ISBN: 978-93-88901-91-8

RESEARCH TRENDS IN COMPUTER SCIENCE, MATHEMATICAL SCIENCE AND INFORMATION TECHNOLOGY

EDITORS:

MRS. RIMPI RANI

MR. SOMNATH S. SANAP

MR. APURAV R. GUPTA

MR. SUBHARUN PAL

Bhumi Publishing

Bhumi Publishing, India

First Edition: September 2023

Research Trends in Computer Science, Mathematical Science and Information Technology

(ISBN: 978-93-88901-91-8)

Editors

Mrs. Rimpi Rani

TPD Malwa College,
Punjabi University,
Rampura Phul,
Dist. Bhatinda (Punjab)

Mr. Somnath S. Sanap

Department of Mathematics,
A.V College of Arts, K.M College of
Commerce and E. S. A. College of Science,
Vasai Road, Palghar, M.S.

Mr. Apurav R. Gupta

Department of Mathematics,
Late Pundalikrao Gawali Arts and
Science Mahavidyalaya,
Shirpur (Jain), M.S.

Mr. Subharun Pal

Swiss School of Management (SSM),
Switzerland and
European International University (EIU),
France

Bhumi Publishing

September 2023

First Edition: September, 2023

ISBN: 978-93-88901-91-8

ISBN 978-93-88901-91-8

9 789388 901918

Published by:

Bhumi Publishing,
Nigave Khalasa, Kolhapur 416207, Maharashtra, India
Website: www.bhumipublishing.com
E-mail: bhumipublishing@gmail.com

Book Available online at:

https://www.bhumipublishing.com/book/

PREFACE

In the ever-evolving landscape of knowledge and innovation, the realms of Computer Science, Mathematical Science, and Information Technology stand as pillars of progress, constantly shaping and reshaping the way we perceive and interact with the world. As we embark on this journey through the pages of "Research Trends in Computer Science, Mathematical Science, and Information Technology," we find ourselves at the intersection of curiosity and discovery.

This compendium serves as a testament to the ceaseless pursuit of understanding, the tireless exploration of uncharted territories, and the collaborative efforts of brilliant minds dedicated to unraveling the mysteries of our digital age. From the intricate algorithms that power our technological marvels to the abstract beauty hidden within the depths of mathematical equations, this book delves into the heart of disciplines that not only define our present but also lay the foundation for an extraordinary future.

As we navigate through the chapters, each penned by experts and visionaries in their respective fields, we witness the emergence of cutting-edge concepts, transformative breakthroughs, and the synergies between disciplines that promise to revolutionize the way we approach problems and seek solutions. The intellectual tapestry woven within these pages reflects the diversity and dynamism of the research landscape, showcasing the multifaceted nature of scientific exploration.

From the practical applications of machine learning algorithms to the elegant theories of abstract mathematics, and the transformative impact of information technology on every facet of our lives, the research trends encapsulated in this volume offer a panoramic view of the intellectual landscape. It is a canvas painted with the strokes of innovation, collaboration, and the relentless pursuit of excellence.

As we dive into the narratives of progress and delve into the depths of knowledge, may this collection inspire the reader—be they a seasoned researcher, an aspiring student, or a curious mind—to join the global conversation, to question the status quo, and to contribute to the ever-expanding tapestry of human understanding.

Welcome to a voyage through the frontiers of Computer Science, Mathematical Science, and Information Technology—an exploration that transcends disciplines and opens doors to a future where the boundaries of what is possible are defined only by the limits of our collective imagination.

Editors

TABLE OF CONTENT

FACING CHALLENGES IN DEVELOPMENT OF MT AND NLP APPLICATIONS IN THE AVIATION DOMAIN FOR INDIAN LANGUAGES

Saptarshi Paul

Department Of Computer Science,

Assam University, Silchar (Central)

Corresponding author E-mail: paulsaptarshi@yahoo.co.in

Abstract:

Developing NLP based real-time projects have always been a challenge for scholars and developers. Collection of resources determines how the project is going to perform and up to what accuracy. TA determines the accuracy of translations and transliterations while other parameters determine how that particular NLP system performs. Proper resource identification, pre-processing them and then using it to get the desired results is the ultimate challenge. In most countries, resource like corpora and retrievable data are scarce. The pandemic has increased the challenge of resource collection and cleaning manifolds. This paper/chapter is aimed at discussing on how to deal successfully and efficiently on the development of NLP based projects that are related to Aeronautics and Aero-science. The features of Aeronautics English are discussed and then the various sources for the resources and tools are taken up in details.

Keywords: NLP, Aviation, Real-time Projects, Online resources.

Abbreviations:

TA	: Translation Accuracy
NLP	: Natural Language Processing
MT	: Machine Translation
AI	: Artificial Intelligence

Introduction:

The Covid-19 Pandemic has restricted the interaction between researchers and guides to be mostly over the various available online modes. Though theoretical classes can be handled through the various paid and free applications, the real challenge is handling real time project development for bachelors and masters dissertations. Interacting and guiding students for development of real time projects has become quite challenging. Projects in Natural Language Processing or NLP require referencing lot of resources such as data repositories and corpora. It also requires that the existing tools and methods be examined in depth. Another aspect is to understand the relation between the source and target language from a linguistic perspective. The rules that govern the source and target language are to be understood in detail. Moreover, Indians being non-native speakers of English add additional pressure in making sure that proper English grammar be followed. Aeronautics English being a specialized form of the English language demands to the point perfection. Aeronautics English is used in maintenance manuals, day to day aviation activities as well as in aero-space. Development of tools and applications involving aeronautics English calls for expert handling of both English for Specific Purposes - ESP (aeronautics in our case) as well as situations in English as a Foreign Language i.e. EFL (English-Indian language pair) in NLP. The main objective of the chapter is to discuss the various issues that have come up due to the pandemic to challenge the successful implementation of various NLP projects. The chapter aims to discuss on how through use of online links, resources can and has been collected and various tools and applications referred. The chapter explores on how the various online workshops (e.g. AICTE-ATAL) and webinars (e.g. cyber sanitization) helps in development of real time projects. Next the chapter highlights that though an array of challenges has arisen out of the pandemic, but these can be countered with taking help of systematic alternatives that are available online (paid, official and open-source). Though accessing many resources and tools need approval from authority through proper channel e.g. NPLT (https://nplt.in/demo/resources/text-corpus) and TDIL (http://www.tdil-dc.in/) many are readily available in form of open source. It also enumerates the various webinars and workshops imparting knowledge and skills for thesis and dissertation documentation. Finally the chapter discusses how the projects can be implemented and tested online while maintaining security and avoiding redundancy, the last part of the chapter is the general conclusion.

