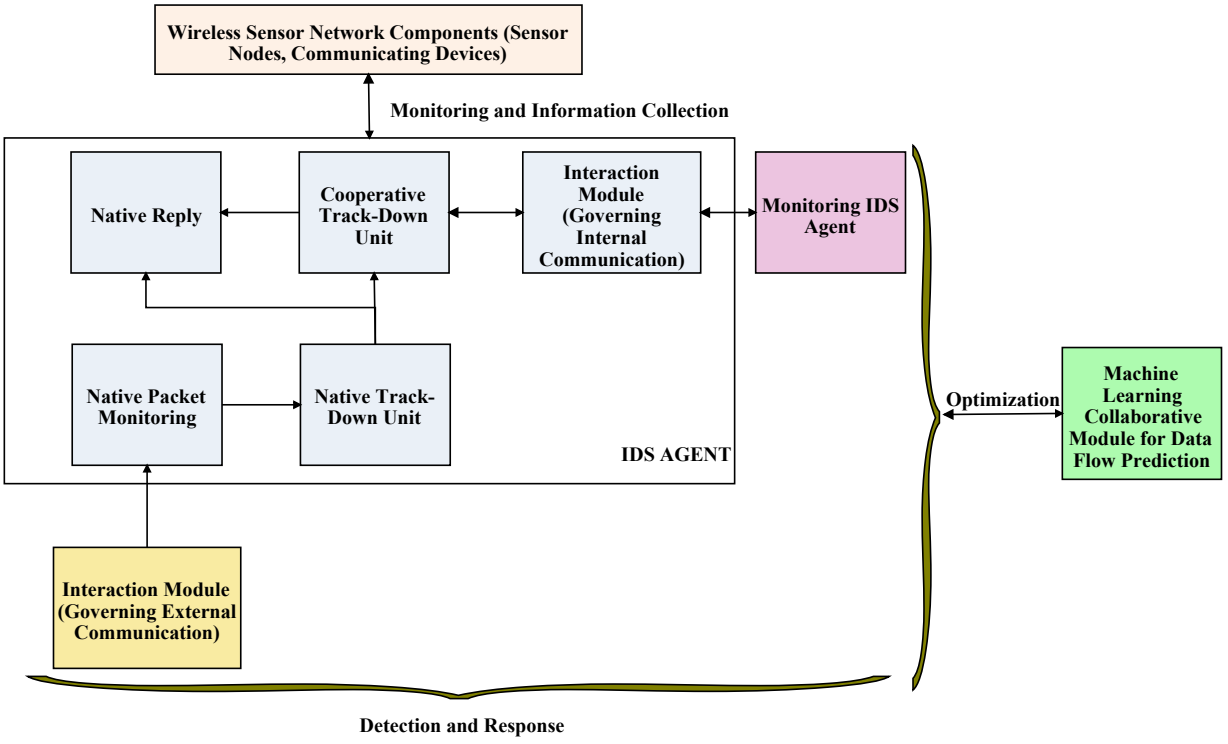


FIGURE 4.3: Proposed Structure of IDS

The native reply module transmits a message to the sink node when any malicious activity (such as packet drop) is detected. The track-down unit shares the information

with other neighboring nodes when an intrusion is detected. The native packet monitoring unit observes the packets locally and also sends it to a track-down unit for the detection of an anomaly. The monitoring IDS agent uses local information of the next-hop node and overhears it. If the sending time of the packet exceeds a predefined threshold, then the node is marked as malicious. And in this way, the watchdog approach detects the malicious node in a network. Moreover, the proposed IDS is capable of detecting packet dropping attack too.

4.2.4 Monitoring IDS Agent

It is a monitoring technique that detects misbehaving nodes in the network and is based on the trust mechanism. This is also termed as the watchdog mechanism. To understand the concept refer to Figure 4.4:-

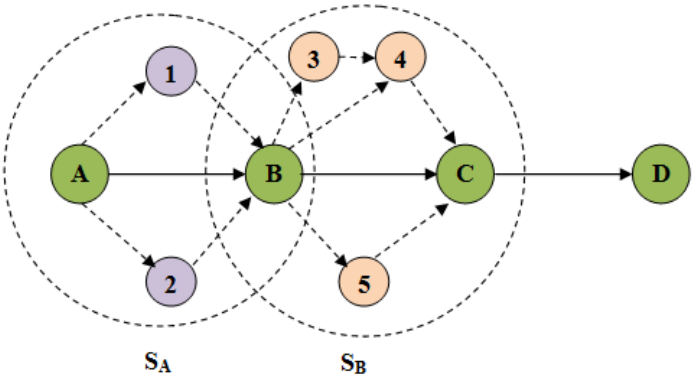


FIGURE 4.4: Monitoring IDS Agent Example Scenario

In this figure, S_A represents the set of nodes hearing messages from A to B and S_B represents the set of nodes hearing messages from B to C. The intersection region which is $S_A \cap S_B$ is much prone to intrusion. The IDS agent is activated for packet monitoring and in case an intrusion is detected, an alarm is raised.

An additional capability of buffering is added to further improve this approach. The agent saves the packet in monitoring buffer before transmitting it to monitor packets relaying from a neighboring node to the next node. In Figure 4.4, A is the sender, C is the receiver and B is an intermediate node. When node A sends a packet to node B, the watchdog in node A verifies whether node B forwards the packet towards node C using sensors overhearing ability within the transceiver range. The node A stores all recently

sent packets in its buffer. Each packet is now compared to the overheard packet. If a match exists, the successful transmission has taken place between node A and node C and thus, node A removes the packet from the buffer. Else, if the transmission is unsuccessful, a packet remains in the buffer for a longer period. This elongated time period is compared with a pre-determined time. If the elongated time exceeds a pre-determined time, failure count is increased or the confidence level/ trust level decreases. Now, this failure count when exceeds a certain threshold limit set by the user, node B is considered as a misbehaving node by the sender node A.

For trust measurement, the Beta trust model [142] is used which tells the trustworthiness of a sensor node. Trust value T is computed as:-

$$T = \frac{s+1}{s+f+2} \tag{4.7}$$

Now, s is the number of times a node forwards the packet and f represents the number of times a node drops the packet. For further reading, [143] is recommended. As an add-on to this research work, the DF prediction using Machine Learning approaches is being carried out in the following sections to decrease the possibility of packet drop, by transmitting the packets at an optimum data flow rate.

4.3 Proposed Workflow Description

This section makes use of the un-normalized and normalized dataset. The un-normalized dataset contains the network performance parameters when the nodes of the network are randomly deployed. Whereas, the normalized dataset considers performance parameters after applying the NORMDIS technique [127] of node deployment. The proposed scheme is described in three phases. The first phase evaluates the un-normalized dataset and the second phase evaluates the normalized dataset. The comparison of both the phases yields certain key findings and recommendation which is useful while deploying WSN for futuristic applications. This task is performed in the third phase. Also, the Machine Learning model predicts DF for WSN. The work suggests the best-suited model for the normalized and un-normalized dataset. A generic workflow is presented in Figure 4.5:

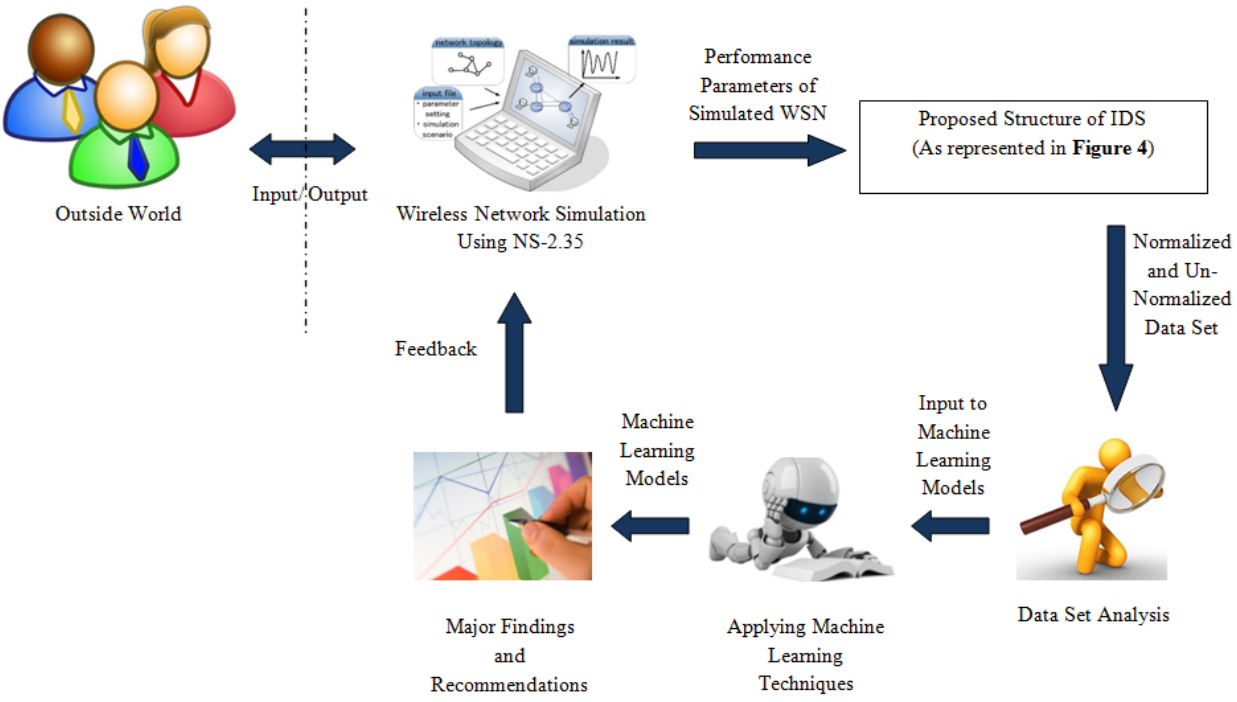


FIGURE 4.5: Generic Workflow

The first phase deals with the random deployment of sensor nodes. The performance parameters computed for this deployment are analyzed for PDR, which is an important metric for dependability evaluation. Similarly, the same operation is performed in the second phase on normalized coordinates. The coordinates are normalized using the NORMDIS technique. Both the results are compared, to justify which network is more dependable. In the last phase, it is determined which Machine Learning model fits best for prediction of the DF performance parameter (which cannot be formulated), in the case of normalized and un-normalized data. This study also limits the effect of flooding attack on futuristic networks by regulating the DF parameter.

The detailed workflow is demonstrated in Figure 4.6. In a nutshell, the Phase-I works for un-normalized data, Phase-II for normalized data and Phase-III includes application of Machine Learning models and results interpretation. The details of the basic terminologies can be referred from Annexure-C.

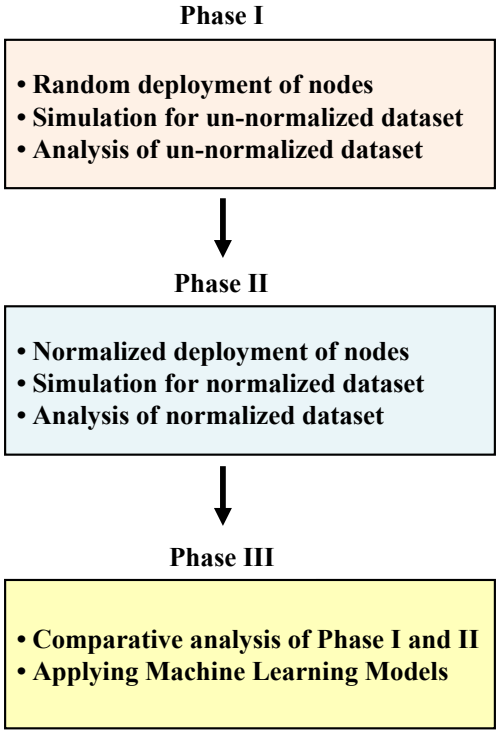


FIGURE 4.6: Detailed Workflow

4.3.1 Simulation Modeling and Dataset Description

The wireless scenarios are constructed using the NS-2.35 simulator. The details of network simulation using NS-2.35 can be referred from Annexure-A. The simulation is carried out for 20 seconds with a warm-up time duration of 4 seconds. The nodes communicate wirelessly using routing protocols AODV and DSR [144, 145, 146]. AODV stands for Ad-hoc On-demand Distance Vector protocol and DSR stands for Dynamic Source Routing protocol. These are reactive protocols that create route on-demand [147, 148]. The major advantage of the reactive (on-demand) approach is that the route is established only when it is required and hence the need to find routes to all other nodes in the network as required with the table-driven approach is eliminated [149]. Both these protocols are suitable for multicasting and unicasting [150]. The simulation setup is being constructed for a maximum of 50 nodes and can be scaled to increase the number of nodes. Initially, the sensor nodes are deployed randomly and communication takes place between the sender and receiver. The simulation parameters required for scenario generation are shown in Table 4.2:

TABLE 4.2: NS-2.35 Simulation Parameters

SN	Parameter	Value
1	Channel Type	Wireless
2	Radio-Propagation Model	Two-Ray Ground
3	Interface Queue Type	DropTail/CMUPriQueue
4	Antenna Model	Omni-directional
5	Max Packet in Interface Queue	150
6	Number of Nodes	10-50
7	Data Flow	0.1-10 Mb
8	Routing Protocols	AODV/DSR
9	X Dimension of Topography	1000 m
10	Y Dimension of Topography	1000 m
11	Simulation Time	20 seconds

This work considers only those performance parameters [151, 152] that are vital in improving the dependability of the network. So, the prediction is carried out for both random deployment and normalized deployment. The performance parameters considered for this work are defined in Table 4.3:

TABLE 4.3: Description of the dataset

SN	Feature	Information
1	SP	Packets Sent
2	RP	Received Packets
3	PF	Packets Dropped
4	PDR	Packet Delivery Ratio (Computed in Percentage)
5	PN	Protocol Name (AODV or DSR)
6	DF	Data Flow
7	NN	Number of Nodes

The dataset used in this work is available as a supplement at $http://bit.ly/JKSPSRAKV$. Table 4.4 shows the snapshot of the dataset generated using the NS-2.35 simulations. This table presents the data in a shuffled form. This is obtained using the random deployment of nodes and is known as un-normalized (un-structured) dataset.

Further, the data obtained using the normal distribution of nodes is termed as the normalized (structured) dataset. This means the normalized dataset is obtained after applying the NORMDIS technique to the un-normalized dataset. The dataset has been constructed by taking an average of 10 consecutive executions.

TABLE 4.4: Sample Dataset

Un-normalized dataset							Normalized dataset						
SP	RP	DP	PDR	PN	DF	NN	SP	RP	DP	PDR	PN	DF	NN
501	500	1	99.80	0	0.2	10	251	250	1	99.60	0	0.1	10
750	667	83	88.93	0	0.3	10	501	500	1	99.80	0	0.2	10
1001	667	334	66.63	0	0.4	10	750	745	5	99.33	0	0.3	10
1251	666	585	53.23	0	0.5	20	1001	973	28	97.20	0	0.4	10
1501	666	835	44.37	0	0.6	20	1251	1200	51	95.92	0	0.5	10
1751	666	1085	38.03	0	0.7	20	1501	1428	73	95.13	0	0.6	10
2000	666	1334	33.3	0	0.8	20	1751	1655	96	94.51	0	0.7	10
251	249	2	99.20	1	0.1	10	2000	1824	176	91.2	0	0.8	10
501	498	3	99.40	1	0.2	10	750	750	0	100	0	0.3	20
750	748	2	99.73	1	0.3	10	1001	997	4	99.60	0	0.4	20
1001	992	9	99.10	1	0.4	10	1251	995	256	79.53	0	0.5	20
1251	991	260	79.21	1	0.5	10	1501	995	506	66.28	0	0.6	20
251	250	1	99.60	1	0.1	30	1751	995	756	56.82	0	0.7	20
501	500	1	99.80	1	0.2	30	2000	995	1005	49.75	0	0.8	20
750	480	270	64	1	0.3	30	501	500	1	99.80	0	0.2	40
1001	487	514	48.65	1	0.4	30	1001	907	94	90.60	0	0.4	40
1251	400	851	31.97	1	0.5	30	251	250	1	99.60	1	0.1	20
1501	492	1009	32.77	1	0.6	30	501	500	1	99.80	1	0.2	20
1751	474	1277	27.07	1	0.7	50	1001	1000	1	99.90	1	0.4	20
2000	484	1516	24.2	1	0.8	50	1251	1250	1	99.92	1	0.5	20
2250	501	1749	22.26	1	0.9	50	1001	1000	1	99.90	1	0.4	50
2501	395	2106	15.79	1	1	50	1501	1500	1	99.93	1	0.6	50
2750	484	2266	17.6	1	1.1	50	2000	2000	0	100	1	0.8	50

4.3.2 Machine Learning Techniques

The Machine Learning models [153, 154] used in the course of this work for predicting the network parameter DF are described in Table 4.5). The details of R programming used for implementing Machine Learning models can be referred from Annexure-B.

1. Linear Model (M1): The dependent variable outcome is computed with the help of an independent variable.
2. Decision Tree (M2): It involves multiple splits until a termination condition is encountered. The nodes of the tree depict events and edges depict decision rules.
3. Extreme Learning Machine (M3): It has a hidden layer and faster training mechanism.
4. Tree Models from Genetic Algorithms (M4): Learning by the regression trees using evolutionary techniques.

5. Generalized Additive Model (M5): The linear form of predictor variable depends on smooth functions. It smoothens each individual predictor variable.

6. Model Tree (M6): It is used for regression problems. Binary recursive partitioning is used for splitting.

7. Projection Pursuit Regression (M7): It is an extension of the additive models. Univariate regression is used in this case.

8. Bayesian Regularized Neural Networks (M8): These are widely used in prediction problems of various application areas and, more recently, for genome-enabled prediction.

9. PartyKit (M9): Embeds the regression models following a tree structure.

10. Generalized Linear Model (M10): It consists of linear predictor, variance and link functions.

11. Linear Regression(M11): It is a statistical process used to create a linear model.

4.3.3 Result Analysis of Normalized and Un-Normalized Dataset

The PDR is a crucial parameter in evaluating the performance of the network [113]. The greater is the value of PDR, higher is the performance of the network and better is its dependability. The Figure 4.7 compares the PDR for normalized and un-normalized dataset. The red curve indicates PDR for an un-normalized dataset and the blue curve indicates a normalized dataset. The protocol used for this justification is AODV.

Similarly, Figure 4.8 compares the PDR for a normalized and un-normalized dataset. The red curve indicates PDR for an un-normalized dataset and the blue curve indicates a normalized dataset. The protocol used for this justification is DSR.

From the above discussion, it is concluded that normalized distribution yields a better outcome as compared to random deployment in the case of AODV and DSR. The Machine Learning models discussed in Table 4.5 are applied for the prediction of DF in the simulated scenarios. The results for the un-normalized and normalized dataset are compared in the course of this work, based on Correlation, Coefficient of Determination,

TABLE 4.5: Machine Learning Techniques

Model	Technique	Reference	Method	Package Included	Tuning Parameters
M1	Linear Model	[155]	lm		Optional
M2	Decision Tree	[156] [157]	rpart	rpart	parms = list(split="information"), control = rpart.control(usesurrogate=0, maxsurrogate=0)
M3	Extreme Learning Machine	[158]	elm	elmNN	nhid=10, actfun="sig"
M4	Tree Models from Genetic Algorithms	[159]	evtree	evtree	minbucket = 10, maxdepth = 2
M5	Generalized Additive Model	[155]	gam	mgcv	Optional
M6	Model Tree	[155]	tree	tree	Optional
M7	Projection Pursuit Regression	[160]	ppr	fRegression	nterms=1
M8	Bayesian Regularized Neural Networks	[80]	brnn	brnn	Optional
M9	PartyKit	[161]	ctree	party	Optional
M10	Generalized Linear Model	[155]	glm		family=poisson
M11	Linear Regression	[162]	LinearRegression		Optional

Root Mean Square Error (RMSE), Accuracy and Time Taken. The dataset used is divided into training-testing data partitions of sizes 50 - 50, 60 - 40, 70 - 30, 80 - 20 (represented in percentage).