Aviation English

The English that is used in aeronautics encompasses civil aviation, maintenance, operation and aero-space terms and terminologies. This unique version of English follows a structured pattern that is used in both spoken and written formats. Aeronautics demands that both the sentences be spoken and drafted in a uniform manner and it applies for both native and non-native speakers of English. To implement it, International Civil Aviation Organization has dictated a uniform and unique pattern that is to be used by Pilots, Air Traffic Controllers, maintenance staff and ground operation employees. Each aircraft and airlines are allocated unique call-signs. These call-signs are achieved by combination of alphanumeric characters. International Civil Aviation Organization has determined on how phonetic equivalents of every word and number are to be spelled and drafted. Table shows how the phonetic pronunciation of numbers in aviation, while Table 2 depicts alphabets equivalent.

Table 1: Phonetic pronunciation of numbers in aviation

Numeral Element	ICAO pronunciation	Numeral Element	ICAO pronunciation
0	ZE-RO	7	SEV-en
1	WAN	8	AIT
2	TOO	9	NIN-er
3	TREE	Decimal	Day-See-Mal
4	FOW-er	Hundred	HUN-dred
5	*FIVE*	Thousand	Tou-Sand
6	SIX		

Table 2: Phonetic pronunciation of alphabets in aviation

English Alphabet	Pronounced as	English Alphabet	Pronounced as
A	*AL-FAH*	N	NO-VEM-BER
B	BRAH-VOH	O	OSS-CAR
C	CHAR-LEE	P	PAPA
D	DELLTAH	Q	KEH-BECK
E	ECK-OH	R	ROW-ME-OH
F	FOKS-TROT	S	SEE-AIR-RAH
G	GOLF	T	TANG-GO
H	HOH-TEL	U	YOU-NEE-FORM
I	IN-DEE-AH	V	VIK-TAH
J	JEW-LEE-ETT	W	WISS-KEY
K	KEY-LOH	X	ECKS-RAY
L	LEE-MAH	Y	YANG-KEE
M	MIKE	Z	ZOO-LOO

NLP projects in aviation and aeronautics

NLP applications generally encompasses any tool or software that is related with development of E-Dictionary, parts-of-speech tagger, pre-processing tools, post processing tools complex and complicated work may include developing bilingual translation models.NLP applications in aeronautics have huge scope as it is an unexplored branch of linguistics. Multiples of tools are used (Paul & Purkhyastha, 2018) around the world by companies but such tools and repositories are absent for Indian languages. This void also creates a huge pool of opportunities for researchers and technical project developers to contribute significantly. Scope exists for creation of tools and models for English and Bengali/native Indian languages. Development of tools as such demands resources ranging from databases to parallel corpora. Such projects demand attention both in the programming aspect (OpenNMT, 2016) as well as for creation of proper bilingual data repositories, which can be achieved through data collection, cleaning, preprocessing and processing before being tested and implemented.

Challenges due to the pandemic

The ongoing Pandemic has forced institutions to be shut down partially leading to the switching of the classes to online mode. Online classes are being conducted over an array of tools that are available in both free and paid forms. Though Paid tools and applications offer higher degree of flexibility and application freedom, the free and open source tools are also quite competent. Though theoretical classes can be conducted quite comfortably in online mode, problem arises in conducting practical experiments, research projects and preparation of respective dissertations. The main challenges that are faced are as follows:

1. Collection of resources

The gathering of resources for the implementation of NLP projects is very important. It not only helps us to understand the various types of NLP tools but also the various methods of translation. Each NLP tool or model requires corpora or data repository to act upon. Apart from that testing the tools to achieve the desired output and higher translation accuracy is a must. The pandemic has made the collection of resources a painstaking affair.

2. Referencing tools

No NLP tools are perfect, especially translation applications. During development of an NLP project, to get an in depth understanding of NLP tools it is needed to practically run them, understand their working methodology and find the ways in which more accuracy can be obtained. Detail understanding calls for discussions and exchange of ideas, among peers and fellow scholars. This has been quite impossible in the ongoing pandemic.

3. Physical execution of the project and determining translation accuracy

NLP tools needs to be executed and checked again and again before a satisfactory result can be obtained. NMT and SMT models need to be preprocessed and post-processed to attain the requisite BLUE or other metrics and score. All these are achieved through interacting with experts from linguistics, AI and NLP. It is more of an inter-disciplinary research wing and less of a single integrated one. The pandemic has thrown us challenges in attaining these interactions.

4. Documentation and structure of the project dissertation

All Projects end with proper structured documentation, achieving which is a skill in itself. Often students/researchers attend workshops and specialized classes to learn the art of documentation, references and bibliography. Along with the other aspects these classes and workshops have also been affected in the ongoing pandemic.

Alternatives approaches

With the advent of the pandemic, alternative approaches had to be devised and implemented and existing one explored to their fullest potential. These methods are mostly in the online medium. Let us explore some of them in this section that helps us in our NLP projects.

1. Accessing online corpora and data repositories

Online resources are available that can be accessed in a multitude of ways. While some resources can be accessed only through proper permission e.g. NPLT (https://nplt.in/demo/resources/text-corpus) and TDIL (http://www.tdil-dc.in/), there are many which can be accessed without any permission or charges. These free resources, especially for aeronautics are mainly restricted to reports of accidents, incidents and hence demands pre and post processing of the data. Maintenance manuals and procedures can be accessed only through proper channels and permissions. Resources for creation of corpora or data repository for aeronautics domain can be accessed from the following options:

1. Airport Authority of India (AAI) - Manual of Air Traffic Services (AAI- Resources,1995)
2. Directorate General of Civil Aviation (DGCA) manuals and accident/incident reports (DGCA reports,1937).
3. International Civil Aviation Organization (ICAO) reports on safety/incidents. (ICAO Documents, 1947)
4. National Transportation safety board-Aviation Accident Data summery (NTSB, 1967).
5. NASA-ASRS incident reports. (ASRS-Reports, 1987)
6. Aviation newsletters and aviation related blogs.

Apart from that a) phraseology manual of Chapter12-AAI (AAI MATS, 1995), b) Pilot’s handbook of Aeronautical knowledge 2016-2020 (Duncan, 2016) and c) Dictionary of aviation English (Dictionary-Aeronautical English, 2007).

2. Online tools and applications

An array of online tools is available (TDIL-2012) that can be used to test our developed corpus and repositories. These tools can give us an idea of how search techniques work in case of E-dictionaries and preprocessing tools. The output of the tools also acts as input to upgrade our corpora and models. Though very few standard tools are available for the aeronautics domain still for other domains multiple of tools are in use. Tools that are included in the TDIL website can be categorized as follows:

1. Freeware
2. Shareware
3. Research

It also consists of communities, a special students corner, list of useful links and Technology handshake corner all in online format. The tools and resources are available in the following types and numbers:

Table 3: TDIL and NPLT online tools and resources

Sequence number	**Source**	**Name of Resources Available**	**Number of Resources available**
1	*TDIL*	Localisation tools	20
2	*TDIL*	Publications	278
3	*TDIL*	Valuators	02
4	*TDIL*	Mobile apps	22
5	*TDIL*	NLP tools	105
6	*TDIL*	Linguistic resources	265
7	*NPLT*	Text Corpus	82
8	*NPLT*	Speech Corpus	44
9	NPLT	Word net	07
10	NPLT	PLS	04
11	NPLT	Tree Bank	16
12	NPLT	CIIL repository	31
13	NPLT	JNU repository	46

Source: NPLT and TDIL

Websites such as that of NPLT (NPLT, 2020), TDIL and CDAC can act as an anchor point for development of NLP tools.

3. Developing dissertation documentation skills

In line with refereeing the online resources, the students and research scholars also need to develop their report and dissertation documentation skills. In the absence of physical classes, the online medium is the only option available. Fortunately, both free and paid webinars and online workshops are being organized by not only institutions but also by organizations such as AICTE in the form of ATAL academy (AICTE-ATAL, 2018) which allows us to become adapt in the skills of documentation among other. ATAL not only provides us with technical workshops but also with skill development ones.

4. Completion and implementation of Projects

Many of the developed NLP models of Neural Machine Translation need to be run in Computers equipped with GPU or equivalent components. Such high end and costly computers are generally available in the laboratories of the institution. An alternative is to run the programs in cloud server, whose services are again available through paid (Google, 2021) and open source applications. Some of the multitudes of features that are available are as follows:

- **Entity analysis:** Identify entities and label by types such as person, organization, location, events, products and media.
- **Sentiment analysis:** Understand the overall sentiment expressed in a block of text.
- **Entity sentiment analysis:** Understand the sentiment for entities identified in a block of text.
- **Syntax analysis:** Extract tokens and sentences, identify parts of speech (PoS) and create dependency parse trees for each sentence.
- **Content classification:** Identify content categories that apply to a block of text.

Testing and avoiding redundancy:

The pandemic has forced the testing phase to be carried out in online mode. For manual testing of NLP outcomes (bilingual), respective linguistics has to certify the output. This can be achieved by send the corpora of the machine translated and human translated through email for verification and grading. While apart from advocating help of time consuming manual evaluation, the different automated scores can be calculated online to determine the Translation Accuracy, e.g. *https://www.letsmt.eu/Bleu.aspx.* Here we can enter the human translated file and machine translated file and the resultant BLEU score is displayed online. All dissertation and project needs to be checked for plagiarism and similarity reports needs to be generated. Use of both URKUND and TURNITIN is advocated for the similarity check. These can also be performed online but care has to be taken that the document under consideration is not stored in any repository to avoid redundant copies of the same documentation.

Conclusion:

This chapter has tried to illustrate the ways in which normally, NLP projects are implemented and the problems that have arisen due to the pandemic acting as barrier for those normal ways. The chapter then describes on how alternative methods using online resources can help us to implement the various NLP projects. These include ways and means from collecting resources to testing, implementation and documentation. Though the pandemic had created an array of problems in implementation of real time projects of NLP, scholars, students and faculties are now rapidly able to overcome these problems with the help of available tools, resources and applications. With time more such resources are expected to become available through online mode thus facilitating smooth conduct of practical and project works.

References:

1. AAI-Resources. (1995). https://www.aai.aero/hi/system/files/resources/
2. AAI MATS. (1995). https://www.aai.aero/en/system/files/resources/1%20Commercial%20Manual%202019.pdf
3. AI-Aviation Resources. (2018). https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-in-aviation-market-106037016.html
4. ASRS-Reports. (1987). https://asrs.arc.nasa.gov/search/reportsets.html
5. AICTE-ATAL. (2018). https://atalacademy.aicte-india.org/participant/workshop
6. DGCA reports. (1937). http://dgca.nic.in/accident/reports/contents_acc_rep.html
7. Dictionary-Aeronautical English. (2007). https://www.theairlinepilots.com/forumarchive/aviation-regulations/aviationdictionary.pdf
8. Duncan, J.S. (2016). *Pilot's Handbook of Aeronautical Knowledge*, U.S. Department of Transportation Federal Aviation Administration Flight Standards Service, FAA-H-8083-25B
9. Google. (2021). https://cloud.google.com/natural-language/pricing
10. ICAO Documents. (1947). https://www.icao.int/safety/airnavigation/AIG/Documents/Forms/AllItems.aspx
11. NPLT. (2020). https://nplt.in/demo/resources/text-corpus
12. NTSB. (1967). https://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx
13. OpenNMT. (2016). https://opennmt.net/OpenNMT-py/main.html
14. Paul & Purkhyastha (2018). NLP Tools used in civil aviation: A Survey. *International Journal of Advanced Research in Computer Science*, Vol. 9, No.2, pp: 109-114
15. TDIL. (2012). http://www.tdil-dc.in/

PREDICTIVE AND SIMULATION MODEL OF VANET FOR V2V COMMUNICATION

H. Parveen Begum* and P. Anusha

Department of Software Engineering,

Periyar Maniammai Institute of Science & Technology (Deemed to be University),

Vallam, Thanjavur 613 403, Tamil Nadu, India

*Corresponding author E-mail: parveenbegum@pmu.edu

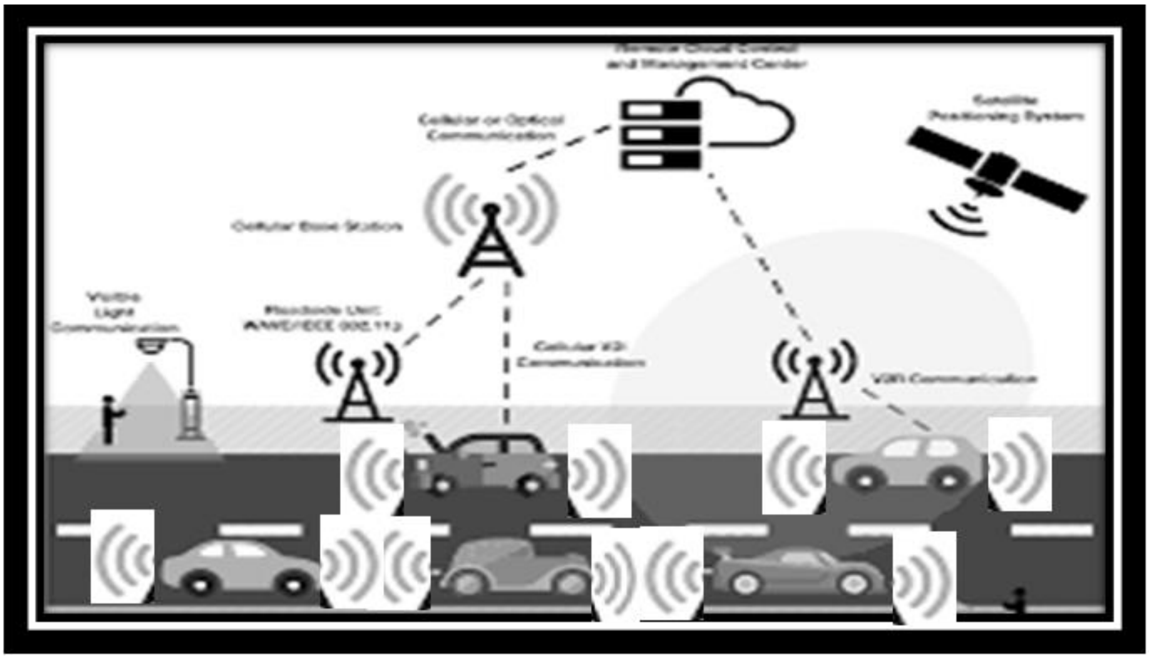
Abstract:

The main aim of this paper is to make communication between vehicle to vehicle. VANET (Vehicle Adhoc NETwork) is the branch of MANET (Mobile Adhoc NETwork). VANET includes several kinetic and akinetic (static) vehicles, which able to transfer and receive data packets without a central access point. The main expectation of this paper is to make the vehicle to communicate with other vehicle through its sensor. Usually Internet of Vehicle has the communication between V2V(vehicle to vehicle), V2I(vehicle to and V2X(vehicle to all), V2V communication may provide very feasible collision free traffic, route rescheduling, network traffic monitoring and many more advantages.