The correlation represents an established relationship between the features of the dataset. All calculated values indicate a positive correlation between the various features of the dataset. It is calculated by:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{4.8}$$

where, a depicts actual value, b depicts predicted value, $\bar{a}$ represents actual values mean, $\bar{b}$ represents predicted values mean and n is the number of instances. The value of correlation lies in [0,1]. More the value tends towards 1, the better is the correlation. The Figure 4.9 compares the correlation for normalized and un-normalized dataset:

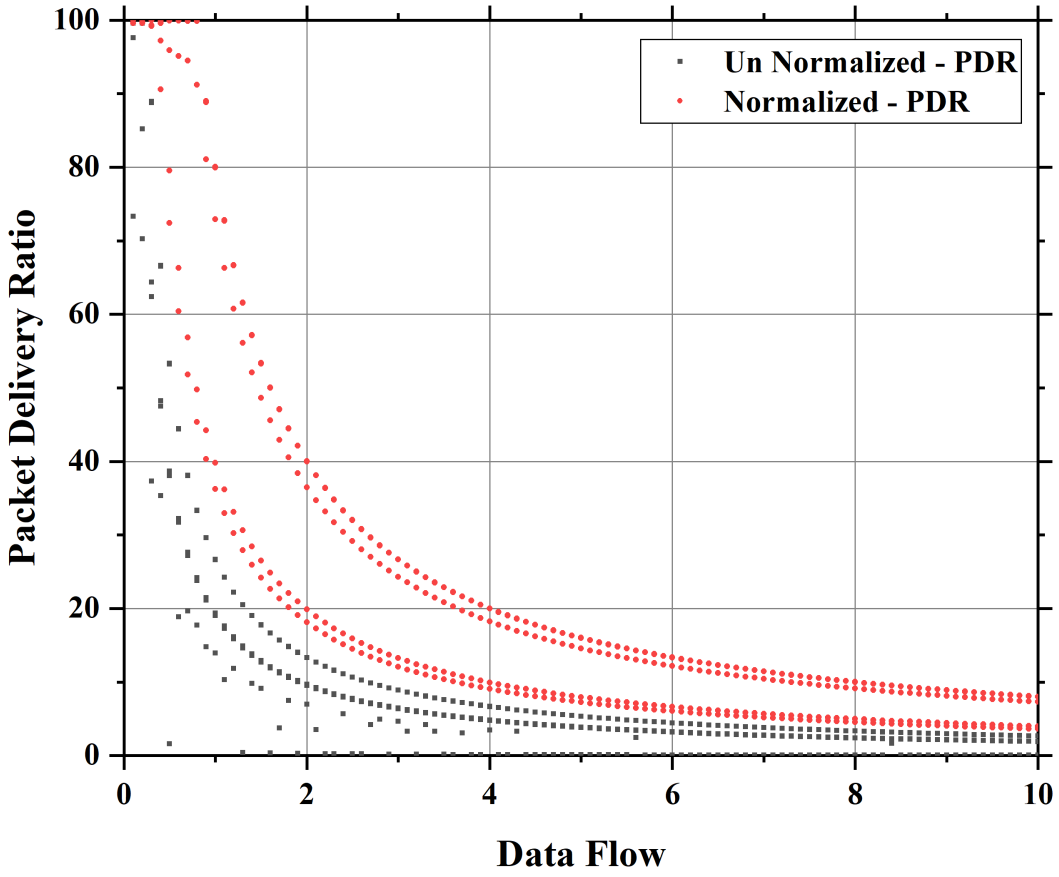


FIGURE 4.7: PDR Vs Data Rate for AODV

Coefficient of Determination indicates the goodness of fit. Higher the value of this coefficient, better fits the model and is calculated as:

$$R^2 = r * r \tag{4.9}$$

where, r depicts the value of correlation. Figure 4.10 compares the Coefficient of Determination for a normalized and un-normalized dataset:

Root Mean Square Error (RMSE) is the variation of actual and predicted values obtained using the Machine Learning models. Lower the RMSE, better is the model. It is computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \tag{4.10}$$

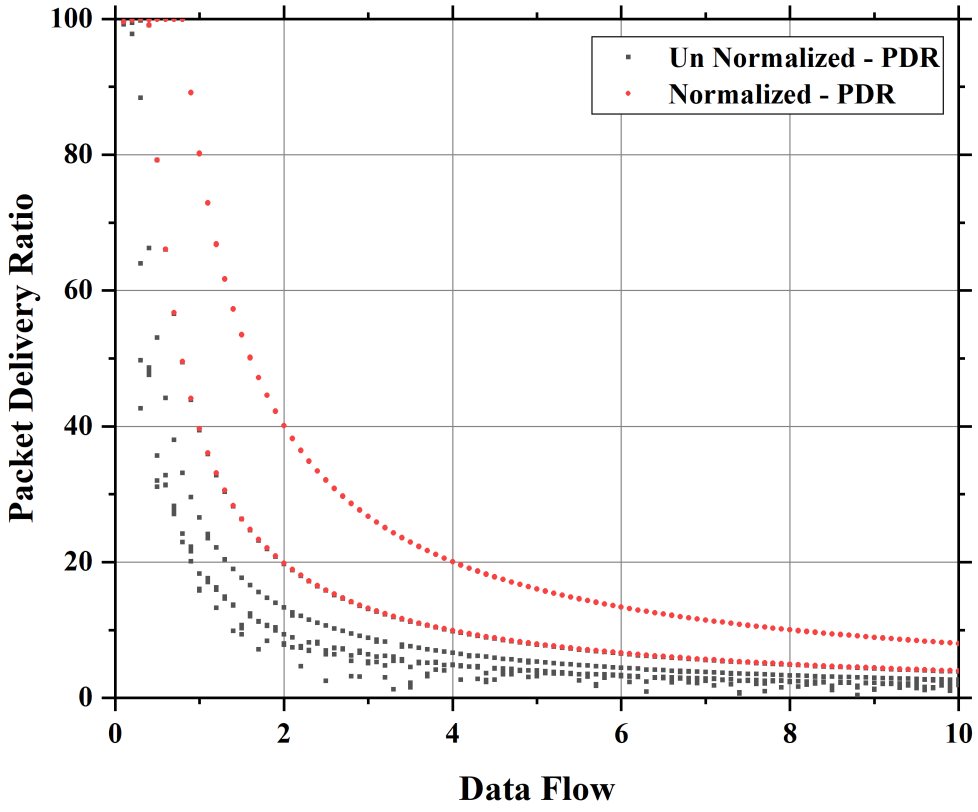


FIGURE 4.8: PDR Vs Data Rate for DSR

where, a is the actual value, p is predicted target and n are the total number of instances. The Figure 4.11 compares the RMSE for normalized and un-normalized dataset:

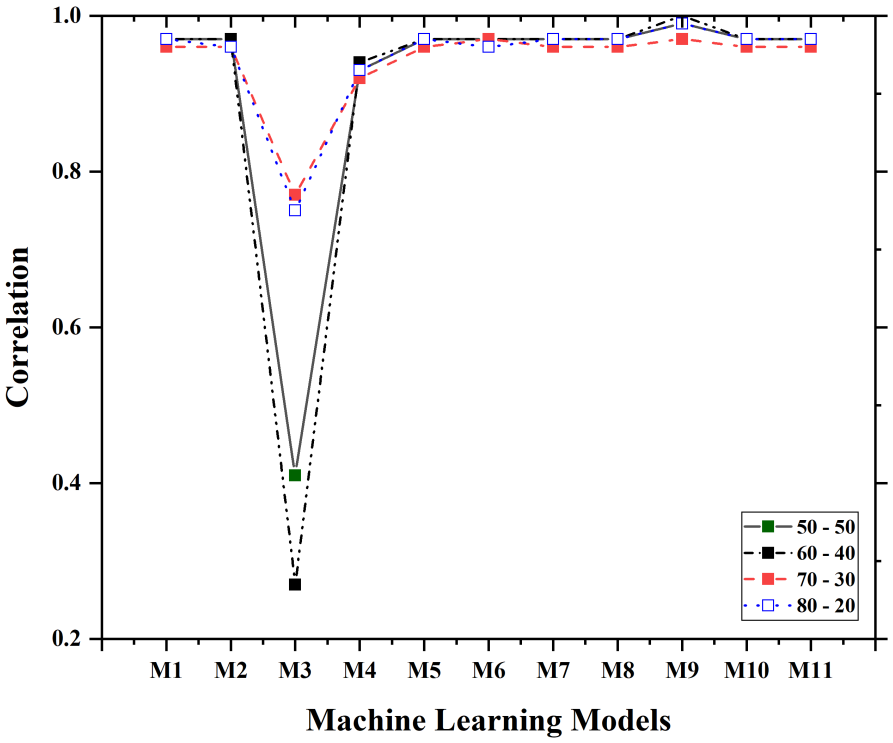
Accuracy measures the correctness of a Machine Learning model. Higher accuracy is preferred for good results. It is computed as:

$$Accuracy = \frac{100}{n}\sum_{i=1}^{n} q_i \tag{4.11}$$

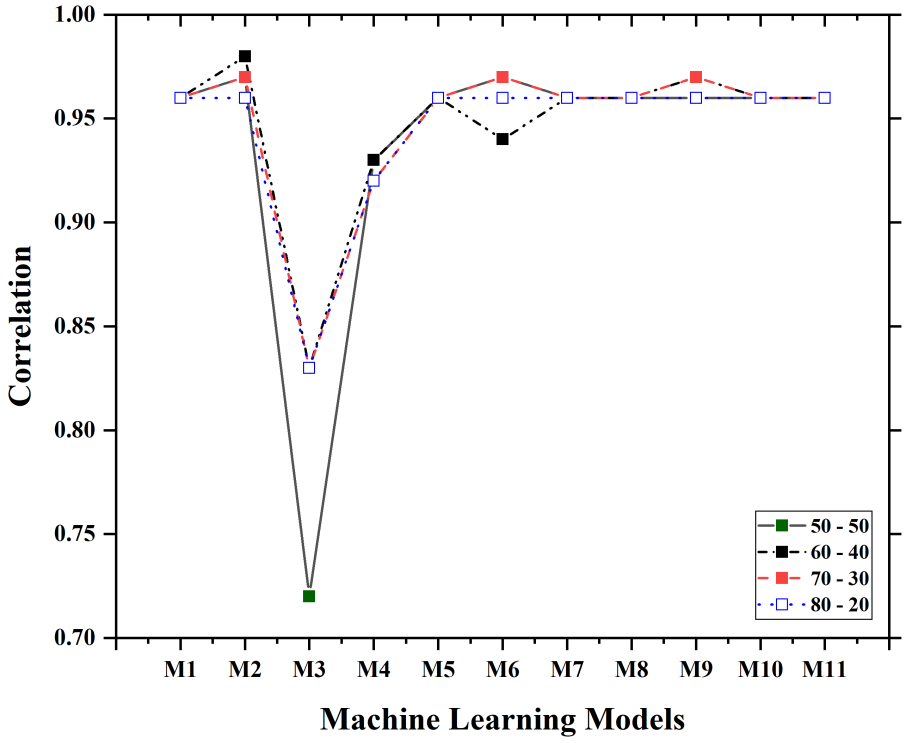
$$q_i = \begin{cases} 1 & if abs(p_i - a_i) \leq err \\ 0 & otherwise \end{cases}$$

where, a depicts actual values, p depicts predicted target, err represents acceptable error and n is number of instances. The Figure 4.12 compares the Accuracy for normalized and un-normalized dataset:

Time Taken is the overall duration taken by a Machine Learning model for execution. The Figure 4.13 compares the time taken for normalized and un-normalized dataset:

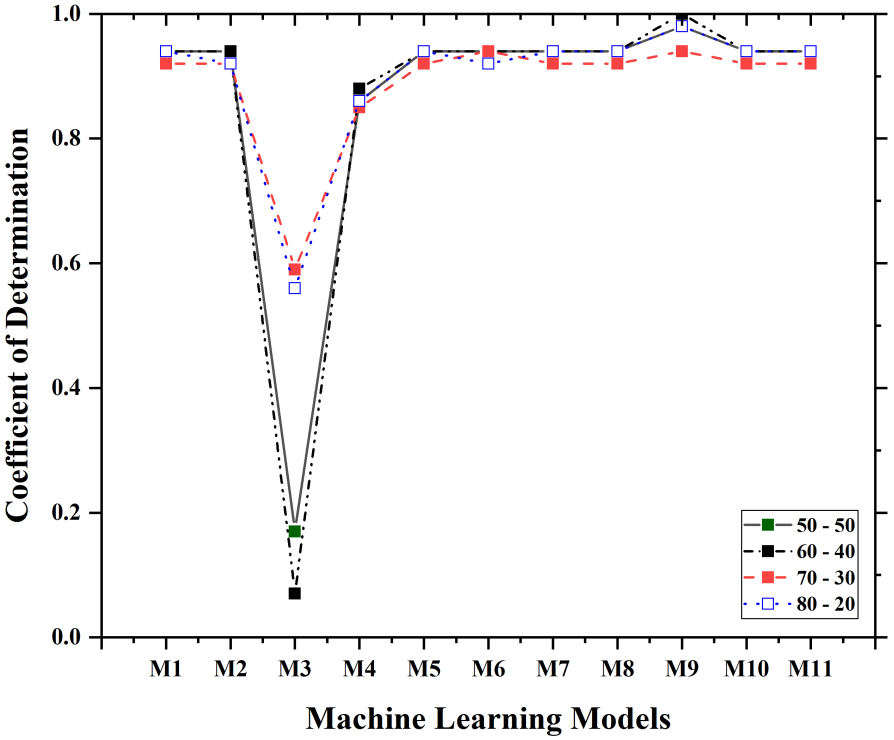


(a) Un-normalized deployment

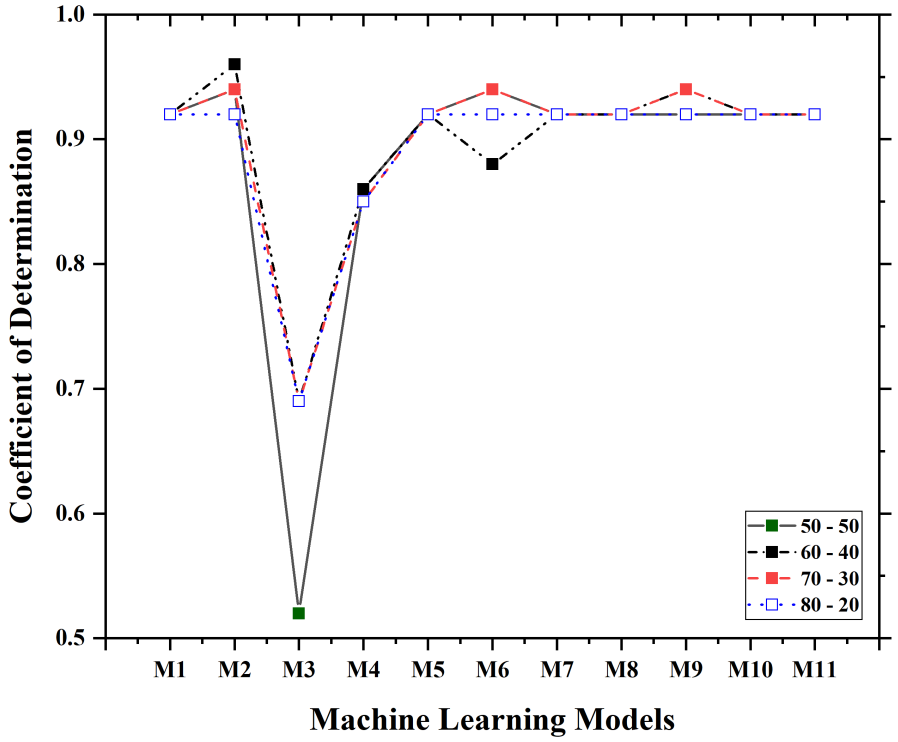


(b) Normalized deployment

FIGURE 4.9: Correlation

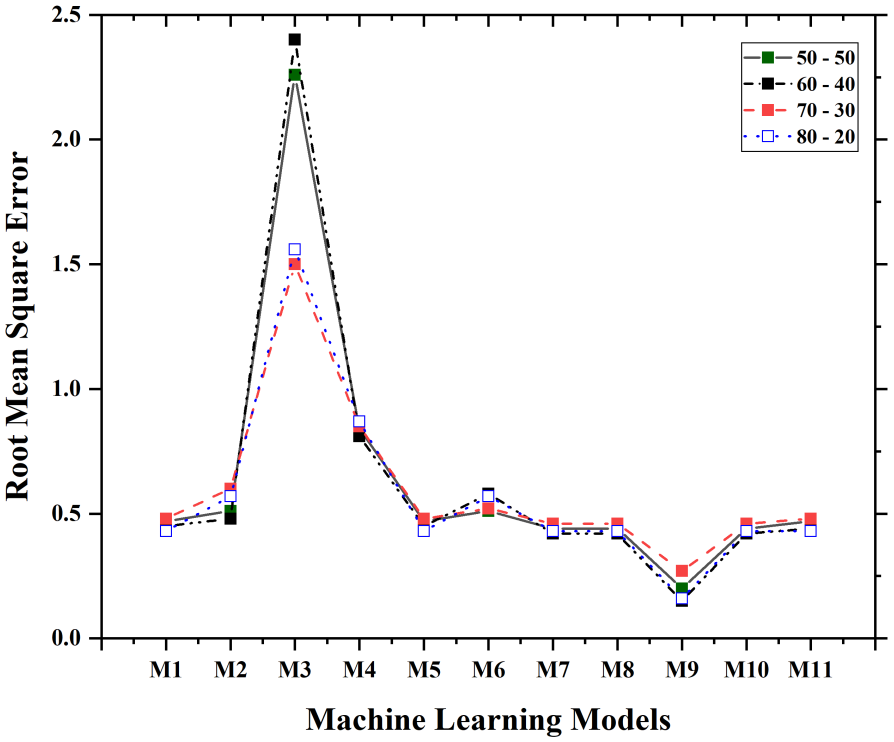


(a) Un-normalized deployment

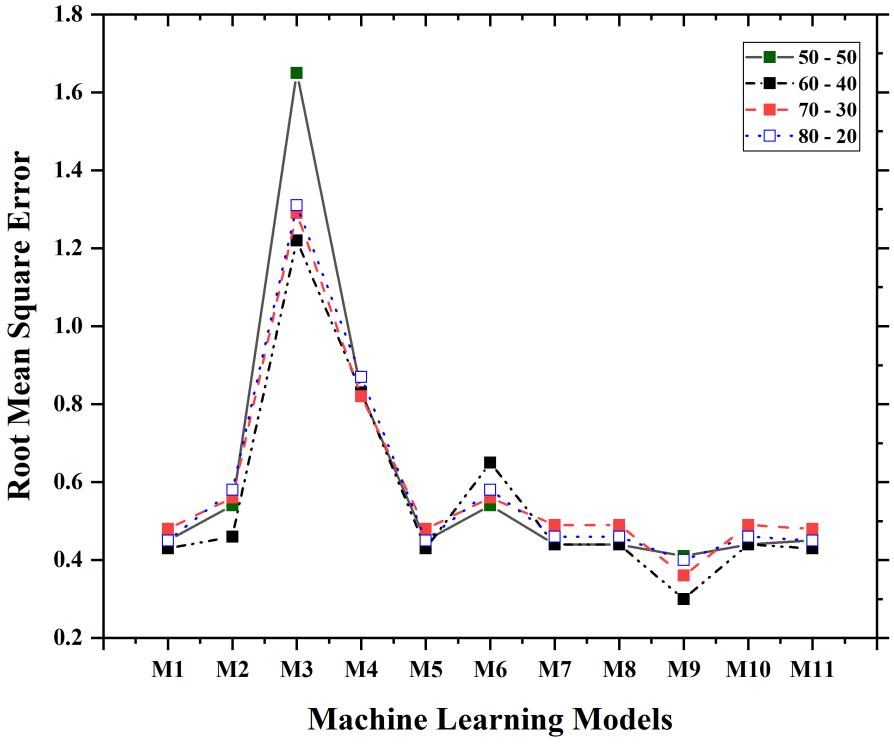


(b) Normalized deployment

FIGURE 4.10: Coefficient of Determination

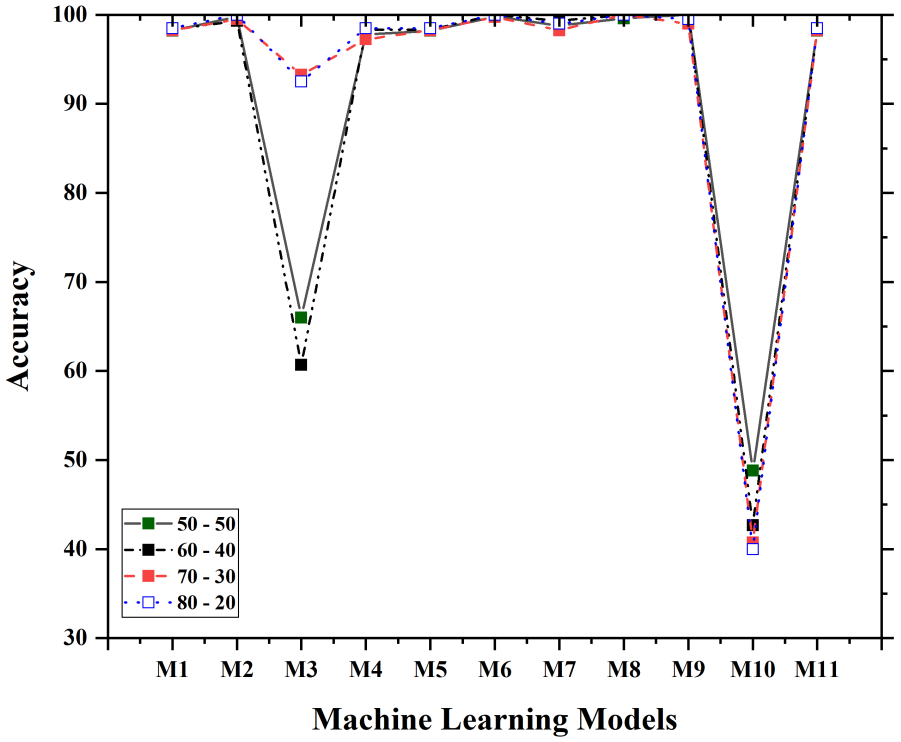


(a) Un-normalized deployment

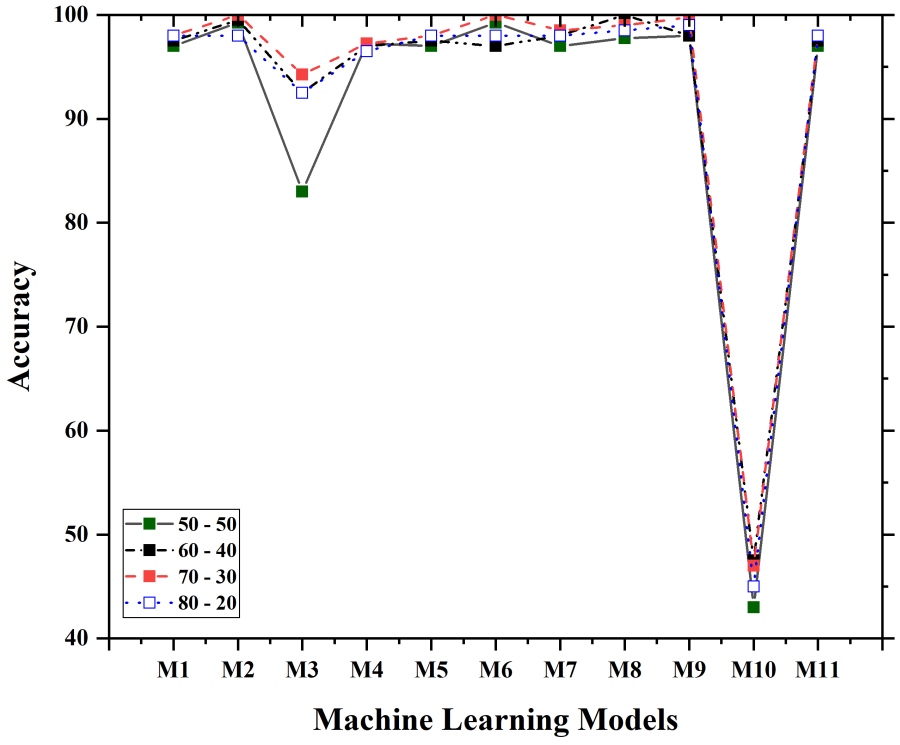


(b) Normalized deployment

FIGURE 4.11: Root Mean Square Error (RMSE)

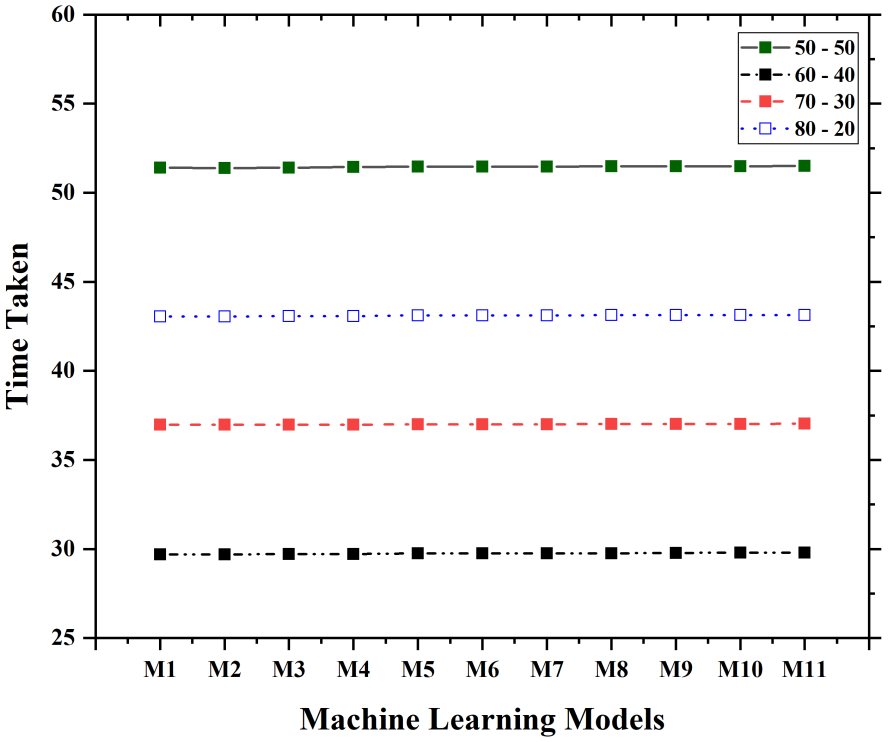


(a) Un-normalized deployment

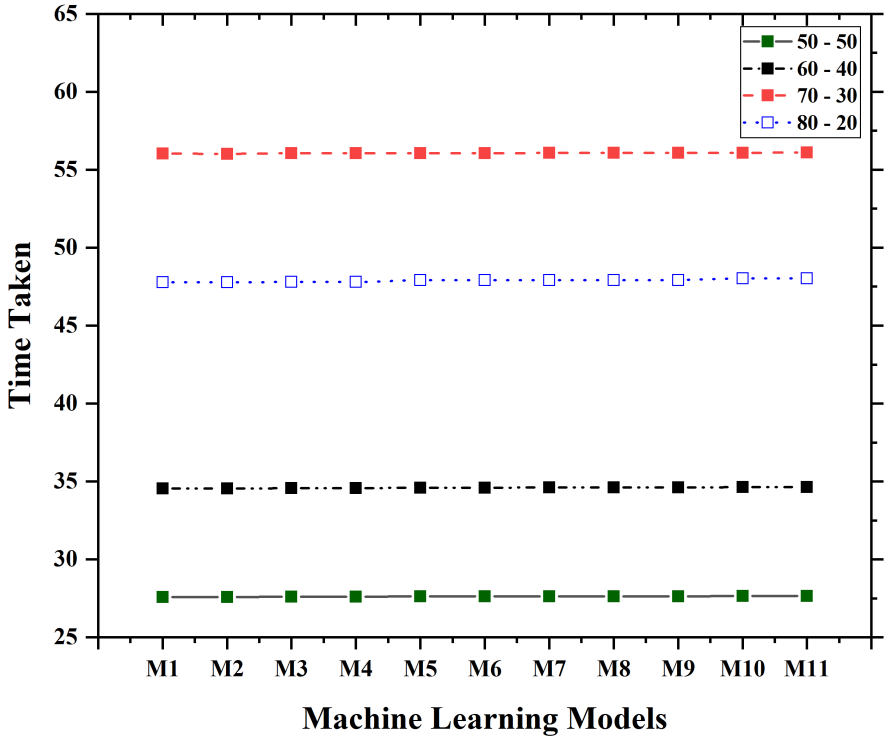


(b) Normalized deployment

FIGURE 4.12: Accuracy



(a) Un-normalized deployment



(b) Normalized deployment

FIGURE 4.13: Time Taken

The above results are scaled in six levels. EX being the highest stands for Excellent, VG stands for Very Good, GD stands for Good, AV stands for Average, PR stands for Poor and NR being the lowest stands for the model which is Not Recommended. The quantitative evaluation can be referred from Annexure-D.

The models which are good in terms of correlation for a normalized dataset are shown in Table 4.6. Decision Tree is the strongly recommended model in terms of Correlation for a 70-30 dataset partition. The models which are good in terms of Coefficient of Determination for a normalized dataset are shown in Table 4.6. Decision Tree and Model Tree fits best for 50-50, 60-40 dataset partition. Decision Tree and PartyKit fits best for 70-30 dataset partition. The models which are good in terms of Root Mean Square Error (RMSE) for a normalized dataset are shown in Table 4.6. Party Kit suits well to all the dataset partitions of 50-50, 60-40, 70-30 and 80-20.

TABLE 4.6: Scaling of Correlation, Coefficient of Determination and Root Mean Square Error in the Normalized Dataset

Models	Correlation				Coefficient of Determination				Root Mean Square Error			
	50-50	60-40	70-30	80-20	50-50	60-40	70-30	80-20	50-50	60-40	70-30	80-20
M1	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV
M2	VG	VG	EX	VG	EX	EX	EX	VG	PR	PR	AV	PR
M3	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
M4	GD	AV	GD	AV	AV	AV	AV	AV	PR	PR	PR	PR
M5	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV
M6	VG	VG	GD	VG	EX	EX	AV	VG	PR	PR	PR	PR
M7	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV
M8	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV
M9	VG	VG	VG	VG	VG	EX	EX	VG	AV	AV	GD	AV
M10	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV
M11	VG	VG	VG	VG	VG	VG	VG	VG	AV	AV	AV	AV

The models which are good in terms of Accuracy for a normalized dataset are shown in Table 4.7. Decision Tree and Model Tree fits best for 50-50 dataset partition. Decision Tree, Bayesian Regularized Neural Networks (BRNN), Party Kit and Model Tree fits best for 60-40 dataset partition. Decision Tree and BRNN fits best for 70-30 dataset partition. Party Kit outperforms in case of 80-20 dataset partition. The models which are good in terms of Time Taken for a normalized dataset is shown in Table 4.7:

TABLE 4.7: Scaling of Accuracy and Time Taken in Normalized Dataset

Model	Accuracy				Time Taken			
	50-50	60-40	70-30	80-20	50-50	60-40	70-30	80-20
M1	VG	VG	VG	VG	EX	NR	VG	PR
M2	EX	EX	EX	VG	EX	NR	VG	PR
M3	PR	GD	GD	GD	EX	NR	VG	PR
M4	VG	VG	VG	VG	EX	NR	VG	PR
M5	VG	VG	VG	VG	EX	NR	VG	PR
M6	EX	EX	VG	VG	EX	NR	VG	PR
M7	VG	VG	VG	VG	EX	NR	VG	PR
M8	VG	EX	EX	VG	EX	NR	VG	PR
M9	VG	EX	VG	EX	EX	NR	VG	PR
M10	NR	NR	NR	NR	EX	NR	VG	PR
M11	VG	VG	VG	VG	EX	NR	VG	PR

Highlights: The following conclusions are drawn from the study of DF in case of a normalized dataset:

1. The best-fit Machine Learning models depends on the dataset partition (Training-Testing) size.
2. Decision Tree is strongly recommended for 50-50 and 70-30 training-testing dataset partition; Party Kit for 60-40 and 80-20.
3. Bayesian Regularized Neural Networks is recommended for 50-50, 60-40 and 80-20 training-testing dataset partition; Generalized Additive Model, Projection Pursuit Regression is best for 70-30.
4. The averagely recommended models for 50-50 and 70-30 partitions are Generalized Linear Model and Tree Model from Genetic Algorithm; for 60-40 partition Generalized Additive Model and Linear Regression model are well-suited; for 80-20 partition, Decision Tree and Model Tree are used.
5. The proposed work highlights the use of Machine Learning to predict the data flow, now, if we can achieve optimum data flow in the network, it suggests that there is no bad node. If in any case, the optimum data flow is not achieved it underlines the malfunctioning of the nodes, thus, necessitating further action to be taken.

4.4 Summary

It is concluded that normalized distribution enhances the Packet Delivery Ratio as compared to randomized distribution. Also, this distribution reduces the flooding effect on the network. The Machine Learning models are applied to both the un-normalized and normalized datasets to predict data flow, which is responsible for flooding in a network scenario. These models perform differently according to the size of training and testing dataset partitions. The strongly recommended models for the prediction of data flow in the normalized dataset are Decision Tree and PartyKit. But, the time taken in the case of un-normalized data is much greater than that of normalized data. Hence, it is concluded that we need different Machine Learning models for defying the flooding attack for different applications i.e. if we have a time-critical application, we can$'$t waste much time in learning. Therefore, Linear Model and Decision Tree are recommended. Whereas on the other extreme we may need to train a model extensively for its accuracy, so in that case, the Party Kit model is recommended. The future scope of this work can be the study of prediction for Denial-of-Service attack using Machine Learning models. The work can be enhanced by considering other network performance parameters.

Chapter 5

Traffic Flow in Wireless Sensor Networks: An Intelligent Neural Network Perspective

"Science may set limits to knowledge, but should not set limits to imagination."

-Bertrand Russell

With the advancement of technology and online trading sectors, it has become a necessity to get fast and correct information in real-time scenarios. The efficient traffic flow[1] has become a perplexing dilemma to the people connected worldwide. In that case, the intelligent technique D-FNN (Dynamic Fuzzy Neural Network) was devised to assuage the traffic flow prediction [163, 164]. It consists of two main modules: the neural network and fuzzy system.

As the core idea of D-FNN is establishing a complete network structure using an automated approach; therefore, the emphasis is on prediction accuracy in a short time span. Whereas, the current real-world scenario requires traffic flow prediction [165] with greater accuracy as well as the lesser error rate. These models hence can be chosen as

[1]The contents of the chapter are under-review in:
Jasminder Kaur Sandhu, Anil Kumar Verma, Prashant Singh Rana, "Predicting traffic flow in wireless sensor networks: An intelligent neural network perspective", Wireless Personal Communications, SCI-Indexed, Impact Factor 1.200

milestones for futuristic networks. Therefore, the key research point is predicting the traffic flow in different networks [166, 167] with higher accuracy and low error rate.

The trend thus far convey that much of the work has been done in this area, such as stochastic differential equations [168], Support Vector Machine [169, 170], gaussian techniques [171], neural networks [172, 173], fuzzy logic [174] and other complex modelling approaches, as discussed in [175, 176, 177, 178, 179]. In these methods, neural networks based techniques are comparatively efficient for prediction of traffic flow. Also, neural network modeling has been performed with fuzzy logic and proves to be optimum. Our work is a novel approach for prediction which deals with both static and dynamic deployment of sensor nodes. This work more efficiently predicts the traffic flow in Wireless Sensor Networks (WSN) by exploiting the advantages of neural networks [180]. The availability of numerous neural network-based models has added an advantage to prediction. The novelty of the work is described in the following facets:

1. The diversified neural network models are studied to predict the traffic flow in the real-time environment of the WSN.
2. A first-hand dataset is being used for this work collected using network simulations and then predictions are made according to Machine Learning models.
3. The statistical measures such as correlation, coefficient of determination, Root Mean Square Error and accuracy are considered for comparison of these models.
4. The models applied are capable of detecting all possible synergies between predictor variables.

5.1 Proposed Workflow for Data Flow Prediction

The dependability of a network is based upon the data flow of that particular network. Regularized data flow helps in achieving optimality for a network. Our proposed workflow deals with the real-time environment of WSN and is divided into the following four phases as shown in Figure 5.1:

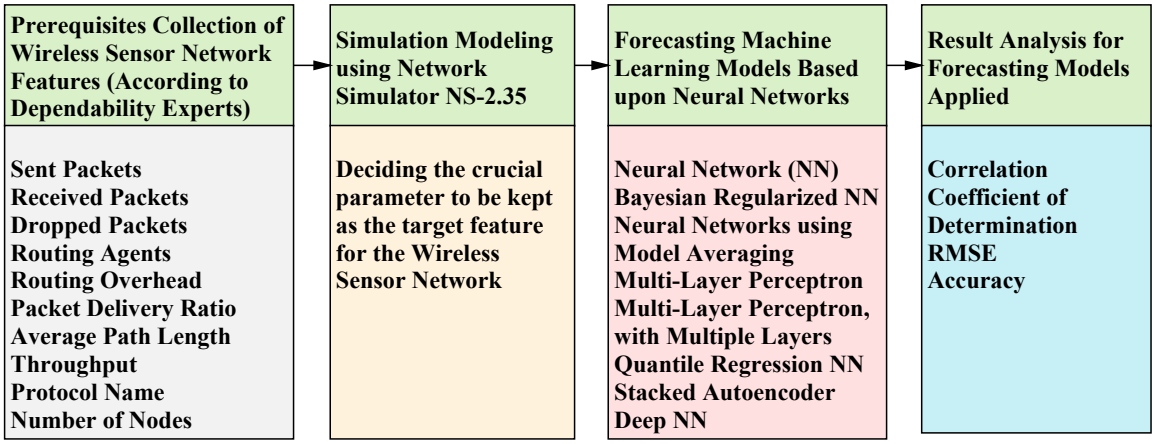


FIGURE 5.1: The Proposed Workflow

In the first phase, the network features are chosen based upon the knowledge of dependability experts as surveyed in the literature. The second phase simulates the WSN scenarios to generate a dataset and also, selects the feature to be predicted to enhance dependability of the network. This in turn enhances performance of the network. The next phase, applies various neural network based Machine Learning models to the dataset which is available as supplement at $http://bit.ly/NetNeural$. The final phase interprets which model is best suited to this application. The model is selected based upon correlation, coefficient of determination, Root Mean Square Error (RMSE) and accuracy. The subsequent sections explain these phases in detail.

5.2 Prerequisites Collection

Prerequisites are also known as characteristics of a network [11, 88, 89]. These have been collected based upon the knowledge of dependability experts as surveyed in the literature and includes:

- **Sent Packets (SP)**: The number of packets dispersed from transmitter to the receiver node.
- **Received Packets (RP)**: These are the packets received at the destination node.
- **Dropped Packets (DP)**: The number of packets expelled during communication from source to destination.

- **Routing Agents (RA)**: It observes the metrics for one-to-one linking in a network and stimulates the routing table.
- **Routing Overhead (RO)**: For successful communication, the number of routing packets engaged forms the routing overhead.
- **Packet Delivery Ratio (PDR)**: It is the ratio of packets delivered to packets sent and is expressed in terms of percentage.
- **Average Path Length (APL)**: It determines the number of hops (popularly known as hop count) a packet has to travel till destination.
- **Throughput (TH)**: It is a measure of how fast the data can be sent through a network and is expressed in Kbps (Kilobits per second). It is preferred over bandwidth because it is a practical measured parameter, whereas, the bandwidth is more of a theoretical parameter.
- **Protocol Name (PN)**: It signifies the protocol being used in communication. This work considers two protocols, namely AODV (Ad-hoc On-demand Distance Vector) and DSR (Dynamic Source Routing) protocol.
- **Number of Nodes (NUM)**: The number of nodes in a network varies from 5 to 50 in number. Both static and dynamic communication takes place between these nodes.

5.3 Simulation Modeling

The simulation modeling has been carried out with the help of network simulator NS-2.35 [91]. The direction of communication between nodes is omnidirectional and takes place using reactive protocols AODV and DSR. The simulation has been performed with variable data flow rates. The details of network simulation using NS-2.35 can be referred from Annexure-A. The simulation parameters set for generating these scenarios are discussed below in Table 5.1:

TABLE 5.1: NS-2 Simulation Parameters

SN	Parameter	Value
1	Channel Type	Wireless
2	Radio-propagation model	Two-Ray Ground
3	Interface queue type	DropTail/CMUPriQueue
4	Antenna Model	Omnidirectional
5	Max packet in Interface Queue	150
6	Number of nodes	5-50
7	Data Flow	0.1-10 Mb
8	Routing protocols	AODV/DSR
9	X dimension of topography	1000 m
10	Y dimension of topography	1000 m
11	Simulation Time	20 ms

5.4 Forecasting Machine Learning Models

The following techniques are applied for data flow prediction in a WSN. All these techniques are based upon the concept of neural networks. The details of R programming used for implementing Machine Learning models can be referred from Annexure-B. An Artificial Neural Network (ANN) or simply Neural Network is described as a reticulum consisting of enormous neurons (or processors). These neurons are densely connected and operate in a synchronized fashion. The reticulum is trained using the examples from the first-hand real-time data.

5.4.1 Neural Network Model

It is an elementary implementation of the neural network, to predict data flow rate or simply data flow of a WSN from the total number of sent packets, received packets, dropped packets, routing agents, routing overhead, packet delivery ratio, average path length, throughput, protocol name, and the number of nodes. The tuning parameters included, to train the neural reticulum includes the number of hidden neurons in every layer, the sum of squared errors is used as an assessment function for error computation and a logical linear output function which is set to false (logical value) as smoothing of result does not take place [181]. The reticulum thus formed, is quite complex.

5.4.2 Bayesian Regularized Neural Network Model

In this work, the Bayesian Regularized Neural Network (BRNN) is used as a novel method to predict network behavior. It has the capability to optimize complex models and are more resilient as compared to ANN. This model is based upon a two-layer network as proffered by MacKay (1991) and Hagan, Foresee (1997) [182], given by the following equations:

$$y_k = f(p_k) + e_k \tag{5.1}$$

$$y_k = \sum_{i=1}^{n} w_i g_i(b_i + \sum_{j=1}^{m} p_{pk} \beta_j^{[i]}) + e_k \tag{5.2}$$

where: $e_k \sim N(0, \sigma_e^2)$
m is the number of neurons
w_i is the weight of the i^{th} neuron, $i = 1, 2, ..., m$
b_i is a bias for the i^{th} neuron, $i = 1, 2, ..., m$
$\beta_j[i]$ is weight of the j^{th} input to the reticulum, $j = 1, 2, ..., m$
$f_i(.)$ is the activation function
Also,

$$f_i = \frac{exp(2p) - 1}{exp(2p) + 1} \tag{5.3}$$

where:

This model minimizes to:

$$Q = \beta E_R + \alpha E_{WB} \tag{5.4}$$

where:
$E_R = \sum_{k=1}^{n} (y_k - \hat{y}_k)^2$, which calculates the Error Sum of Squares
E_{BW} is the Sum of square of network parameters (biases and weights)
$\beta = \frac{1}{2\sigma_e^2}$
$\alpha = \frac{1}{2\sigma_\theta^2}$

σ_θ^2 is a dispersion parameter used for weights and biases
The optimal training parameters include the number of neurons, the logical parameter normalize, maximum number of iterations to train the model, the minimum gradient, numeric tolerance for checking convergence in the $Bai's$ algorithm.

5.4.3 Neural Network using Model Averaging

This model comprehends various neural network models. It can be applied to both regression and classification problems. Random seeds are generated to fit the same neural network model. The output from each of the models is averaged and used for prediction [183]. Besides, these models can also be established using Bagging. The optimal parameters used for tuning this model are size, linout, trace. The value of size has been set to 5 which indicates that 5 nodes are contained in the hidden layer. The parameter linout = TRUE specifies that the output is procured using a linear function.

5.4.4 Multi-Layer Perceptron Model (MLP)

It is based upon the supervised learning technique known as back-propagation for training the network. Each node is a neuron with non-linear activation function except the input nodes [184]. This function can mainly be modeled as:

(i) Hyperbolic tangent function with a range of -1 to 1

$$f(a_i) = tanh(a_i) \tag{5.5}$$

(ii) Logistic function with a range of 0 to 1

$$f(a_i) = (1 + e^{-a_i})^{-1} \tag{5.6}$$

where:
$f(a_i)$ represents output of i_{th} neuron and a_i is weighted summation of input synapses
One or more non-linear layers can be present between the input and output layer as shown in Figure 5.2:

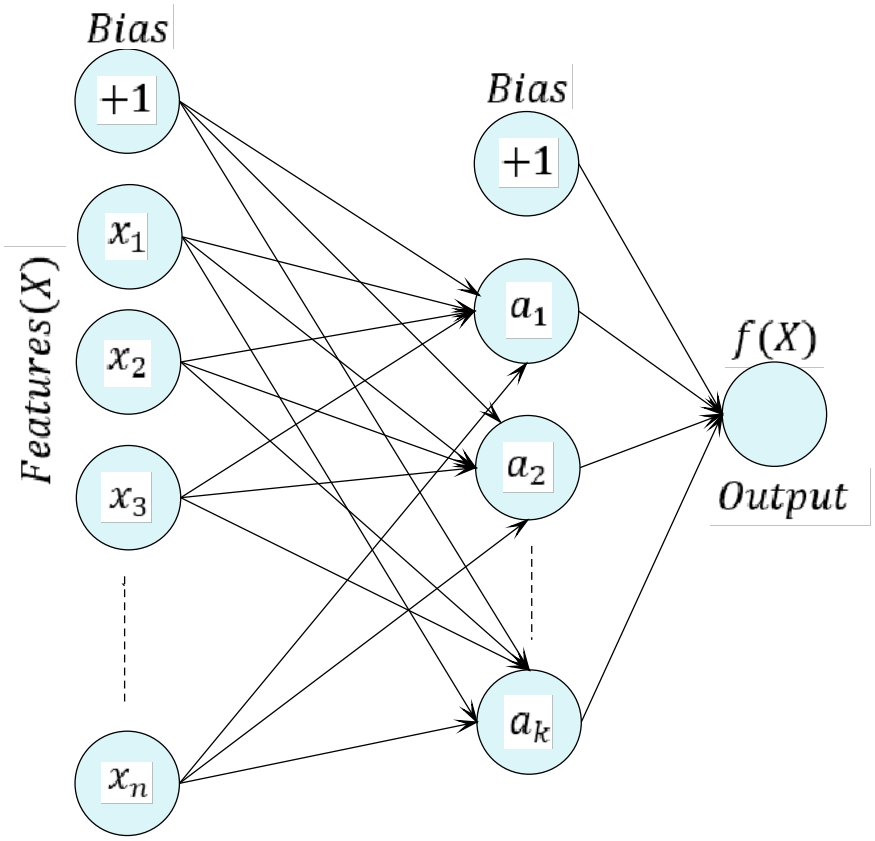


FIGURE 5.2: Generic Structure of MLP with One Hidden Layer

The input layer contains $x_i|x_1, x_2, ...x_n$ specifying the input features of the dataset. The output layer receives input from last hidden layer, obtained by linear weighted summation $w_1x_1 + w_2x_2 + ... + w_nx_n$ and a non-linear activation function, and finally metamorphose them into output values. It is well suited for our application area as it has the aptness to learn in real-time environments.

5.4.5 Multi-Layer Perceptron, with Multiple Layers (MLP-ML)

This model is efficient for non-linearly separable problems. Each intermediate layer receives activation from previous layer of processing unit and sends activations to the next layer of the network. It is an extension to the MLP model, adding on multiple intermediate layers as shown below in Figure 5.3. The input layer is labelled zero. $f_n(x)$

represents different activation functions. n being the layer number. 'O' symbolically represents output and 'w' represents weight.

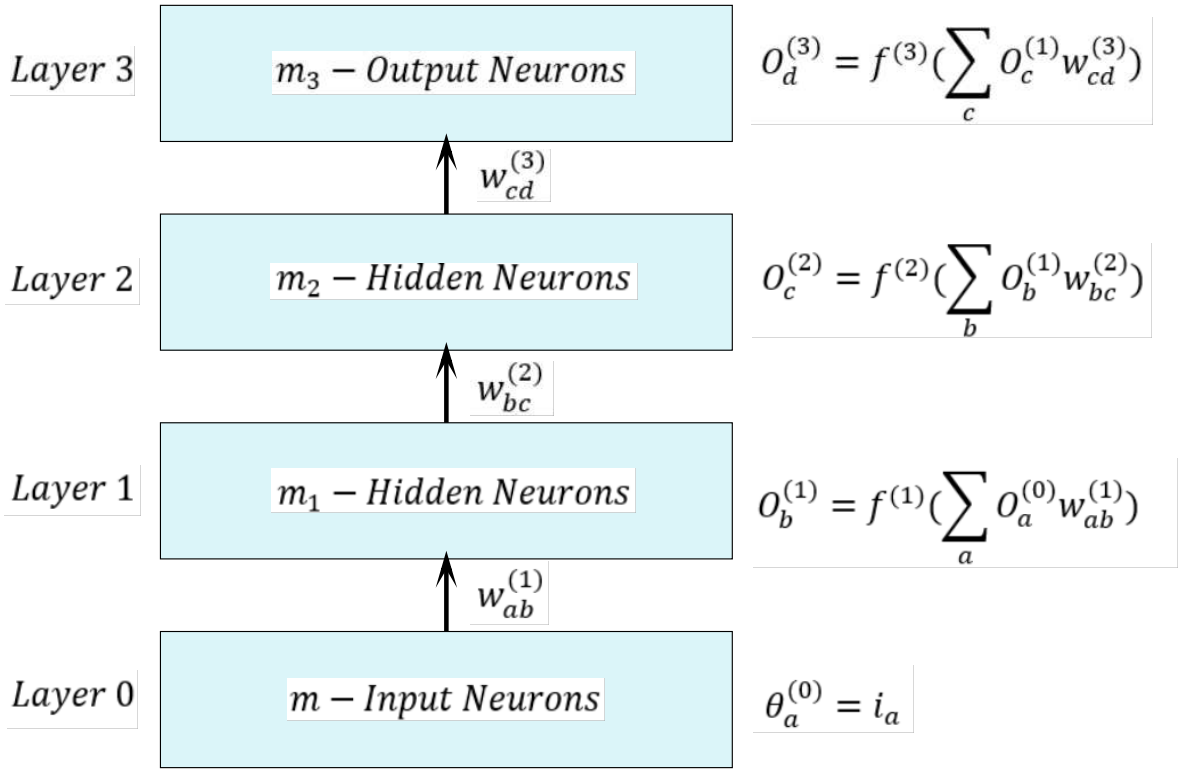


FIGURE 5.3: MLP Model with Multiple Hidden Layers

A MLP of T layers means T-layers network with T layer of weights and T layer of processing units. The flexibility of the structure is because multiple layers can be further added depending upon the problem domain [185].

5.4.6 Quantile Regression Neural Network Model (QRNN)

It is a consolidation of ANN with quantile regression implemented linearly. It can be applied on mixed variables (continuous or discrete form). The work which explains QRNN is White (1992) and Burgess (1995) [186]. The input is represented using $a_i(m)$ and output using $out(m)$, as illustrated in Figure 5.4:

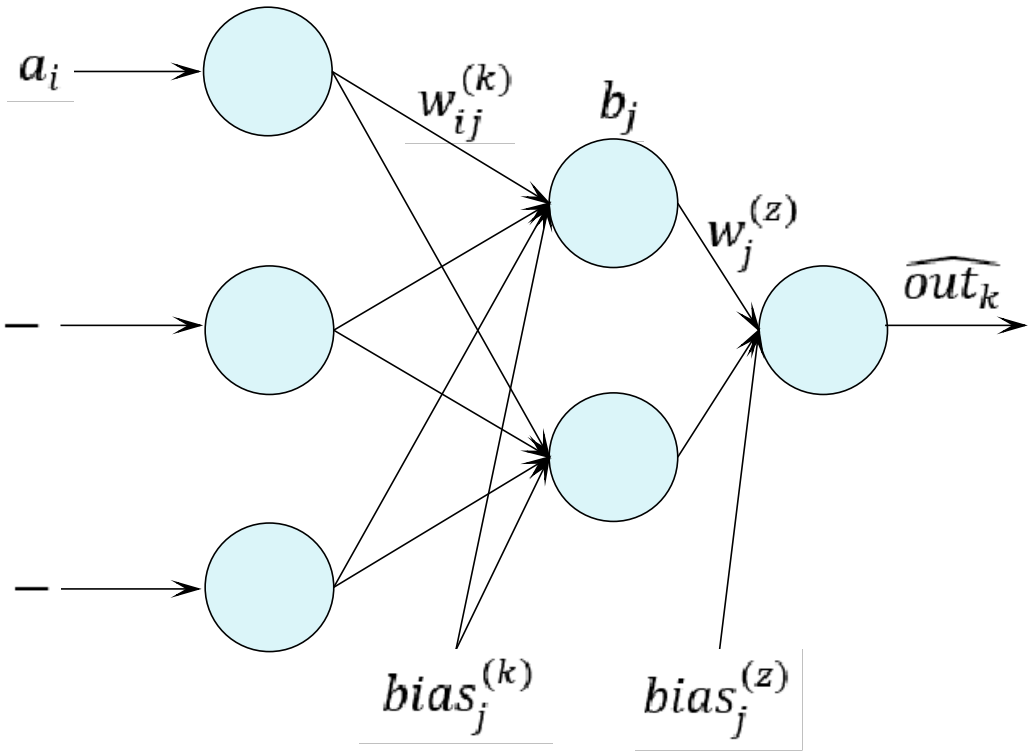


FIGURE 5.4: QRNN Reticulum

The output of j_{th} hidden layer is calculated as:

$$b_j(m) = tanh(\sum_{i=1}^{I} a_i(m) w_{ij}^{(h)} + bias_j^{(h)}) \tag{5.7}$$

where:

tanh is the hyperbolic tangent function

w_{ij}^{h} is the hidden layer weights

$bias_j^{h}$ is the bias for the hidden layer

The final equation is given by:

$$out_k^{(m)} = trf(\sum_{i=1}^{J} b_j(m) w_j^{(z)} + bias_j^{(z)}) \tag{5.8}$$

where:

$trf(*)$ is transfer function to the output function

$bias_j^{(z)}$ is the output layer bias

$w_j^{(z)}$ are weights of output layer

5.4.7 Stacked Autoencoder Deep Neural Network Model (DNN)

It is the best algorithm to represent and learn from input features. Suppose an n-layer stacked autoencoder is created. The first layer learns from first-order features of the raw input. The second layer learns from patterns appeared in the first layer. In a nutshell, n-layer learns from $(n-1)$ layer patterns. The main advantage of DNN is that it can learn from even higher-order features and has great articulating power [187]. In our work, resampling of data has been performed using 3-fold cross-validation. The tuning parameter layer 2, layer 3 and visible dropout is kept as zero. So, the final values used for the model were layer 1 = 3, layer 2 = 0, layer 3 = 0, hidden dropout = 0 and visible dropout = 0. The Root Mean Square Error (RMSE) was used to select the optimal model using the smallest value. The resampling results across tuning parameters are shown in Table 5.2:

TABLE 5.2: Resampling Results of DNN

Layer 1	Hidden Dropout	RMSE	R Squared
1	0.0	0.9068000	0.9091507
1	0.1	1.3034881	0.8268814
2	0.0	0.7351507	0.9385537
2	0.1	1.1945516	0.9106305
3	0.0	0.6244204	0.9540994
3	0.1	1.2439046	0.9176864

5.5 Result Analysis for Forecasting Machine Learning Models

In network modelling, neural network based Machine Learning techniques are used to predict the output variable data flow. Various other performance metrics elaborated in Section 5.2 forms the basic input variables. All the Machine Learning models implemented uses formula as described below:

$$DF \sim f(SP, RP, DP, RA, RO, PDR, APL, TH, PN, NUM) \tag{5.9}$$

In the above equations, function $f()$ specifies the input variables to predict data flow of network.