Keywords: VANET, Adhoc, V2V Communication.

Introduction

V2V communication is the most thrust area which was the research undertaken from 2003 and originally started from 1990. V2V communication is to find shortest route, congestion free road, and assistance for line change, for toll, for parking vehicle, to find some specific location like ATM/Mall/House, seeking permission to overtake, to refill fuel tank / charge battery, antitheft of vehicles and many more. Communication between vehicles is the solution for major accidents happening in the congested roads. In most of the time human may get confused while taking on spot decision to move left or right or even to press brake or to give accelerator which lead major accidents. If we handover the decision making job to the vehicle itself, it is safe because in the fraction of seconds it can analyze thousands of possibilities and choose best one, plus if it communicate to the nearby vehicle it can avoid on hitting the opposite vehicle or can get space from next vehicle.


Remote Cloud Control and Management Center
Cellular or Optical Communication
Satellite Positioning System
Cellular Base Station
Visible Light Communication
Roadside Unit WAVE/IEEE 802.11p
Cellular V2I Communication
V2R Communication

Methodology:

The methodology of VANET for V2V communication is consists of four phases

- Investigation
- Develop a hardware system
- Study and analysis
- Evaluation

Investigation

Fundamental HW requirements for VANET for V2V Communication

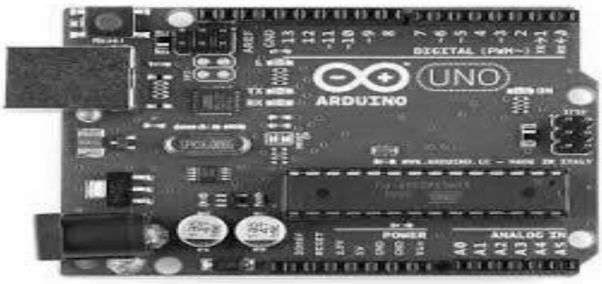
Horn / Buzzer –

Horn / Buzzer are used to produce sound from vehicle, based on the decibels 60dB – 130dB will be suitable for this job. Sound intensity is inversely proposition to the distance. Which means when distance increases the sound will be decreases. Based on this when the sensor sense louder sound intensity, then immediately the action has to take like automatically reduce the speed or use break or move towards left or right to give space to the opposite vehicle.

Sensors:

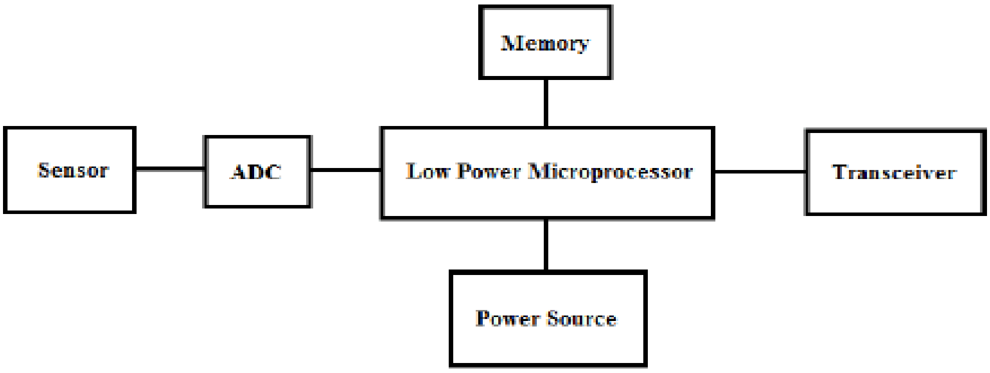
Human have five sense organs are eye, ear, nose, mouth, skin. Where eyes for sight, ears for sound, nose for smell, mouth for taste and skin for temperature. V2V communication requires several types of sensors 1. To sense air moist and temperature, to predict rain and switch on car wiper, to change mode of AC. 2. To sense signal of other vehicles like horn sound, ultrasound waves to sense the vehicle's distance. 3. Light recognition sensor to sense Traffic signal. Fortunately, no need to sense taste and smell now may be implemented in future.

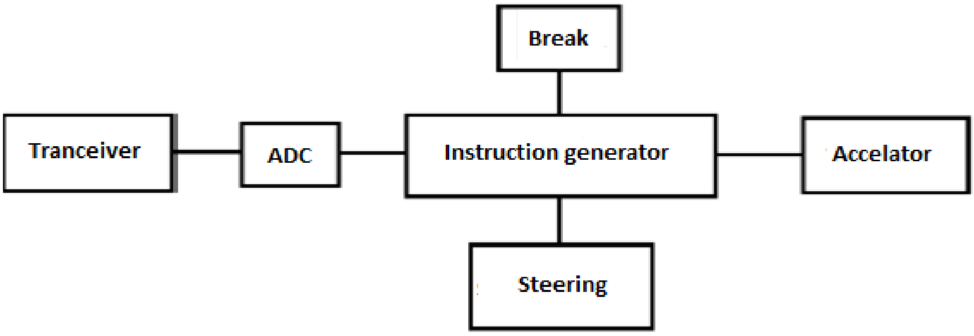
For this project we need a potential sensor because the sound intensity may be come from any side either from opposite side or from back side or even from next line which does not disturb the sensor's line. First, sense the sound intensity then analyze strength of the sound intensity to calculate the distance plus should find the direction also. Based on all parameters the instruction has to generate whether to reduce speed or to push up brake or to move.