The data flow parameter falls under the regression problem. Various metrics which can be calculated for a regression problem are correlation, coefficient of determination, RMSE and accuracy. The following sub-sections explain these metrics:

5.5.1 Correlation (r)

Correlation provides information about how actual and predicted values are linked. It can be calculated as:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{5.10}$$

where, a depicts actual value, b depicts predicted value, $\bar{a}$ represents actual values mean, $\bar{b}$ represents predicted values mean and n is the number of instances. Value of correlation lies in [0,1]. The more value tends towards 1, the better is the correlation. The results of correlation on varied training-testing dataset partitions is illustrated in Table 5.3 below:

TABLE 5.3: Correlation

Models	50-50	60-40	70-30	80-20	90-10
NN	0.98	0.97	0.97	0.97	0.98
BRNN	0.98	0.97	0.97	0.97	0.98
NN-MA	0.88	0.94	0.91	0.96	0.91
MLP	0.99	0.99	0.98	0.99	0.99
MLP-ML	0.98	0.99	0.99	0.99	0.99
QRNN	0.98	1	1	1	1
DNN	0.98	0.97	0.98	0.98	0.98

The correlation have been elaborated with the help of Figure 5.5:

5.5.2 Coefficient of Determination (R^2)

The Coefficient of Determination (R^2) is the primary outcome of regression evaluation and its value lies in the interval $0 < R^2 < 1$. The more value tends towards 1, the better

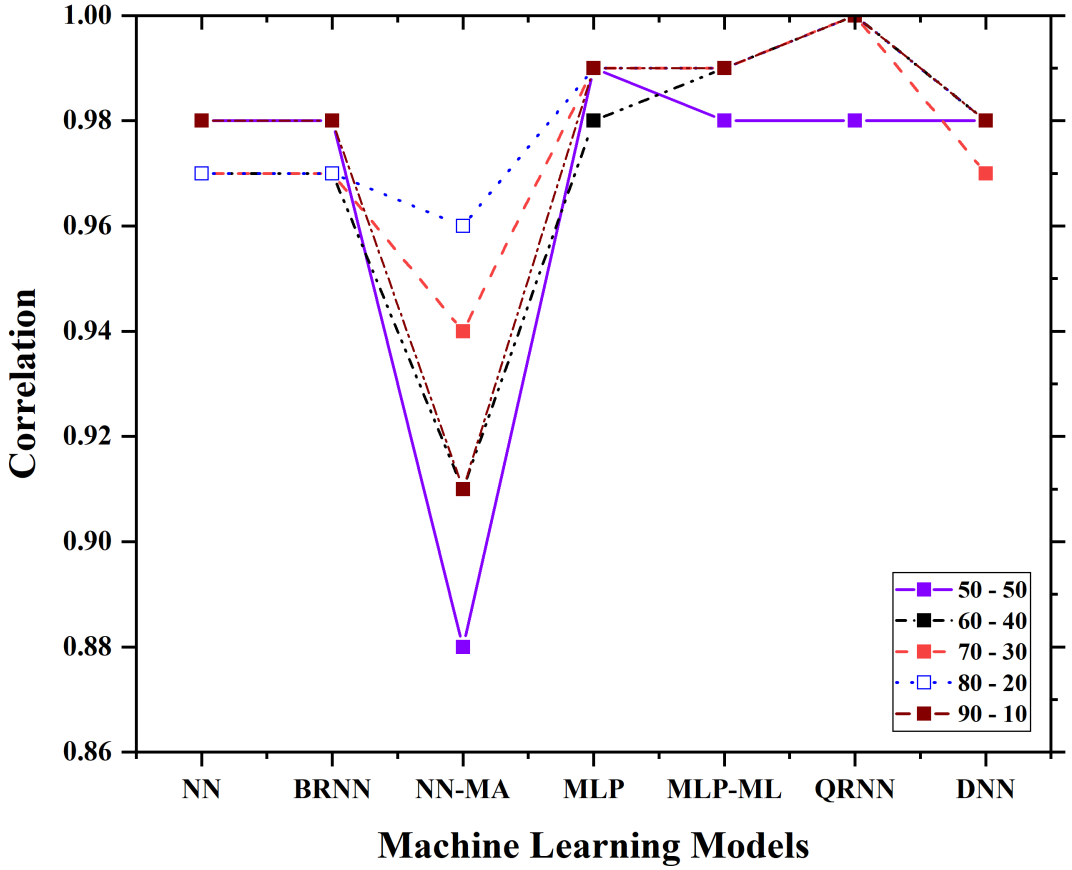


FIGURE 5.5: Correlation Analysis

is the regression model. If the value is zero that means the regression model is a failure. Mathematically, it is the square of correlation. It can be computed as:

$$R^2 = r * r \tag{5.11}$$

The results of Coefficient of Determination on varied training-testing dataset partitions is illustrated in Table 5.4 below:

TABLE 5.4: Coefficient of Determination

Models	50-50	60-40	70-30	80-20	90-10
NN	0.96	0.94	0.94	0.94	0.96
BRNN	0.96	0.94	0.94	0.94	0.96
NN-MA	0.77	0.88	0.83	0.92	0.83
MLP	0.98	0.98	0.96	0.98	0.98
MLP-ML	0.96	0.98	0.98	0.98	0.98
QRNN	0.96	1	1	1	1
DNN	0.96	0.94	0.96	0.96	0.96

The coefficient of determination have been elaborated with the help of Figure 5.6:

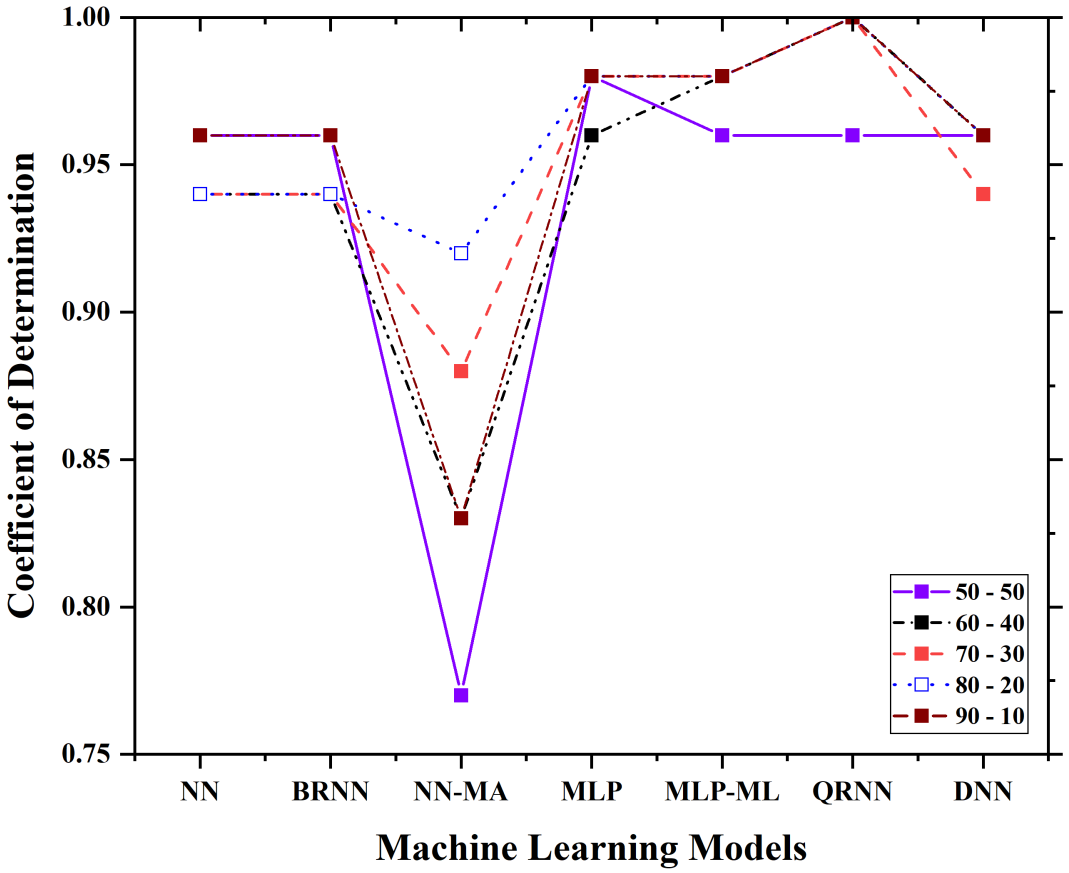


FIGURE 5.6: Coefficient of Determination Analysis

5.5.3 Root Mean Squared Error

Root Mean Square Error (RMSE) tells us how much error is present between the actual and predicted values and is used for numerical analysis of predicted results. It can be computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \tag{5.12}$$

where, a is actual target, p is predicted target and n is the total number of instances. The results of RMSE on varied training-testing dataset partitions is illustrated in Table 5.5 below. The RMSE have been elaborated with the help of Figure 5.7:

TABLE 5.5: Root Mean Square Error

Models	50-50	60-40	70-30	80-20	90-10
NN	0.43	0.44	0.43	0.42	0.39
BRNN	0.38	0.29	0.26	0.28	0.25
NN-MA	1.32	2.18	2	1.75	1.64
MLP	0.6	0.24	0.43	0.42	0.35
MLP-ML	0.31	0.43	0.36	0.37	0.46
QRNN	0.16	0.09	0.09	0.09	0.09
DNN	0.38	0.54	0.42	0.37	0.31

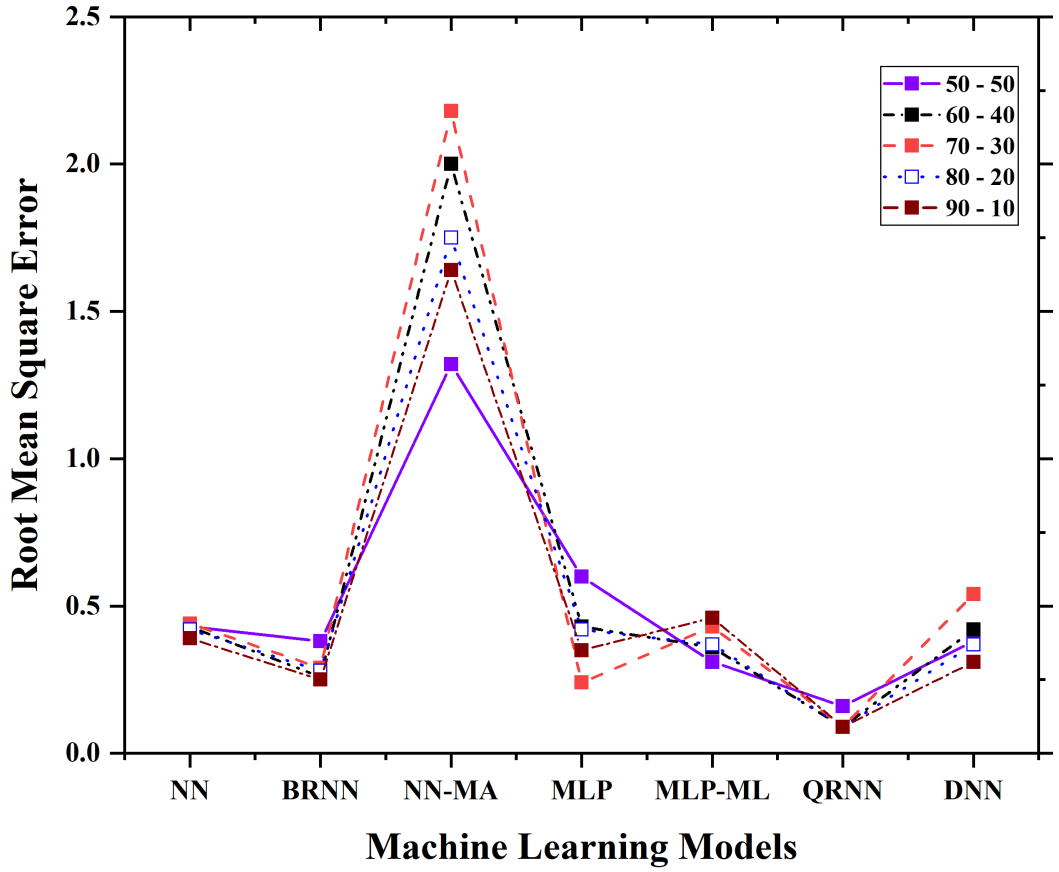


FIGURE 5.7: RMSE Analysis

5.5.4 Accuracy

The accuracy represents how close the predicted value is to the actual value within acceptable error limits. It predicts which model best suits a problem with an acceptable error rate kept at less than or equal to 2. It is computed as:

$$Accuracy = \frac{100}{n} \sum_{i=1}^{n} q_i \tag{5.13}$$

$$q_i = \begin{cases} 1 & if abs(p_i - a_i) \leq err \\ 0 & otherwise \end{cases}$$

where, a depicts actual target, p depicts predicted target, err represents acceptable error and n is number of instances. The results of accuracy (expressed in percentage) on varied training-testing dataset partitions is illustrated in Table 5.6 below. Also, the accuracy have been elaborated with the help of Figure 5.8:

TABLE 5.6: Accuracy

Models	50-50	60-40	70-30	80-20	90-10
NN	97.4	96.25	96.83	96.75	98
BRNN	99.2	98.88	100	98.5	99.5
NN-MA	73.1	97.88	85.5	95.25	74.5
MLP	98.5	99.88	99.67	100	99.5
MLP-ML	98.8	100	100	99.5	100
QRNN	98.1	99.88	100	100	99.5
DNN	97.6	97.75	98.17	98.25	98.5

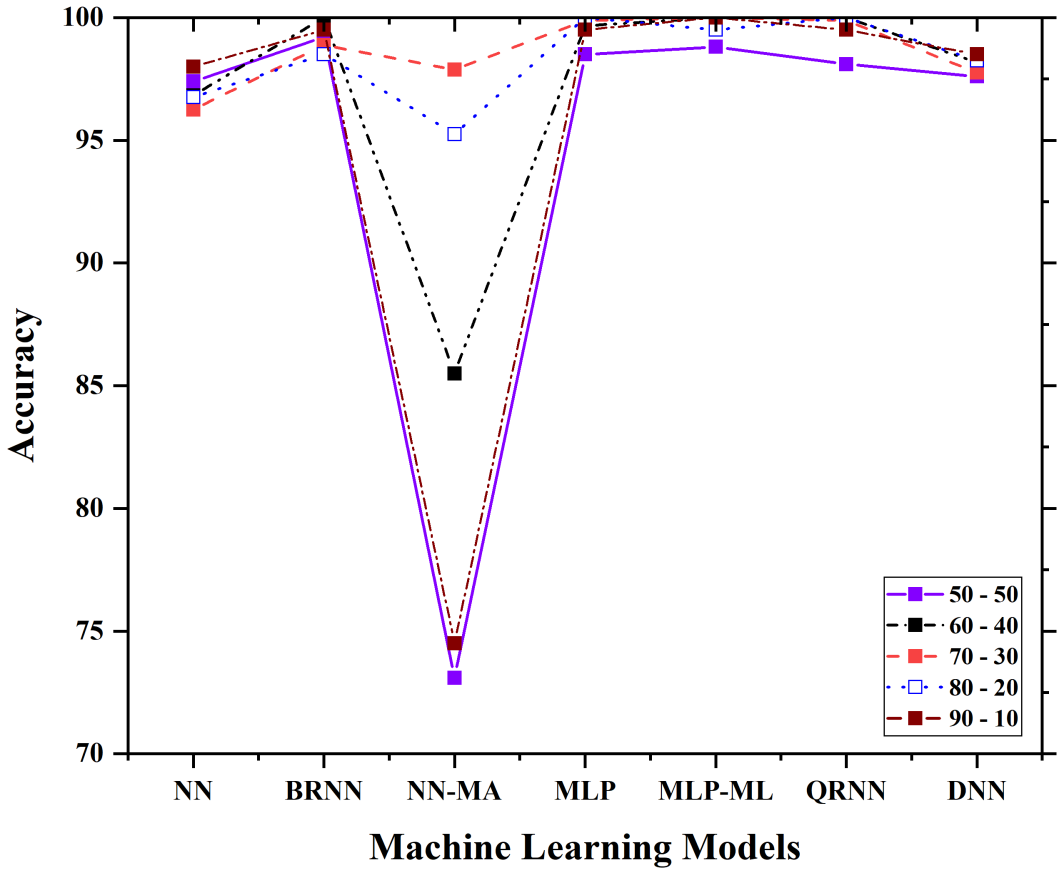


FIGURE 5.8: Accuracy Analysis

5.6 Summary

In this chapter, neural network based Machine Learning techniques have been applied to predict the data flow. These techniques include seven (7) different approaches, namely neural network, bayesian regularized neural networks, neural networks using model averaging, multi-layer perceptron, multi-layer perceptron with multiple layers, quantile regression neural network and stacked autoencoder deep neural network. Based on the numerical experiments performed upon dataset partitions of 50-50, 60-40, 70-30, 80-20 and 90-10 (training-testing partition sizes), we conclude that for efficient results, quantile regression neural network and multi-layer perceptron with multiple layers can be used. For quantile regression neural network, the correlation, coefficient of determination, the root mean square error and accuracy is 1, 1, 0.09 and 99.5% respectively. Further, multi-layer perceptron with multiple layers, the correlation , coefficient of determination, root mean square error and accuracy is 0.99, 0.98, 0.36 and 100%. In conclusion, these models can be used as workable substitute to the existing methods of traffic flow prediction in Wireless Sensor Network.

Chapter 6

Conclusion and Future Scope

"The skill of writing is to create a context in which other people can think."

-Edwin Schlossberg

6.1 Conclusions

Dependability is a key factor for the measurement of Quality of Service in Wireless Sensor Networks. The network needs to be regulated based on certain characteristics, discussed in the course of this work. Further, the negative and positive traits need to be taken care of while considering network scalability and adaptability. This thesis is organized in three correlative sections, all of which can be either used separately or unitedly. The main benefit of these sections is: asserting the run-time reliability of the network; enhancing the dependability of network under flooding attack; and intelligent traffic flow prediction. This chapter concludes the thesis and discusses the directions of future work:

Reliability Prediction Framework: Reliability is defined as the capability of a network to perform its intended task under certain conditions for a stated timespan. There are many tools for modeling and analyzing the reliability of a network. As the intricacy of various networks is increasing, there is a need for many sophisticated methods for reliability analysis. The term reliability is used as an umbrella term to capture various attributes such as safety, availability, security, and ease of use. The existing methods have

many shortcomings which include inadequacy of a novel framework and inefficacy to handle scalable networks. This thesis presents a novel framework that predicts the overall reliability of the networks in terms of performance metrics such as, sent packets, received packets, packets forfeit, packet delivery ratio and throughput.

Security based Dependability Enhancement: The most imperative aspect of a dependable network is its security. This thesis considers the flooding attack in which, an attacker repeatedly sends packets at higher rates, thereby causing packet drop and exhaustion of the link capacity resulting in communication failure. To detect this attack, an Intrusion Detection System based on the randomized and the normalized deployment of nodes is proposed. Further, Machine Learning techniques are implemented to enhance the dependability of Wireless Sensor Network under flooding attack. The data flow is a significant parameter for governing the flooding effect on the network. It is found that Machine Learning models play a significant role in the prediction of the data flow, which is related to the packet delivery ratio of the network.

Intelligent Neural Network based Traffic Prediction: Further, leveraging the benefit of neural networks, Machine Learning models based on the concept of Neural Networks are developed to accurately forecast traffic flow. Various intelligent techniques, namely Neural Network, Bayesian Regularized Neural Networks, Neural Networks using Model Averaging, Multi-Layer Perceptron, Multi-Layer Perceptron with Multiple Layers, Quantile Regression Neural Network and Stacked Autoencoder Deep Neural Network are analyzed. The models are thus designed and tested. Based on the experiments performed on the first-hand data obtained using simulations, it is observed that Quantile Regression Neural Network and Multi-Layer Perceptron with Multiple Layers models outperformed in terms of Accuracy and Root Mean Square Error.

6.2 Future Scope

Research is an iterative and continuous procedure. The work presented in the thesis focuses on solving the prediction problems of data flow in Wireless Sensor Networks. There are several directions in which this research work could be expanded. Some of the suggestions for future work are as follows:

1. This thesis explores some of the Machine Learning models. Several other models are available and can be explored for further improvement in prediction performance.

2. The two dependability attributes have been addressed in this thesis, Reliability, and Security. Left-out attributes such as Trust, Safety, Survivability, Fault Detection and Tolerance can be considered as an extension of this work.

3. Various other techniques for ensemble designing can also be explored.

4. Artificial Intelligence and Machine Learning have reached a critical tipping point and will increasingly augment and extend virtually every technology-enabled service, thing, or application. This research can be extended to some major industries such as:

    - Drug Discovery
    - Healthcare
    - Social Network Analysis
    - Financial Trading

The limitations of the study are described as characteristics of methodology that influenced the main findings of this research work. The main limitation of this work is that it is based upon simulations. It would be very interesting to perform this experiment on the real test beds. The dataset thus collected will provide much better results in terms of quality measures, reliability and security.