Develop a hardware system:

Block Diagram for V2V Communication

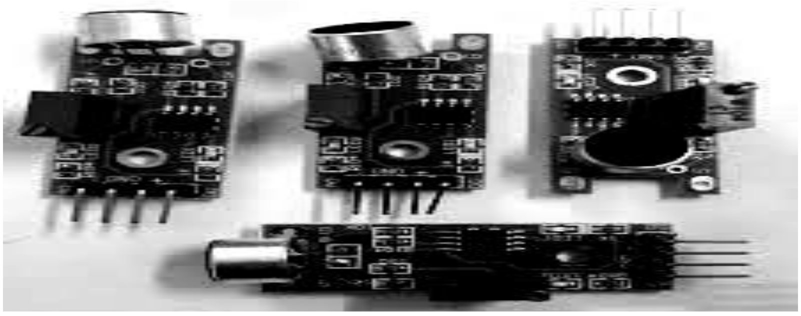




Signal Sensors

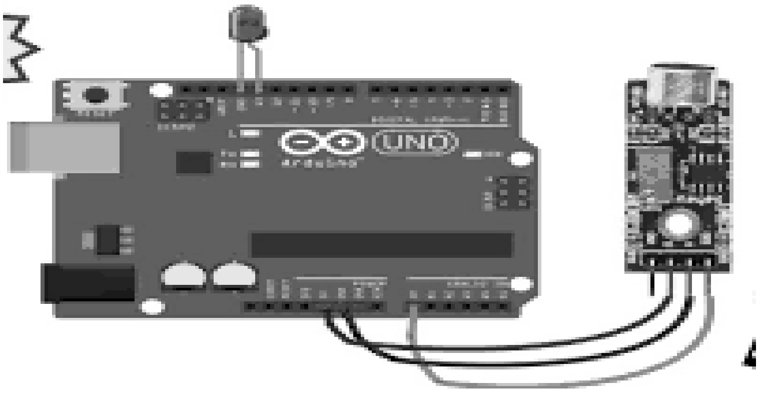
Sensors are used to sense signals like sound, air moisture, from the surroundings.

Sound sensor



Signal Convertors

Convertors are used to convert signals from sensors into commands for the corresponding devices. Traditionally many different types of converters are available but now compressed circuits are available in a single micro controller chip



Signal Transmitters

Transmitters are used to transmit the generated commands to the corresponding parts of the vehicles. Transmitters are either devices or USB cables HDMI etc

LMP

Low Power Microprocessors will take response for all of the process of hardware connected together.

Fundamental SW requirements for VANET for V2V Communication

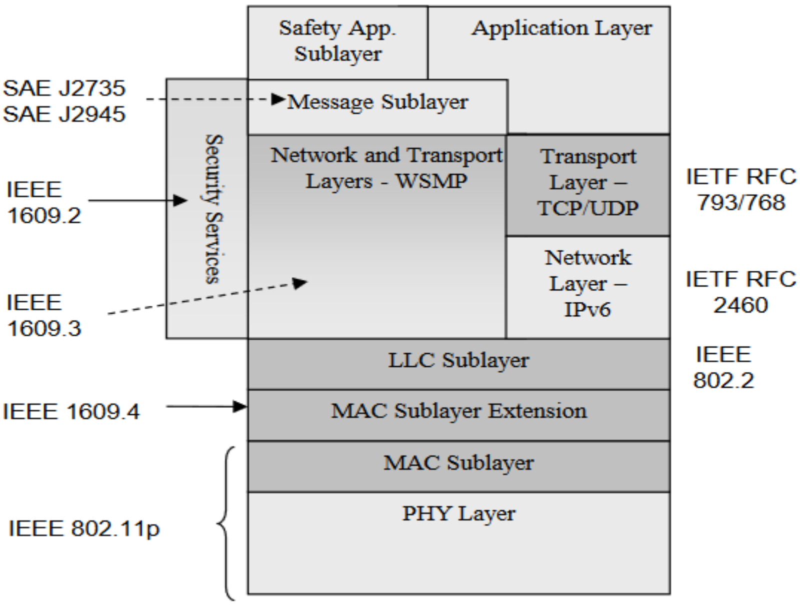
The fundamental Software Requirements for the proposed system is java and python along with ITS – Intelligent Transport System that is IEEE802.11P Protocol standard.

Java and Python – "jython"

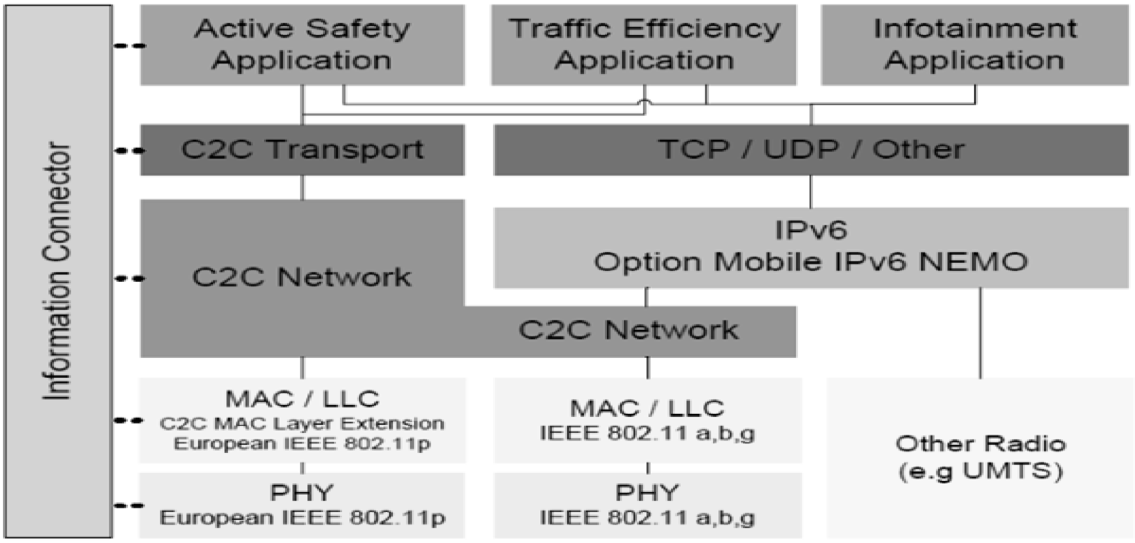
Java is the platform independent programming language which very much suitable for all type of applications. It provides more flexibility than any other previously available languages. But, when compare with coding style of java, python overcomes the java coding style, because the coding style of Python is eternally simple than any other competitive languages available. Of course, some tough tasks can be handled by java alone in this case we may use java + python = jython, is the project which utilizes the benefits of java and python together.

Protocol required for VANET V2V Communication

IEEE 802.11P – ITS Intelligent Transport System is the protocol used for VANET for V2V communication. This protocol is an efficient protocol for wireless adhoc VANET V2V communication because there is no fixed layout and route map for network components. This protocol provides layout and route map for the network components in an on demand manner. It generates the layout and route map quickly and more efficiently.