References

[1] Đoko Banđur, Branimir Jakšić, Miloš Banđur, and Srđan Jović. An analysis of energy efficiency in wireless sensor networks (wsns) applied in smart agriculture. *Computers and Electronics in Agriculture*, 156:500–507, 2019.

[2] Carol Habib, Abdallah Makhoul, Rony Darazi, and Raphaël Couturier. Health risk assessment and decision-making for patient monitoring and decision-support using wireless body sensor networks. *Information Fusion*, 47:10–22, 2019.

[3] DG Reina, Mohamed Askalani, SL Toral, Federico Barrero, Eleana Asimakopoulou, and Nik Bessis. A survey on multihop ad hoc networks for disaster response scenarios. *International Journal of Distributed Sensor Networks*, 11(10):647037, 2015.

[4] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[5] R Chellappa Doss, A Jennings, and N Shenoy. A review of current mobility prediction techniques for ad hoc networks. In *The Fourth IASTED International Multi-Conference, Banff, Canada*, pages 536–542, 2004.

[6] Jasminder Kaur Sandhu and Sharad Saxena. Data agglomeration in wireless sensor networks. In *Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference on*, pages 285–289. IEEE, 2014.

[7] Yahiya Gazi, M Sarosh Umar, and Mohd Sadiq. Classification of nfrs for information system. *International Journal of Computer Applications*, 115(22), 2015.

[8] Mohammed Zaki Hasan, Fadi Al-Turjman, and Hussain Al-Rizzo. Analysis of cross-layer design of quality-of-service forward geographic wireless sensor

network routing strategies in green internet of things. *IEEE Access*, 6: 20371–20389, 2018.

[9] Satyabrata Chakrabarti and Amitabh Mishra. Qos issues in ad hoc wireless networks. *IEEE communications magazine*, 39(2):142–148, 2001.

[10] Jacques Bahi, Wiem Elghazel, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni. Reliable diagnostics using wireless sensor networks. *Computers in Industry*, 104:103–115, 2019.

[11] Mohamed Al-Kuwaiti, Nicholas Kyriakopoulos, and Sayed Hussein. A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability. *Communications Surveys & Tutorials, IEEE*, 11(2):106–124, 2009.

[12] DC Mishra, RK Sharma, Mayank Dawar, and M Hanmandlu. Two layers of security for color video by matrix affine cipher with two-dimensional discrete wavelet transform. *Fractals*, 23(04):1550037, 2015.

[13] Boudewijn R Haverkort, Holger Hermanns, and J-P Katoen. On the use of model checking techniques for dependability evaluation. In *Reliable Distributed Systems, 2000. SRDS-2000. Proceedings The 19th IEEE Symposium on*, pages 228–237. IEEE, 2000.

[14] Ivanovitch Silva, Luiz Affonso Guedes, Paulo Portugal, and Francisco Vasques. Reliability and availability evaluation of wireless sensor networks for industrial applications. *Sensors*, 12(1):806–838, 2012.

[15] Felix Salfner, Maren Lenk, and Miroslaw Malek. A survey of online failure prediction methods. *ACM Computing Surveys (CSUR)*, 42(3):10, 2010.

[16] Shui Yu, Wanlei Zhou, Robin Doss, and Weijia Jia. Traceback of ddos attacks using entropy variations. *IEEE Transactions on Parallel and Distributed Systems*, 22(3): 412–425, 2011.

[17] D Praveen Kumar, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion*, 49:1–25, 2019.

[18] Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, and Hwee-Pink Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[19] Pedro Larranaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, José A Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, 7(1):86–112, 2006.

[20] Jarosław Rzeszótko and Sinh Hoa Nguyen. Machine learning for traffic prediction. *Fundamenta Informaticae*, 119(3-4):407–420, 2012.

[21] Girik Pachauri and Sandeep Sharma. Anomaly detection in medical wireless sensor networks using machine learning algorithms. *Procedia Computer Science*, 70: 325–333, 2015.

[22] Kaveri Kadam and Navin Srivastava. Application of machine learning (reinforcement learning) for routing in wireless sensor networks (wsns). In *Physics and Technology of Sensors (ISPTS), 2012 1st International Symposium on*, pages 349–352. IEEE, 2012.

[23] Edwin Prem Kumar Gilbert, M Lydia, K Baskaran, and Elijah Blessing Rajsingh. Trust aware fault tolerant prediction model for wireless sensor network based measurements in smart grid environment. *Sustainable Computing: Informatics and Systems*, 23:29–37, 2019.

[24] Tayyab Khan, Karan Singh, Mohamed Abdel-Basset, Hoang Viet Long, Satya P Singh, Manisha Manjul, et al. A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks. *IEEE Access*, 7:58221–58240, 2019.

[25] Michael Akerele, Irfan Al-Anbagi, and Melike Erol-Kantarci. A fiber-wireless sensor networks qos mechanism for smart grid applications. *IEEE Access*, 7: 37601–37610, 2019.

[26] Tong Zhang, Lisha Yan, and Yuan Yang. Trust evaluation method for clustered wireless sensor networks based on cloud model. *Wireless Networks*, 24(3): 777–797, 2018.

[27] João Almeida, Joaquim Ferreira, and Arnaldo SR Oliveira. A medium guardian for enhanced dependability in safety-critical wireless systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):965–976, 2018.

[28] Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. A novel framework for reliable network prediction of small scale wireless sensor networks (sswsns). *Fundamenta Informaticae*, 160(3):303–341, 2018.

[29] Zeyu Zhang, Amjad Mehmood, Lei Shu, Zhiqiang Huo, Yu Zhang, and Mithun Mukherjee. A survey on fault diagnosis in wireless sensor networks. *IEEE Access*, 6:11349–11364, 2018.

[30] Adam B Noel, Abderrazak Abdaoui, Tarek Elfouly, Mohamed Hossam Ahmed, Ahmed Badawy, and Mohamed S Shehata. Structural health monitoring using wireless sensor networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 19(3):1403–1423, 2017.

[31] Huang-Chen Lee, Kai-Hsiang Ke, Yao-Min Fang, Bing-Jean Lee, and Teng-Chieh Chan. Open-source wireless sensor system for long-term monitoring of slope movement. *IEEE Trans. Instrumentation and Measurement*, 66(4):767–776, 2017.

[32] Dingwen Yuan, Salil S Kanhere, and Matthias Hollick. Instrumenting wireless sensor networks—a survey on the metrics that matter. *Pervasive and Mobile Computing*, 37:45–62, 2017.

[33] Jó Ueyama, Bruno S Faiçal, Leandro Y Mano, Guilherme Bayer, Gustavo Pessin, and Pedro H Gomes. Enhancing reliability in wireless sensor networks for adaptive river monitoring systems: Reflections on their long-term deployment in brazil. *Computers, Environment and Urban Systems*, 65:41–52, 2017.

[34] Tamoghna Ojha, Sudip Misra, and Narendra Singh Raghuwanshi. Sensing-cloud: Leveraging the benefits for agricultural applications. *Computers and electronics in agriculture*, 135:96–107, 2017.

[35] Amir Ehsani Zonouz, Liudong Xing, Vinod M Vokkarane, and Yan Sun. Hybrid wireless sensor networks: a reliability, cost and energy-aware approach. *IET Wireless Sensor Systems*, 6(2):42–48, 2016.

[36] Wiem Elghazel, Jacques Bahi, Christophe Guyeux, Mourad Hakem, Kamal Medjaher, and Noureddine Zerhouni. Dependability of wireless sensor networks for industrial prognostics and health management. *Computers in Industry*, 68:1–15, 2015.

[37] Muhammad Adeel Mahmood, Winston KG Seah, and Ian Welch. Reliability in wireless sensor networks: A survey and challenges ahead. *Computer Networks*, 79:166–187, 2015.

[38] Arslan Munir, Joseph Antoon, and Ann Gordon-Ross. Modeling and analysis of fault detection and fault tolerance in wireless sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 14(1):3, 2015.

[39] Guangjie Han, Jinfang Jiang, Lei Shu, Jianwei Niu, and Han-Chieh Chao. Management and applications of trust in wireless sensor networks: A survey. *Journal of Computer and System Sciences*, 80(3):602–617, 2014.

[40] Antônio Dâmaso, Nelson Rosa, and Paulo Maciel. Reliability of wireless sensor networks. *Sensors*, 14(9):15760–15785, 2014.

[41] Joseph E Mbowe and George S Oreku. Quality of service in wireless sensor networks. *Wireless Sensor Network*, 6(02):19, 2014.

[42] B Silva, G Callou, E Tavares, P Maciel, J Figueiredo, E Sousa, C Araujo, F Magnani, and F Neves. Astro: An integrated environment for dependability and sustainability evaluation. *Sustainable computing: informatics and systems*, 3 (1):1–17, 2013.

[43] Simona Bernardi, José Merseguer, and Dorina C Petriu. *Model-driven dependability assessment of software systems*. Springer, 2013.

[44] Waseem Ahmed and Yong Wei Wu. A survey on reliability in distributed systems. *Journal of Computer and System Sciences*, 79(8):1243–1255, 2013.

[45] Sazia Parvin, Farookh Khadeer Hussain, Jong Sou Park, and Dong Seong Kim. A survivability model in wireless sensor networks. *Computers & Mathematics with Applications*, 64(12):3666–3682, 2012.

[46] Yongxian Song, Ting Chen, Juanli Ma, Yuan Feng, and Xianjin Zhang. Design and analysis for reliability of wireless sensor network. *JNW*, 7(12):2003–2010, 2012.

[47] Jaime Chen, Manuel Díaz, Luis Llopis, Bartolomé Rubio, and José M Troya. A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection. *Journal of Network and Computer Applications*, 34(4):1225–1239, 2011.

[48] Yves Langeron, Anne Barros, Antoine Grall, and Christophe Bérenguer. Dependability assessment of network-based safety-related system. *Journal of Loss Prevention in the Process Industries*, 24(5):622–631, 2011.

[49] Yung-Ruei Chang, Chin-Yu Huang, and Sy-Yen Kuo. Performance assessment and reliability analysis of dependable and distributed computing systems based on bdd and recursive merge. *Applied Mathematics and Computation*, 217(1):403–413, 2010.

[50] James PG Sterbenz, David Hutchison, Egemen K Çetinkaya, Abdul Jabbar, Justin P Rohrer, Marcus Schöller, and Paul Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.

[51] Zhengyi Le, Eric Becker, Dimitrios G Konstantinides, Chirs Ding, and Fillia Makedon. Modeling reliability for wireless sensor node coverage in assistive testbeds. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, page 46. ACM, 2010.

[52] Dario Bruneo, Antonio Puliafito, and Marco Scarpa. Dependability analysis of wireless sensor networks with active-sleep cycles and redundant nodes. In *Proceedings of the First Workshop on DYnamic Aspects in DEpendability Models for Fault-Tolerant Systems*, pages 25–30. ACM, 2010.

[53] Valeria Cardellini, Emiliano Casalicchio, Vincenzo Grassi, Francesco Lo Presti, and Raffaela Mirandola. Towards self-adaptation for dependable service-oriented systems. In *Architecting Dependable Systems VI*, pages 24–48. Springer, 2009.

[54] Lihua Xu, Hadar Ziv, Thomas A Alspaugh, and Debra J Richardson. An architectural pattern for non-functional dependability requirements. *Journal of Systems and Software*, 79(10):1370–1378, 2006.

[55] Lance Doherty and Dana A Teasdale. Towards 100% reliability in wireless monitoring networks. In *Proceedings of the 3rd ACM international workshop on*

Performance evaluation of wireless ad hoc, sensor and ubiquitous networks, pages 132–135. ACM, 2006.

[56] Amirhosein Taherkordi, M Alkaee Taleghan, and Mohsen Sharifi. Dependability considerations in wireless sensor networks applications. *Journal of Networks*, 1(6): 28–35, 2006.

[57] Michael G Hinchey, James L Rash, Christopher A Rouff, and Denis Gračanin. Achieving dependability in sensor networks through automated requirements-based programming. *Computer Communications*, 29(2):246–256, 2006.

[58] Cláudia Betous-Almeida and Karama Kanoun. Construction and stepwise refinement of dependability models. *Performance Evaluation*, 56(1-4):277–306, 2004.

[59] Mohamed Kaâniche, Jean-Claude Laprie, and Jean-Paul Blanquart. A framework for dependability engineering of critical computing systems. *Safety Science*, 40(9): 731–752, 2002.

[60] Walter J Gutjahr. Software dependability evaluation based on markov usage models. *Performance evaluation*, 40(4):199–222, 2000.

[61] Agapios Platis, Nikolaos Limnios, and Marc Le Du. Dependability analysis of systems modeled by non-homogeneous markov chains. *Reliability Engineering & System Safety*, 61(3):235–249, 1998.

[62] Manish Malhotra and Kishor S Trivedi. Dependability modeling using petri-nets. *IEEE Transactions on reliability*, 44(3):428–440, 1995.

[63] Noe Lopez-Benitez. Dependability analysis of distributed computing systems using stochastic petri nets. In *Reliable Distributed Systems, 1992. Proceedings., 11th Symposium on*, pages 85–92. IEEE, 1992.

[64] Abdulaziz Alarifi and Amr Tolba. Optimizing the network energy of cloud assisted internet of things by using the adaptive neural learning approach in wireless sensor networks. *Computers in Industry*, 106:133–141, 2019.

[65] Songyut Phoemphon, Chakchai So-In, and Dusit Tao Niyato. A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm

optimization for wireless sensor network localization. *Applied Soft Computing*, 65:101–120, 2018.

[66] Lior Turgeman, Jerrold H May, and Roberta Sciulli. Insights from a machine learning model for predicting the hospital length of stay (los) at the time of admission. *Expert Systems with Applications*, 78:376–385, 2017.

[67] Julian Hagenauer and Marco Helbich. A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications*, 78: 273–282, 2017.

[68] Luis M Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017.

[69] Ke Hu, Ashfaqur Rahman, Hari Bhrugubanda, and Vijay Sivaraman. Hazeest: Machine learning based metropolitan air pollution estimation from fixed and mobile sensors. *IEEE Sensors Journal*, 17(11):3517–3525, 2017.

[70] Michael E Cholette, Pietro Borghesani, Egidio Di Gialleonardo, and Francesco Braghin. Using support vector machines for the computationally efficient identification of acceptable design parameters in computer-aided engineering applications. *Expert Systems with Applications*, 81:39–52, 2017.

[71] Kai Cheng, Zhenzhou Lu, Yuhao Wei, Yan Shi, and Yicheng Zhou. Mixed kernel function support vector regression for global sensitivity analysis. *Mechanical Systems and Signal Processing*, 96:201–214, 2017.

[72] Bin Weng, Mohamed A Ahmed, and Fadel M Megahed. Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, 79:153–163, 2017.

[73] Santosh Singh Rathore and Sandeep Kumar. Towards an ensemble based system for predicting the number of software faults. *Expert Systems with Applications*, 82: 357–382, 2017.

[74] Mohammad Ehsanul Karim, John Petkau, Paul Gustafson, Helen Tremlett, and The Beams Study Group. On the application of statistical learning approaches to construct inverse probability weights in marginal structural cox models: Hedging

against weight-model misspecification. *Communications in Statistics-Simulation and Computation*, pages 1–30, 2017.

[75] Lingjian Yang, Songsong Liu, Sophia Tsoka, and Lazaros G Papageorgiou. A regression tree approach using mathematical programming. *Expert Systems with Applications*, 78:347–357, 2017.

[76] Abbas Ali Rezaee and Seyyed Ehsan Golparvar. Conditional inference tree modelling of competing motivators of the positioning of concessive clauses: The case of a non-native corpus. 2017.

[77] Jidong J Yang and Bashan Zuo. Performance of smart sensor detectors for stop-bar detection at signalized intersections. *Journal of Transportation Engineering, Part A: Systems*, 143(6):04017020, 2017.

[78] Brandon Heung, Matúš Hodúl, and Margaret G Schmidt. Comparing the use of training data derived from legacy soil pits and soil survey polygons for mapping soil classes. *Geoderma*, 290:51–68, 2017.

[79] Isabel Pôças, João Gonçalves, Patrícia Malva Costa, Igor Gonçalves, Luís S Pereira, and Mario Cunha. Hyperspectral-based predictive modelling of grapevine water status in the portuguese douro wine region. *International Journal of Applied Earth Observation and Geoinformation*, 58:177–190, 2017.

[80] Jonathan L Ticknor. A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, 40(14):5501–5506, 2013.

[81] Sabina Smusz, Rafał Kurczab, and Andrzej J Bojarski. A multidimensional analysis of machine learning methods performance in the classification of bioactive compounds. *Chemometrics and Intelligent Laboratory Systems*, 128:89–100, 2013.

[82] Hui-Yi Lin, Y Ann Chen, Ya-Yu Tsai, Xiaotao Qu, Tung-Sung Tseng, and Jong Y Park. Trm: A powerful two-stage machine learning approach for identifying snp-snp interactions. *Annals of human genetics*, 76(1):53–62, 2012.

[83] Chih-Chia Yao, Pao-Ta Yu, and Ruo-Wei Hung. Extractive support vector algorithm on support vector machines for image restoration. *Fundamenta Informaticae*, 90(1-2):171–190, 2009.

[84] Simone Borra and Agostino Di Ciaccio. Improving nonparametric regression methods by bagging and boosting. *Computational Statistics & Data Analysis*, 38 (4):407–420, 2002.

[85] Mianxiong Dong, Kaoru Ota, Anfeng Liu, and Minyi Guo. Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 27(1):225–236, 2016.

[86] Philipp Limbourg. *Dependability modelling under uncertainty*. Springer, 2008.

[87] Sharad Sharma, Shakti Kumar, and Brahmjit Singh. Routing in wireless mesh networks: Three new nature inspired approaches. *Wireless Personal Communications*, 83(4):3157–3179, 2015.

[88] F Richard Yu, Bo Sun, Vikram Krishnamurthy, and Saqib Ali. Application layer qos optimization for multimedia transmission over cognitive radio networks. *Wireless Networks*, 17(2):371–383, 2011.

[89] Xu Xu, Weifa Liang, and Tim Wark. Data quality maximization in sensor networks with a mobile sink. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–8. IEEE, 2011.

[90] Chien-Erh Weng and Tsung-Wen Lai. An energy-efficient routing algorithm based on relative identification and direction for wireless sensor networks. *Wireless personal communications*, 69(1):253–268, 2013.

[91] Teerawat Issariyakul and Ekram Hossain. *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.

[92] Xiaohua Tian, Jiaqi Liu, and Xinbing Wang. Streamline architecture of network simulator to facilitate teaching of computer networking. In *Proceedings of the ACM Turing 50th Celebration Conference-China*, page 2. ACM, 2017.

[93] Manon Caubet, Mercedes Román Dobarco, Dominique Arrouays, Budiman Minasny, and Nicolas PA Saby. Merging country, continental and global predictions of soil texture: Lessons from ensemble modelling in france. *Geoderma*, 337:99–110, 2019.

[94] A Liaw and M Wiener. Classification and Regression by randomForest. *R News*, 2 (3):18–22, 2002. URL `http://cran.r-project.org/doc/Rnews`.

[95] SS Keerthi and EG Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46(1):351–360, 2002.

[96] M Riedmiller and H Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE Int Conf on Neural Nets*, pages 586–591, 1993.

[97] Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, Ian H Witten, and Len Trigg. Weka. In *Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. Springer, 2005.

[98] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[99] Frank R Burden and David A Winkler. Robust qsar models using bayesian regularized neural networks. *Journal of medicinal chemistry*, 42(16):3183–3187, 1999.

[100] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[101] Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. evtree: Evolutionary learning of globally optimal classification and regression trees in r. Technical report, Working Papers in Economics and Statistics, 2011.

[102] Mohamed Amine Kafi, Jalel Ben Othman, and Nadjib Badache. A survey on reliability protocols in wireless sensor networks. *ACM Computing Surveys (CSUR)*, 50(2):31, 2017.

[103] Valério Rosset, Matheus A Paulo, Juliana G Cespedes, and Mariá CV Nascimento. Enhancing the reliability on data delivery and energy efficiency by combining swarm intelligence and community detection in large-scale wsns. *Expert Systems with Applications*, 78:89–102, 2017.

[104] Rakesh Ranjan Swain and Pabitra Mohan Khilar. Composite fault diagnosis in wireless sensor networks using neural networks. *Wireless Personal Communications*, 95(3):2507–2548, 2017.

[105] Kishor Shridharbhai Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. PHI, 2009.

[106] JF Mas. Receiver operating characteristic (roc) analysis. In *Geomatic Approaches for Modeling Land Change Scenarios*, pages 465–467. Springer, 2018.

[107] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC bioinformatics*, 12(1):77, 2011.

[108] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[109] AM Baker, FC Hsu, and FS Gayzik. A method to measure predictive ability of an injury risk curve using an observation-adjusted area under the receiver operating characteristic curve. *Journal of Biomechanics*, 2018.

[110] Fei Hu, Waqaas Siddiqui, and Krishna Sankar. Scalable security in wireless sensor and actuator networks (wsans). *Security in Sensor Networks*, page 177, 2016.

[111] Wazir Zada Khan, Md Shohrab Hossain, Mohammed Y Aalsalem, NM Saad, and Mohammed Atiquzzaman. A cost analysis framework for claimer reporter witness based clone detection schemes in wsns. *Journal of Network and Computer Applications*, 63:68–85, 2016.

[112] Chien-Erh Weng, Vishal Sharma, Hsing-Chung Chen, and Chuan-Hsien Mao. Peer: Proximity-based energy-efficient routing algorithm for wireless sensor networks. *J. Internet Serv. Inf. Secur.*, 6(1):47–56, 2016.

[113] Abderrahmane Baadache and Ali Belmehdi. Fighting against packet dropping misbehavior in multi-hop wireless ad hoc networks. *Journal of Network and Computer Applications*, 35(3):1130–1139, 2012.

[114] Siqi Ma, David Lo, and Ning Xi. Collaborative'many to many'ddos detection in cloud. *International Journal of Ad Hoc and Ubiquitous Computing*, 23(3-4): 192–202, 2016.

[115] Sabrina Sicari, Alessandra Rizzardi, Daniele Miorandi, and Alberto Coen-Porisini. Reato: Reacting to denial of service attacks in the internet of things. *Computer Networks*, 137:37–48, 2018.

[116] Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5): 643–666, 2004.

[117] Shui Yu, Wanlei Zhou, and Robin Doss. Information theory based detection against network behavior mimicking ddos attacks. *IEEE Communications Letters*, 12(4), 2008.

[118] Alexey G Finogeev and Anton A Finogeev. Information attacks and security in wireless sensor networks of industrial scada systems. *Journal of Industrial Information Integration*, 5:6–16, 2017.

[119] Bounpadith Kannhavong, Hidehisa Nakayama, Yoshiaki Nemoto, Nei Kato, and Abbas Jamalipour. A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless communications*, 14(5), 2007.

[120] Bharat Bhushan and Gadadhar Sahoo. Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks. *Wireless Personal Communications*, 98(2):2037–2077, 2018.

[121] Saqib Ali, Taiseera Al Balushi, Zia Nadir, and Omar Khadeer Hussain. Wsn security mechanisms for cps. In *Cyber Security for Cyber Physical Systems*, pages 65–87. Springer, 2018.

[122] Vishal Sharma and Rajesh Kumar. Three-tier neural model for service provisioning over collaborative flying ad hoc networks. *Neural Computing and Applications*, 29 (10):837–856, 2018.

[123] Jean-Claude Laprie. Dependability: Basic concepts and terminology. In *Dependability: Basic Concepts and Terminology*, pages 3–245. Springer, 1992.

[124] Vishal Sharma, Roberto Sabatini, and Subramanian Ramasamy. Uavs assisted delay optimization in heterogeneous wireless networks. *IEEE Communications Letters*, 20(12):2526–2529, 2016.

[125] Yanli Yu, Keqiu Li, Wanlei Zhou, and Ping Li. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and computer Applications*, 35(3):867–880, 2012.

[126] Vishal Sharma, Rajesh Kumar, and Prashant Singh Rana. Self-healing neural model for stabilization against failures over networked uavs. *IEEE Communications Letters*, 19(11):2013–2016, 2015.

[127] Jasminder Kaur Sandhu and Sharad Saxena. Procuring wireless sensor actuator network security. In *Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing*, pages 905–916. Springer, 2016.

[128] Mustapha Reda Senouci, Abdelhamid Mellouk, and Amar Aissani. Random deployment of wireless sensor networks: a survey and approach. *International Journal of Ad Hoc and Ubiquitous Computing*, 15(1-3):133–146, 2014.

[129] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 80–91. ACM, 2002.

[130] Jun Zheng and Abbas Jamalipour. *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.

[131] Jérôme François, Issam Aib, and Raouf Boutaba. Firecol: a collaborative protection network for the detection of flooding ddos attacks. *IEEE/ACM Transactions on Networking (TON)*, 20(6):1828–1841, 2012.

[132] Dai Jianjian, Tao Yang, and Yang Feiyue. A novel intrusion detection system based on iabrbfsvm for wireless sensor networks. *Procedia Computer Science*, 131:1113–1121, 2018.

[133] Tarfa Hamed, Jason B Ernst, and Stefan C Kremer. A survey and taxonomy on data and pre-processing techniques of intrusion detection systems. In *Computer and Network Security Essentials*, pages 113–134. Springer, 2018.

[134] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

[135] Rodrigo Roman, Jianying Zhou, and Javier Lopez. Applying intrusion detection systems to wireless sensor networks. In *IEEE Consumer Communications & Networking Conference (CCNC 2006)*, 2006.

[136] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.

[137] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 1–13. ACM, 2003.

[138] Vikas Mittal, Sunil Gupta, and Tanupriya Choudhury. Comparative analysis of authentication and access control protocols against malicious attacks in wireless sensor networks. In *Smart Computing and Informatics*, pages 255–262. Springer, 2018.

[139] Swagata Biswas, Ria Das, and Punyasha Chatterjee. Energy-efficient connected target coverage in multi-hop wireless sensor networks. In *Industry Interactive Innovations in Science, Engineering and Technology*, pages 411–421. Springer, 2018.

[140] Qussai Yaseen, Firas Albalas, Yaser Jararwah, and Mahmoud Al-Ayyoub. Leveraging fog computing and software defined systems for selective forwarding attacks detection in mobile wireless sensor networks. *Transactions on Emerging Telecommunications Technologies*, 29(4):e3183, 2018.

[141] Ashfaq Hussain Farooqi and Farrukh Aslam Khan. Intrusion detection systems for wireless sensor networks: A survey. In *Communication and networking*, pages 234–241. Springer, 2009.

[142] Audun Josang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th bled electronic commerce conference*, volume 5, pages 2502–2511, 2002.

[143] Robert Mitchell and Ray Chen. A survey of intrusion detection in wireless network applications. *Computer Communications*, 42:1–23, 2014.

[144] Vinod Kumar Verma, Surinder Singh, and Nagendra P Pathak. Analysis of scalability for aodv routing protocol in wireless sensor networks. *Optik-International Journal for Light and Electron Optics*, 125(2):748–750, 2014.

[145] Carolina Del-Valle-Soto, Carlos Mex-Perera, Raul Monroy, and Juan Arturo Nolazco-Flores. On the routing protocol influence on the resilience of wireless sensor networks to jamming attacks. *Sensors*, 15(4):7619–7649, 2015.

[146] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[147] Junfeng Wang, Yiming Miao, Ping Zhou, M Shamim Hossain, and Sk Md Mizanur Rahman. A software defined network routing in wireless multihop network. *Journal of Network and Computer Applications*, 85:76–83, 2017.

[148] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks*, 2(1):1–22, 2004.

[149] Jangeun Jun and Mihail L Sichitiu. Mrp: Wireless mesh networks routing protocol. *Computer Communications*, 31(7):1413–1435, 2008.

[150] Amit Sarkar and T Senthil Murugan. Routing protocols for wireless sensor networks: What the literature says? *Alexandria Engineering Journal*, 55(4): 3173–3183, 2016.

[151] Constantinos Marios Angelopoulos, Sotiris Nikoletseas, Theofanis P Raptis, Christoforos Raptopoulos, and Filippos Vasilakis. Improving sensor network performance with wireless energy transfer. *International Journal of Ad Hoc and Ubiquitous Computing*, 20(3):159–171, 2015.

[152] Vishal Sharma, Fei Song, Ilsun You, and Mohammed Atiquzzaman. Energy efficient device discovery for reliable communication in 5g-based iot and bsns using unmanned aerial vehicles. *Journal of Network and Computer Applications*, 97:79–95, 2017.

[153] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[154] S. Weichwald, T. Fomina, B. Schölkopf, and M. Grosse-Wentrup. Optimal coding in biological and artificial neural networks. 2016. URL `http://arxiv.org/abs/1605.07094`.

[155] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*, volume 124. CRC press, 2016.

[156] Swathi Jamjala Narayanan, Rajen B Bhatt, and Boominathan Perumal. Improving the accuracy of fuzzy decision tree by direct back propagation with adaptive learning rate and momentum factor for user localization. *Procedia Computer Science*, 89:506–513, 2016.

[157] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct): 49–75, 2000.

[158] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems*, 27(4):809–821, 2016.

[159] Dong-sheng Liu and Shu-jiang Fan. A modified decision tree algorithm based on genetic algorithm for mobile user classification problem. *The Scientific World Journal*, 2014, 2014.

[160] Ricardo Oentoe, Kentaro Hasebe, and Masahiko Nomura. Permeability prediction using projection pursuit regression in the abadi field. 2016.

[161] Torsten Hothorn, Achim Zeileis, and Maintainer Torsten Hothorn. Package 'partykit'. 2016.

[162] Raj Jagannathan. A linear regression approach for determining explicit expressions for option prices for equity option pricing models with dependent volatility and return processes. *Journal of Mathematical Finance*, 6(02):303, 2016.

[163] Haitao Li. Research on prediction of traffic flow based on dynamic fuzzy neural networks. *Neural Computing and Applications*, 27(7):1969–1980, 2015.

[164] Sankhadeep Chatterjee, Sarbartha Sarkar, Sirshendu Hore, Nilanjan Dey, Amira S Ashour, and Valentina E Balas. Particle swarm optimization trained neural network for structural failure prediction of multistoried rc buildings. *Neural Computing and Applications*, 28(8):2005–2016, 2017.

[165] Noboru Takeichi. Arrival traffic scheduling performance subject to futuristic traffic statistics. In *AIAA Modeling and Simulation Technologies Conference*, page 1323, 2017.

[166] Hemmati Ehsan Sheikhan, Mansour. Transient chaotic neural network-based disjoint multipath routing for mobile ad-hoc networks. *Neural Computing and Applications*, 21(6):1403–1412, 2012.

[167] Kang Sanggil Lim, Yujin. Path management method using partially connected neural network in large-scale heterogeneous sensor network. *Neural Computing and Applications*, 21(8):1931–1936, 2012.

[168] Yalda Rajabzadeh, Amir Hossein Rezaie, and Hamidreza Amindavar. Short-term traffic flow prediction using time-varying vasicek model. *Transportation Research Part C: Emerging Technologies*, 74:168–181, 2017.

[169] Jie Cao, Zhiyi Fang, Guannan Qu, Hongyu Sun, and Dan Zhang. An accurate traffic classification model based on support vector machines. *International Journal of Network Management*, 2017.

[170] Anyu Cheng, Xiao Jiang, Yongfu Li, Chao Zhang, and Hao Zhu. Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method. *Physica A: Statistical Mechanics and its Applications*, 466:422–434, 2017.

[171] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Dynamic route planning with real-time traffic predictions. *Information Systems*, 64:258–265, 2017.

[172] Victoria J Hodge, Rajesh Krishnan, Jim Austin, John Polak, and Tom Jackson. Short-term prediction of traffic flow using a binary neural network. *Neural Computing and Applications*, 25(7-8):1639–1655, 2014.

[173] Ramin Yasdi. Prediction of road traffic using a neural network approach. *Neural computing & applications*, 8(2):135–142, 1999.

[174] Haitao Li. Research on prediction of traffic flow based on dynamic fuzzy neural networks. *Neural Computing and Applications*, 27(7):1969–1980, 2016.

[175] Johannes Klepsch, Claudia Klüppelberg, and Taoran Wei. Prediction of functional arma processes with an application to traffic data. *Econometrics and Statistics*, 1: 128–149, 2017.

[176] Wenbin Hu, Liping Yan, Huan Wang, Bo Du, and Dacheng Tao. Real-time traffic jams prediction inspired by biham, middleton and levine (bml) model. *Information Sciences*, 381:209–228, 2017.

[177] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[178] Yu Peng, Miao Lei, Jun-Bao Li, and Xi-Yuan Peng. A novel hybridization of echo state networks and multiplicative seasonal arima model for mobile communication traffic series forecasting. *Neural Computing and Applications*, 24(3-4):883–890, 2014.

[179] Javad Abdi, Behzad Moshiri, Baher Abdulhai, and Ali Khaki Sedigh. Short-term traffic flow forecasting: parametric and nonparametric approaches via emotional temporal difference learning. *Neural Computing and Applications*, 23(1):141–159, 2013.

[180] Dayashankar Singh, Maitreyee Dutta, and Sarvpal H Singh. Neural network based handwritten hindi character recognition. In *ACM International Conference (Compute 09)*, pages 9–10, 2009.

[181] Sharaf Alkheder, Madhar Taamneh, and Salah Taamneh. Severity prediction of traffic accident using an artificial neural network. *Journal of Forecasting*, 36(1): 100–108, 2017.

[182] Zhi Sun, Ying Chen, Xinyang Li, Xiaolin Qin, and Huiyong Wang. A bayesian regularized artificial neural network for adaptive optics forecasting. *Optics Communications*, 382:519–527, 2017.

[183] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[184] Sonali Singh, C Sudhakar Reddy, S Vazeed Pasha, Kalloli Dutta, KRL Saranya, and KV Satish. Modeling the spatial dynamics of deforestation and fragmentation using multi-layer perceptron neural network and landscape fragmentation tool. *Ecological Engineering*, 99:543–551, 2017.

[185] Umut Orhan, Mahmut Hekim, and Mahmut Ozer. Eeg signals classification using the k-means clustering and a multilayer perceptron neural network model. *Expert Systems with Applications*, 38(10):13475–13481, 2011.

[186] Alex J Cannon. Quantile regression neural networks: Implementation in r and application to precipitation downscaling. *Computers & Geosciences*, 37(9): 1277–1284, 2011.

[187] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[188] Víctor Díez, Aitor Arriola, Iñaki Val, and Manuel Velez. Reliability evaluation of point-to-point links based on ieee 802.15. 4 physical layer for iwsan applications. *AEU-International Journal of Electronics and Communications*, 113:152967, 2020.

[189] Vivek Deshpande and Vladimir Poulkov. Model to improve quality of service in wireless sensor network. In *Data Management, Analytics and Innovation*, pages 475–483. Springer, 2020.

List of Publications

Journals

1. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "A Novel Framework for Reliable Network Prediction of Small Scale Wireless Sensor Networks", **Fundamenta Informaticae, IOS Press**, **SCI-Indexed Journal**, http://dx.doi.org/10.3233/FI-2018-1685, **Published.**

2. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "Enhancing Dependability of Wireless Sensor Network under Flooding Attack: A Machine Learning Perspective", **International Journal of Ad-Hoc and Ubiquitous Computing, Inderscience**, **SCI-Indexed Journal**, **Published.**

3. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "Predicting Traffic Flow in Wireless Sensor Networks: A Neural Network Perspective", **Wireless Personal Communications, Springer**, **SCI-Indexed Journal**, **Under-Review.**

4. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "A Expert Approach for Data Flow Prediction: Case Study of Wireless Sensor Networks", **Wireless Personal Communications, Springer**, **SCI-Indexed Journal**, **Published.**

Conference Proceedings

1. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "A Data-Driven Framework for Survivable Wireless Sensor Networks" In Contemporary Computing (IC3), **11th International Conference on, pp. 1-6. IEEE, 2018, Acceptance Rate: 27.70%**

2. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "RCDR: Reliability Control Framework for Data Rate Prediction in Wireless Sensor Networks" In Computing, Power, and Communication (GUCON), **International Conference on, pp. 1-5. IEEE, 2018, Acceptance Rate: 31.59%**

Book Chapters

1. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "A Compendium of Security Issues in Wireless Sensor Networks." **In Computer and Cyber Security: Principles, Algorithms, Applications and Perspectives, pp. 1-20. Taylor and Francis Group, 2018.**

2. Jasminder Kaur Sandhu, Anil Kumar Verma, and Prashant Singh Rana. "Next Generation Adaptable Opportunistic Sensing Based Wireless Sensor Networks: A Machine Learning Perspective" **In Machine Learning for Computer and Cyber Security: Principles, Algorithms, and Practices, Taylor and Francis Group, 2019.**

Appendix A

Network Simulator (NS-2.35)

Network Simulator Version 2.35 (NS-2.35) is a discrete event network simulator with an inclination towards the object-oriented programming paradigm. It was developed at the UC Berkely and drafted in C++ and OTcl (Object Oriented Extension of Tool Command Language). It is very advantageous when simulating a local or wide area networking scenario. The user view of NS-2.35 is presented in Figure A.1:

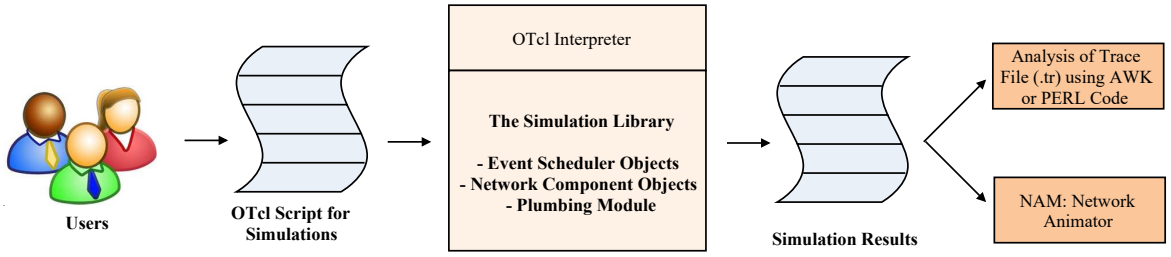


FIGURE A.1: User View of NS-2.35

The network simulation program is written in OTcl Script and the script is saved with a .tcl extension. The program is executed using a OTcl script interpreter. The interpreter contains the libraries namely, event scheduler object, network component objects, network setup helping modules (plumbing modules). The primary function of an event scheduler is to keep track of the simulation time and executes all events according to their current time. To execute an OTcl script, an event scheduler is initialized, network objects are

created in a particular topology, and the originator and destination object are identified to effective packets transmittal. The most important feature of NS is the plumbing module, and is used for the path establishment. This is done with the help of network objects or compound objects.

NS is written in OTcl and C++ both. The data path and control path implementations are segregated in NS. The network components and the event scheduler are written in C++, in order to save the event and packet processing time. OTcl linking links the compiled C++ objects with the matching OTcl object. This means the OTcl can control the C++ objects.