IEEE 802.11P Protocol Stack

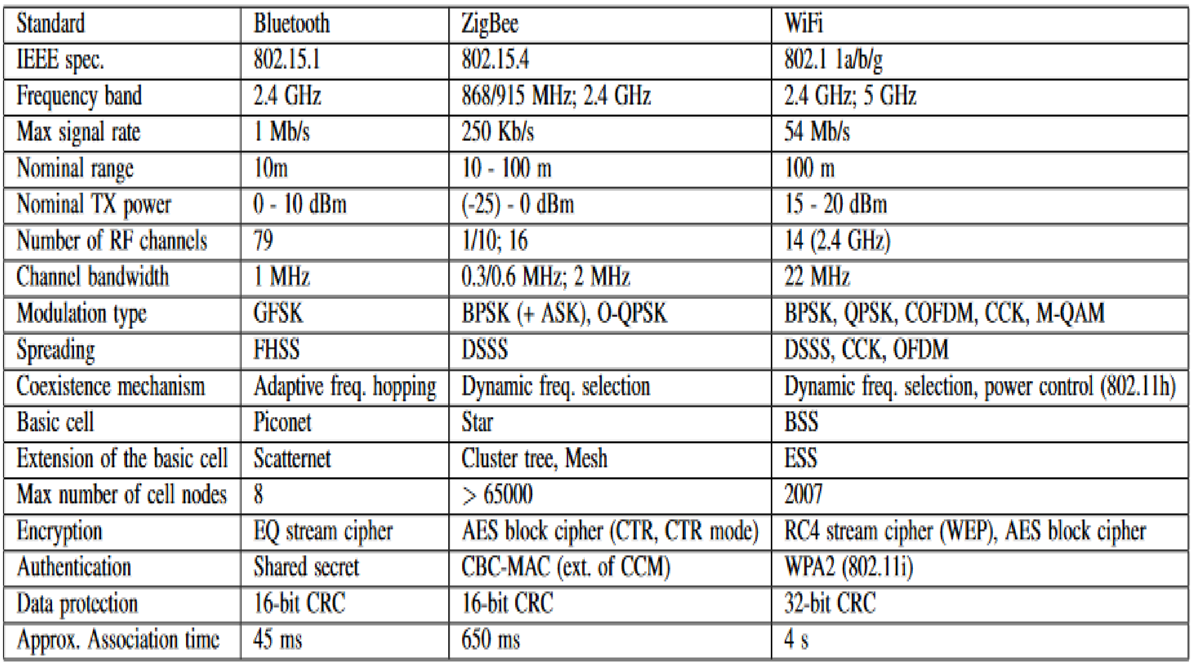


IEEE 802.11P Protocol Stack

Study and analysis:

Signal Strength and Range

Table 1: Comparison of various wireless communication



Standard	Bluetooth	ZigBee	WiFi
IEEE spec.	802.15.1	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10m	10 - 100 m	100 m
Nominal TX power	0 - 10 dBm	(-25) - 0 dBm	15 - 20 dBm
Number of RF channels	79	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	0.3/0.6 MHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Spreading	FHSS	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptive freq. hopping	Dynamic freq. selection	Dynamic freq. selection, power control (802.11h)
Basic cell	Piconet	Star	BSS
Extension of the basic cell	Scatternet	Cluster tree, Mesh	ESS
Max number of cell nodes	8	> 65000	2007
Encryption	EQ stream cipher	AES block cipher (CTR, CTR mode)	RC4 stream cipher (WEP), AES block cipher
Authentication	Shared secret	CBC-MAC (ext. of CCM)	WPA2 (802.11i)
Data protection	16-bit CRC	16-bit CRC	32-bit CRC
Approx. Association time	45 ms	650 ms	4 s

Algorithms for V2V communication

Algorithm 1: V2V Communication time(L)

```
Input: geographic locations of vehicles (a and b)and their transmission ranges ($R_a$ and $R_b$)
For each vehicle pair a and b do
    For $V_{rel}$ =1 to 120 miles/hr do
        Compute $D_{a,b}$ [$\sqrt{((x_a - x_b)^2+(y_a - y_b)^2)}$]
        If $D_{a,b} < min\{R_a, R_b\}$ then
            Compute $L= D_{a,b} / V_{rel}$
        Else
            L=0
        Endif
    End for
End for
Output: time(L) for different relative speeds
```

Algorithm 2: Message Size Calculation

```
Input: time (L) from Algorithm 1, Association time A, data rate $D_r$
If 0 < L then
```

```
3.  Compute $T = \max\{0, L-A\}$
4.  Compute message size with header $MSH = D_r * T$
5.  Actual message size $MS = MSH$ – Header – size
6.  End if
7.  Output: Message Size (MS)
```

Algorithm 3: P_8 Calculation

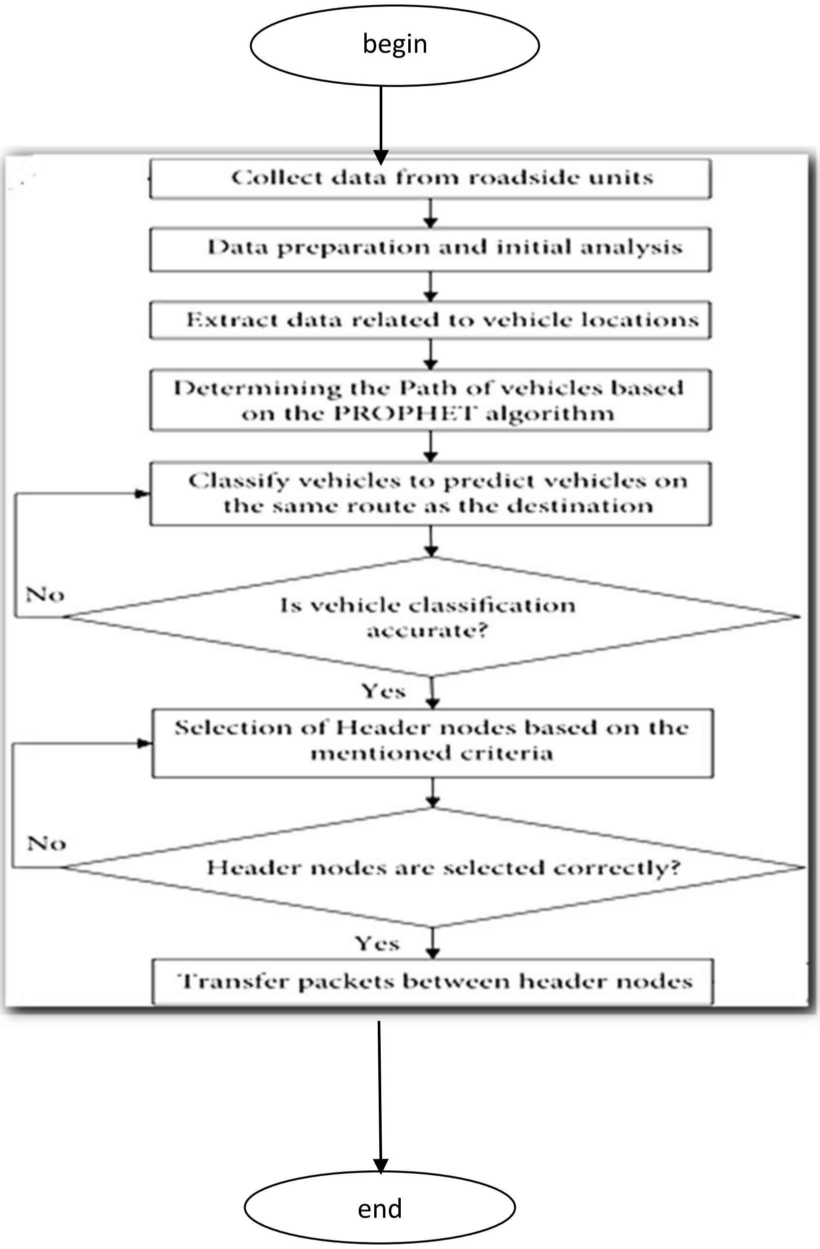
```
1.  Input: geographic locations of vehicles (a and b) and their transmission ranges ($R_a$ and
    $R_b$), $A, D_r, S, \beta$, Simulation time and success = 0
2.  Repeat
3.  For each vehicle pair a and b do
4.  For $V_{rel}$ = random(0,70) do
5.  If $D_{a,b} < \min\{R_a, R_b\}$ then
6.  Compute $L = D_{a,b} / V_{rel}$
7.  Else
8.  L=0
9.  End if
10. Compute T using (
    $$\begin{aligned} \overline{T} &= \frac{1}{\beta} e^{-A\beta} \times P_r \\ &= \frac{1}{\beta} e^{-A\beta} \times min\left\{1, \left[\frac{\left(\frac{D_{i,j}}{v_{rel}}\right)}{\left(A + \frac{S}{D_r}\right)}\right]\right\} \end{aligned}$$
11. If A+ T<= L then
12. Success=success+1
13. End if
14. End for
15. End for
16. Until of trials and simulation Time
17. $P_8$ = success / trials
18. Output: $P_8$
```