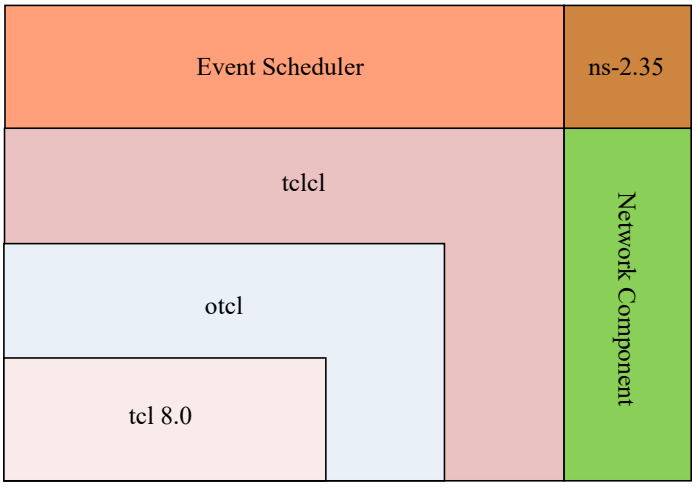


FIGURE A.2: Architecture View of NS-2.35

The user view of NS-2.35 is presented in Figure A.2. The user designs the network simulation scenario in Tcl using the OTcl library in the left bottom corner of this architecture. The network components and event scheduler is implemented in C++. These are linked using the Otcl linkage implemented with the help of tclcl. These modules combined together make a complete network design scenario in NS-2.35 with various linked libraries. Therefore, the main steps in implementing a NS-2.35 script include the steps: creating a network design and identifying the source and the destination node, writing a tcl script with various network objects and their corresponding linkages, executing the tcl script to get two resultant files, namely the Network Animator (NAM) for visualization and the trace file with extension .tr. The trace files is further processed using the AWK or PERL code to calculate the various network performance parameters such as delay, throughput, packets drop.

The simulation process of NS-2.35 includes: Simulation Design, Configuring and Executing Simulation, and the Post Simulation Interpretation. The Simulation Design designs the network scenario to be implemented. The user defines the simulation setup and also decides the performance parameter to be computed. The Configuring and Executing Simulation deals with the network components such as clock synchronization, and traffic flow. The Post Simulation refers to the process of debugging and interpretation of the final results.

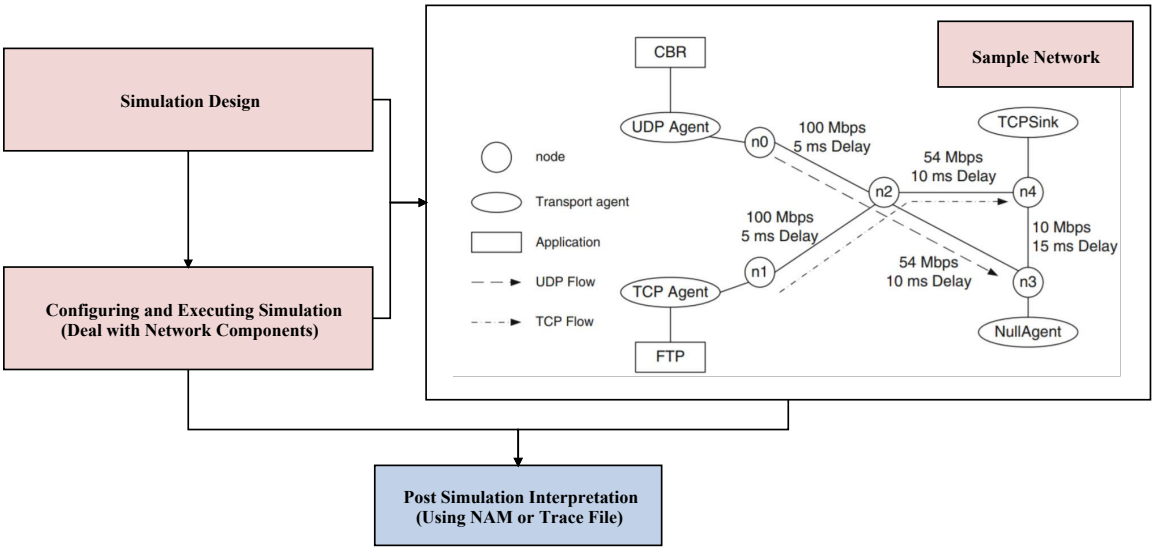


FIGURE A.3: The NS-2.35 Simulation Process

Appendix B

R Programming Language

R is a programming language which also provides a complete environment for statistical evaluations, pictorial representation and reporting. R was developed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand. It is currently being enhanced under the guidance of the R Development Core Team. The main advantage of R is that it is the most popular language among the Machine Learning experts and is freely available online. The main characteristics of R are:

- R is simple and effective language. It contains various constructs such as conditional, looping, functions, recursion, and arrays.
- Arrays, vectors, lists and matrices add to the basic functionalities of R.
- R provides a vast variety of tools for data analysis.
- The graphical visualization of R provides excellent platform for understanding the data more closely.

R provides an excellent collection of packages stored under the directory named as "library" in the R environment. Some of the packages are installed by default while installation of R. Other packages can be downloaded and installed online when required. R is supported by a Comprehensive R Archive Network (popularly known as CRAN). CRAN provides updated packages for R support. It is a network of web servers and ftp across the globe which store upgraded packages. This is the most advanced feature in R which enables the user to fetch latest updates and corresponding documentations.

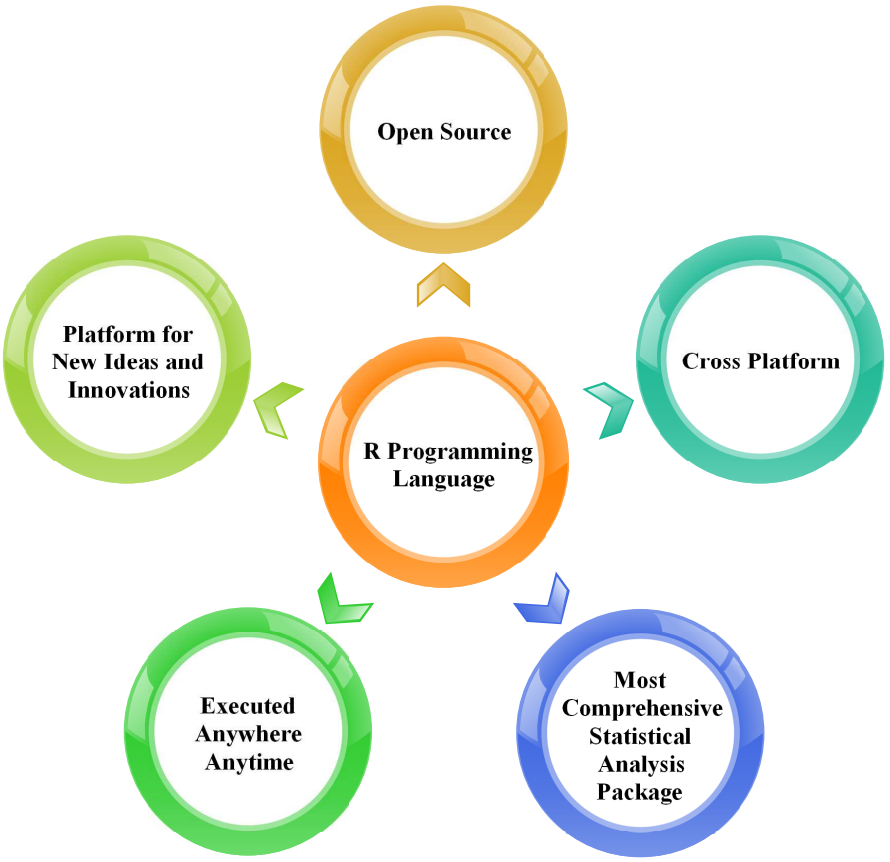


FIGURE B.1: The R Programming Language

The main advantages of using R is summed up in Figure B.1. R is an open-source software which can be used by any user anywhere and anytime. It provides the most comprehensive support for statistical analysis. It is cross-platform and can run on many operating systems. Also, new packages and bug support are provided online by many coders.

Further, the most attractive feature provided by R is the Data Wrangling. Data Wrangling is the process of cleaning large and complex datasets. R provides many packages for this data manipulation such as dplyr, readr, data.table package. The R programming language makes the execution of Machine Learning algorithms a lot more approachable and easy. There are many packages for Machine Learning such as caret, randomforest, party, and brnn.

The Machine Learning models used in the course of this research work are illustrated in Table B.1 below:

TABLE B.1: Machine Learning Models

Model Name	Method	Package
Cubist	cubist	Cubist
Random Forest	rf	randomForest
Support Vector Machine	svm	e1071
Neural Network	neuralnet	neuralnet
Weka Lazy Model	IBk	glm
Conditional Inference Tree	ctree	party
Bayesian Regularized Neural Network	brnn	brnn
Bagged MARS	bagEarth	earth
Bagged CART	bagging	ipred
Tree Model from Genetic Algorithm	evtree	evtree
Linear Model	lm	
Decision Tree	rpart	rpart
Extreme Learning Model	elm	elmNN
Generalized Additive Model	gam	mgcv
Model Tree	tree	tree
Projection Pursuit Regression	ppr	fRegression
PartyKit	ctree	party
Generalized Linear Model	glm	
Linear Regression	LinearRegression	LinearRegression
Extreme Learning Machine	elmtrain	elmNN
Multilayer Perceptron	mlp	caret
Multilayer Perceptron with Multiple Layers	mlpML	caret
Quantile Regression Neural Network	qrnn	caret
Deep Neural Network	dnn	deepnet

Appendix C

Basic Definitions

Denial-of-Service Attack: The denial-of-service attack makes a network non-functional and unavailable to the authenticated users. The main categories of this attack include: crashing and flooding. Crashing destabilize the network thereby making it unavailable to the users. Whereas, flooding occurs when the amount of traffic transmitted on the network exceeds the buffering capacity of the server. Buffer overflow is the most popular flooding attack on the network.

DSR: DSR is the Dynamic Source Routing protocol. It uses both uni-directional and bi-directional links. It uses source routing for transmitting the data through the network.

AODV: AODV is the Ad-Hoc On Demand Distance Vector routing protocol. It uses only bi-directional links. It supports the concept of multicasting and scalability. It creates the path between nodes on-demand i.e. only when the source node sends a request.

Intrusion Detection System: An Intrusion Detection System is a system capable of monitoring suspicious activities of the network and reporting or issuing alerts when required. It monitors the traffic traversing in the network. It also detects the system configuration error. Further, it is capable of accessing the integrity of directory and files on the system.

Failure Count: Failure count is the number of packets dropped by the intermediate node. It is represented by a positive integer.

Normalized Dataset: The normalized dataset contains the performance parameter values when the flooding attack has been mitigated and the co-ordinate values have been modified.

Normalized Distribution: When the sensor nodes are deployed using normal distribution, to reduce the effect of flooding attack on the network. Such a distribution is hence termed as normalized distribution.

Normalized Deployment: The efficient positioning of nodes using normalized distribution, results in the normalized deployment. The values of x and y-axis are computed after mitigating the effect of flooding attack from the network.

Random Deployment: The positioning of nodes on any random position corresponding to the x and y-axis, results in random deployment.

Structured Data: The data present in normalized dataset is known as structured data.

Threshold Value: The maximum permissible value of a parameter is known as the threshold value of that parameter.

Trust Level or Confidence Level: It is based upon the beta trust model and determines the trustworthiness of a sensor node. The beta trust model has performed well in the field of mobile networks and social networks. The trustworthiness of a particular sensor node is calculated on the basis of direct and indirect trust. The direct trust calculation is done in static environment, whereas, indirect trust is useful in mobile environment.

Un-Normalized or Un-Structured Dataset: The un-normalized data set contains the performance parameter values when the nodes are randomly deployed.

Machine Learning: It is an application of Artificial Intelligence. It is a type of learning in which machines learn on its own without being explicitly programmed. It is of two types: supervised and unsupervised learning. In a supervised learning model, labelled training data is provided as input. Some of the supervised learning techniques include: Support Vector Machine, Artificial Neural Networks, Logistic Regression, and Random Forest. Whereas, in unsupervised model, unlabelled training data is provided as input to extract the feature patterns. Some of the unsupervised learning techniques include: Principal Component Analysis, Clustering, and Auto-encoders.

Appendix D

Quantitative Evaluation

TABLE D.1: Correlation

Training-Testing →	**50-50**		**60-40**		**70-30**		**80-20**	
Machine Learning Models ↓	**UN**	**N**	**UN**	**N**	**UN**	**N**	**UN**	**N**
Linear Model	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96
Decision Tree	0.97	0.97	0.96	0.97	0.97	0.98	0.96	0.96
Extreme Learning Machine	0.41	0.72	0.77	0.83	0.27	0.83	0.75	0.83
Tree Models from Genetic Algorithms	0.93	0.93	0.92	0.92	0.94	0.93	0.93	0.92
Generalized Additive Model	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96
Model Tree	0.97	0.97	0.97	0.97	0.97	0.94	0.96	0.96
Projection Pursuit Regression	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96
Bayesian Regularized Neural Network	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96
PartyKit	0.99	0.96	0.97	0.97	1	0.97	0.99	0.96
Generalized Linear Model	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96
Linear Regression	0.97	0.96	0.96	0.96	0.97	0.96	0.97	0.96

UN = Un-Normalized Dataset; N = Normalized Dataset

TABLE D.2: Coefficient of Determination

Training-Testing →	50-50		60-40		70-30		80-20	
Machine Learning Models ↓	**UN**	**N**	**UN**	**N**	**UN**	**N**	**UN**	**N**
Linear Model	0.9	0.92	0.94	0.92	0.9	0.92	0.94	0.92
Decision Tree	0.92	0.94	0.94	0.94	0.94	0.96	0.92	0.92
Extreme Learning Machine	0.06	0.52	0.61	0.69	0.56	0.69	0.56	0.69
Tree Models from Genetic Algorithms	0.83	0.86	0.85	0.85	0.83	0.86	0.86	0.85
Generalized Additive Model	0.9	0.92	0.94	0.92	0.9	0.92	0.94	0.92
Model Tree	0.94	0.94	0.94	0.94	0.96	0.88	0.92	0.92
Projection Pursuit Regression	0.92	0.92	0.94	0.92	0.92	0.92	0.94	0.92
Bayesian Regularized Neural Network	0.92	0.92	0.94	0.92	0.92	0.92	0.94	0.92
PartyKit	0.94	0.92	0.98	0.94	0.98	0.94	0.98	0.92
Generalized Linear Model	0.92	0.92	0.94	0.92	0.92	0.92	0.94	0.92
Linear Regression	0.9	0.92	0.94	0.92	0.9	0.92	0.94	0.92

UN = Un-Normalized Dataset; N = Normalized Dataset

TABLE D.3: Root Mean Square Error

Training-Testing →	50-50		60-40		70-30		80-20	
Machine Learning Models ↓	**UN**	**N**	**UN**	**N**	**UN**	**N**	**UN**	**N**
Linear Model	0.46	0.45	0.5	0.48	0.51	0.43	0.43	0.45
Decision Tree	0.57	0.54	0.53	0.56	0.49	0.46	0.57	0.58
Extreme Learning Machine	2.3	1.65	1.45	1.29	1.53	1.22	1.56	1.31
Tree Models from Genetic Algorithms	0.87	0.83	0.81	0.82	0.85	0.83	0.87	0.87
Generalized Additive Model	0.46	0.45	0.5	0.48	0.51	0.43	0.43	0.45
Model Tree	0.5	0.54	0.53	0.56	0.46	0.65	0.57	0.58
Projection Pursuit Regression	0.45	0.44	0.5	0.49	0.49	0.44	0.43	0.46
Bayesian Regularized Neural Network	0.45	0.44	0.5	0.49	0.49	0.44	0.43	0.46
PartyKit	0.31	0.41	0.2	0.36	0.19	0.3	0.16	0.4
Generalized Linear Model	0.45	0.44	0.5	0.49	0.49	0.44	0.43	0.46
Linear Regression	0.46	0.45	0.5	0.48	0.51	0.43	0.43	0.45

UN = Un-Normalized Dataset; N = Normalized Dataset

TABLE D.4: Accuracy (Percentage)

Training-Testing →	50-50		60-40		70-30		80-20	
Machine Learning Models ↓	**UN**	**N**	**UN**	**N**	**UN**	**N**	**UN**	**N**
Linear Model	98.2	97	98.25	98	98.33	97.5	98.5	98
Decision Tree	99.8	99.25	99.5	100	99.33	99.5	100	98
Extreme Learning Machine	66	83	93.25	94.25	60.67	92.5	92.5	92.5
Tree Models from Genetic Algorithms	97.8	97.25	97.25	97.25	98.33	97	98.5	96.5
Generalized Additive Model	98.2	97	98.25	98	98.33	97.5	98.5	98
Model Tree	99.8	99.25	99.75	100	100	97	100	98
Projection Pursuit Regression	98.8	97	98.25	98.5	99.33	98	99	98
Bayesian Regularized Neural Network	99.6	97.75	100	99	100	100	100	98.5
PartyKit	100	98	99	99.75	100	98	99.5	99
Generalized Linear Model	48.8	43	40.75	47	42.67	47.5	40	45
Linear Regression	98.2	97	98.25	98	98.33	97.5	98.5	98

UN = Un-Normalized Dataset; N = Normalized Dataset

TABLE D.5: Time Taken (Seconds)

Training-Testing →	50-50		60-40		70-30		80-20	
Machine Learning Models ↓	**UN**	**N**	**UN**	**N**	**UN**	**N**	**UN**	**N**
Linear Model	51.4	27.58	36.97	56.03	29.69	34.55	43.05	47.78
Decision Tree	51.39	27.58	36.97	56.02	29.69	34.55	43.05	47.78
Extreme Learning Machine	51.4	27.6	36.98	56.05	29.72	34.57	43.08	47.8
Tree Models from Genetic Algorithms	51.45	27.61	36.98	56.05	29.72	34.57	43.08	47.8
Generalized Additive Model	51.46	27.63	37	56.06	29.75	34.6	43.11	47.91
Model Tree	51.46	27.63	37	56.06	29.75	34.6	43.11	47.91
Projection Pursuit Regression	51.46	27.63	37	56.08	29.76	34.61	43.11	47.91
Bayesian Regularized Neural Network	51.48	27.64	37.02	56.08	29.76	34.61	43.13	47.91
PartyKit	51.48	27.64	37.02	56.09	29.78	34.61	43.13	47.92
Generalized Linear Model	51.48	27.66	37.02	56.09	29.8	34.63	43.14	48.03
Linear Regression	51.5	27.66	37.03	56.11	29.8	34.64	43.14	48.03

UN = Un-Normalized Dataset; N = Normalized Dataset

TABLE D.6: Scaling of Results:

Scale	Rating	Abbreviation used
Excellent	5	EX
Very Good	4	VG
Good	3	GA
Average	2	AV
Poor	1	PR
Not Recommended	0	NR

TABLE D.7: Suggested Recommendations for Normalized Dataset are:

Training Testing Dataset	Strongly Recommended	Recommended	Average Recommendation	Not Recommended
50-50	Decision Tree, Model Tree	Linear, GAM, PPR, BRNN, PartyKit, Linear Regression	TMGA, GLM	ELM
60-40	PartyKit	Decision Tree, Model Tree, BRNN	Linear, GAM, Linear Regression	ELM, TMGA, GLM
70-30	Decision Tree, BRNN, PartyKit	Linear, GAM, PPR, Linear Regression	TMGA, Model Tree, GLM	ELM
80-20	PartyKit	Linear, GAM, PPR, BRNN, Linear Regression	Decision Tree, Model Tree	ELM, TMGA, GLM