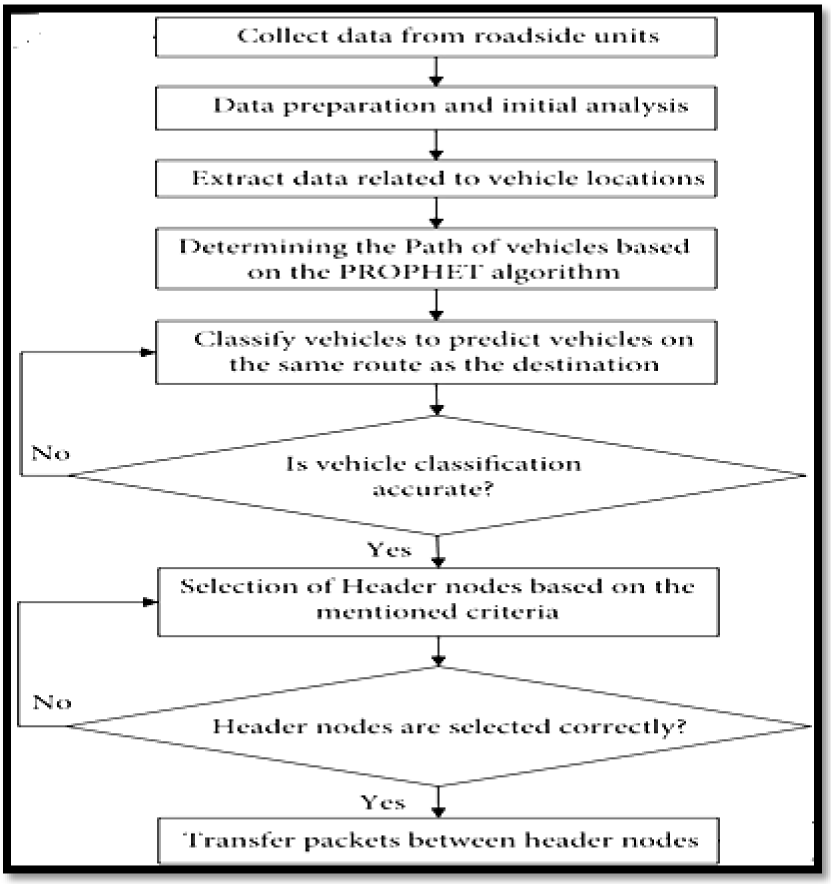
Cluster based Routing Protocol for VANET

Routing protocols are used to find best path between source and destination through the one or more network and store the path in routing table, for data transmission. Basically Routing protocols are three types they are Distance Vector Routing Protocol, Linked state Routing Protocol and Advanced Distance Vector Routing Protocol. In VANET communication, single type of routing protocol could not be able to give desire routing results so, we have go with cluster based routing protocol that is different types of protocols for different communication.

Evaluation and Conclusion of VANET for V2V communication:

If my proposed work is implemented then the road accidents rate will be reduce significantly. The human work pressure lessens gradually. Automobile began to make decision also, the vehicles initiates the conversation with other vehicles and decides how to handle both the normal and abnormal situations. Once we set up the communication between the vehicles through the network and protocol, the vehicles take over the leisure things from us.


begin
Collect data from roadside units
Data preparation and initial analysis
Extract data related to vehicle locations
Determining the Path of vehicles based on the PROPHET algorithm
Classify vehicles to predict vehicles on the same route as the destination
No
Is vehicle classification accurate?
Yes
Selection of Header nodes based on the mentioned criteria
No
Header nodes are selected correctly?
Yes
Transfer packets between header nodes
end


Collect data from roadside units
Data preparation and initial analysis
Extract data related to vehicle locations
Determining the Path of vehicles based on the PROPHET algorithm
Classify vehicles to predict vehicles on the same route as the destination
No
Is vehicle classification accurate?
Yes
Selection of Header nodes based on the mentioned criteria
No
Header nodes are selected correctly?
Yes
Transfer packets between header nodes

Flow chart of Cluster based routing protocol model

References:

1. U.S. Department of Transportation. (2013, April 13). IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE). Retrieved November 14, 2014.
2. European Telecommunications Standards Institute. (2013, April 16). Final draft ETSI ES 202 663 V1.1.0 (2009-11). Retrieved from https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.03.01_30/en_302663v010301v.pdf
3. ASTM International. (2007, July 15). E2213-03 Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems. doi:10.1520/E2213-03R10.
4. ETSI. (2013, September 9). Intelligent Transport Systems. Retrieved from the ETSI website: https://www.etsi.org/archived/302663/01.03.01_30/en_302663v010301v.pdf

SELECTION OF BEST CLOUD COMPUTING SERVICE MODEL ACCORDING TO BUSINESS NEEDS

Sumit Chopra*, Dinesh Bhatia, Gagandeep Singh and Rajesh Sharma

GNA University, Phagwara

*Corresponding author E-mail: sumit.chopra@gnauniversity.edu.in

Abstract:

Cloud computing has become a critical technology for businesses and organizations, enabling them to access scalable and flexible computing resources on demand. This paper provides an introduction to cloud computing, including its service models, deployment models, and key components. The paper also discusses the benefits and challenges of cloud computing, as well as its applications in various industries. In addition, the paper examines the security concerns correlated with cloud computing, including access management, data protection, and compliance. Finally, the paper provides an overview of the best practices for cloud cost optimization and migration strategies. By providing a comprehensive overview of cloud computing, this paper aims to help businesses and organizations understand the potential of this technology and how it can be leveraged to achieve their goals.

Keywords: Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), Public Cloud, Private Cloud, Hybrid Cloud, Multi-Cloud, Cloud Security, Cloud Storage

Introduction:

Cloud computing refers to the provision of various computing services over the internet, encompassing resources such as servers, storage, databases, software, and other related services. It allows individuals and organizations to access these resources on demand without needing to own and manage their own computing infrastructure [3]. In very layman's language, we can say that cloud computing is nothing but using resources of service providers like Amazon Web Services, Google Cloud, Microsoft Azure, etc remotely from our location & these resources are kept in the Data Centres of respective Cloud Service Providers [7].

Data Centers

A data center is a place where a large amount of computer equipment is stored and managed, including servers, storage devices, and networking equipment. It is where companies store and manage the data, they need to run their businesses, such as customer information, financial records, and other important data. Data centers are usually equipped with backup power supplies, cooling systems, and security measures to protect the equipment and data from damage or theft. Think of a data center as a big storage room for computers that allows companies to keep their information safe and accessible.

Types of Cloud Computing Service Models

The many services that cloud providers deliver to their clients are referred to as cloud computing service models. We can control or access all three using a web browser or any other client software like apps. Instead of creating new copies of files, again and again, we can use the concept of version control through cloud computing. Three primary categories of cloud computing service models are:

1. Infrastructure-as-a-Service (IaaS)
2. Platform-as-a-Service (PaaS)
3. Software-as-a-Service (SaaS)

1. IaaS:

It makes the resources placed at remote locations available online over the internet. Without having to buy and maintain real hardware, consumers can access computational resources like virtual machines (VMs), storage, and networking infrastructure as they need them [13]. The cloud vendor is responsible for the supervision of the underlying structure, while customers have control over the operating system, applications, and data running on the virtual machines [5].

Key Components of IaaS:

1. **Virtual Machines (VMs):** Virtual machines are the core building blocks of IaaS. A VM is a software-based representation of a physical machine that emulates the hardware and runs an operating system, allowing users to install and run applications as if they were running on a physical machine. IaaS providers offer a range of VM sizes and configurations to meet different needs, such as CPU, memory, and storage capacity.
2. **Storage:** Companies that offer cloud services provide different types of storage options. These include object storage for things like pictures and videos, block storage for things like databases that need to be really fast, and file storage for sharing files between different programs.
3. **Networking:** IaaS providers offer networking services that allow customers to connect and manage their VMs, storage, and other resources. Networking services include virtual private clouds, load balancers, and firewalls. Customers can create and manage their own virtual networks, subnets, and security groups to control traffic flow and access to resources.

Suppose you're working on an AI/ML project which needs high configuration machine to work properly but you don't have that machine. Then you Launched a Virtual Machine (EC2) on AWS with that high configuration then you'll be having full control on that resource, you can configure all things you want to do on that resource [2].

2. PaaS

PaaS is a sort of service that offers a platform for developers to develop, deploy, and maintain applications without having to worry about the infrastructure behind them. This allows organizations to bring their applications to market faster by freeing developers from the burden of infrastructure management.

PaaS is particularly advantageous because it delivers a number of tools, middleware, and databases that developers consume to make and launch custom applications. This includes support for various programming languages and frameworks like Python, Ruby, Java, and Node.js, as well as unified expansion environments (IDEs) and code editors.

PaaS also provides middleware services that can be used to build and deploy scalable and highly available applications. Middleware services include application servers, web servers, messaging systems, and data integration tools. These services help to facilitate communication between different components of an application, such as between a web server and a database [4].

PaaS providers also offer a range of databases, including relational databases, NoSQL databases, and in-memory data stores. These databases can be used to store and manage data for applications, such as customer information, transaction data, and user profiles.

PaaS is exemplified by the utilization of various cloud platforms, such as Salesforce and Google App Engine. These platforms offer a PaaS service model, enabling businesses to develop and distribute customized applications [6]. The Salesforce platform provides an array of development tools, middleware services, and databases, which can be leveraged to create and deploy tailored applications. Similarly, Google App Engine is a PaaS offering that facilitates the construction and deployment of custom applications on the Google Cloud Platform, with the aid of diverse development tools, middleware services, and databases [14].

PaaS can be a cost-effective option for businesses, as they only pay for the computing resources and development tools they use. This means that they do not need to invest in physical hardware and infrastructure, which can be expensive to purchase and maintain [15]. PaaS can also be highly scalable, allowing businesses to handle sudden spikes in traffic or workload without having any kind of Downtime.

3. SaaS

A cloud computing service model called SaaS allows organizations to use software applications through the internet except having to install, operate, or update the program on their own machines. With SaaS, firms can concentrate on their core business operations rather than IT management, which makes it a popular choice for organizations wishing to lower the costs and complexity of administering software programs.

SaaS providers are responsible for maintaining the software applications, including software updates, security patches, and backups, which helps to reduce the burden on IT staff and lower costs.

SaaS can be a cost-effective option for businesses, as they only pay for the software and services they need, without having to invest in physical hardware and infrastructure. SaaS providers are responsible for maintaining the software applications, which helps to reduce the burden on IT staff and lower costs. SaaS can also be highly scalable, allowing businesses to handle sudden spikes in demand without having to invest in additional hardware or software.

Google Workspace is an excellent example of Software as a Service (SaaS) in action. Formerly known as G Suite, Google Workspace is a toolset of productivity and association tools that allow businesses and individuals to work seamlessly together in the cloud [1].

One example of SaaS in action is the use of the Microsoft Office 365 suite. Microsoft Office 365 is a SaaS offering that provides businesses with access to the full suite of MS Office apps over the internet. The software applications are hosted in the cloud by Microsoft, which means that businesses do not require installing or keeping up with the programme on their own PCs.

Office 365 also provides a range of additional services, including email, file storage and sharing, and collaboration tools such as Microsoft Teams. These services are fully integrated with the software applications, which allows businesses to work more efficiently and collaboratively.

Management in Service models

Cloud computing offers different service models that define the extent of services supervised by the cloud service provider and what responsibilities are left to the customer [11]. IaaS is a type of model where the cloud service provider is in charge of overseeing the fundamental structure components, like servers, networking, and storage. On the other hand, the customer is accountable for managing the applications, operating system, and data. In other words, the consumer has complete control over their virtual machines, including the installation and management of software, and configuring and maintaining the operating system. The cloud service provider's responsibility is limited to ensuring the underlying infrastructure is operational and available and providing basic management tools for infrastructure management [8].

In PaaS, the cloud service provider manages the entire platform on which the customer can build and run applications. This includes managing the infrastructure, operating system, and middleware, as well as providing development tools, databases, and other services. The client is responsible for managing the data and application code. This means that the customer has a high level of control over the applications they build and run but without the overhead of managing the underlying infrastructure. The cloud service provider is responsible for ensuring that the platform is available and performing well, and for providing tools for managing applications on the platform.

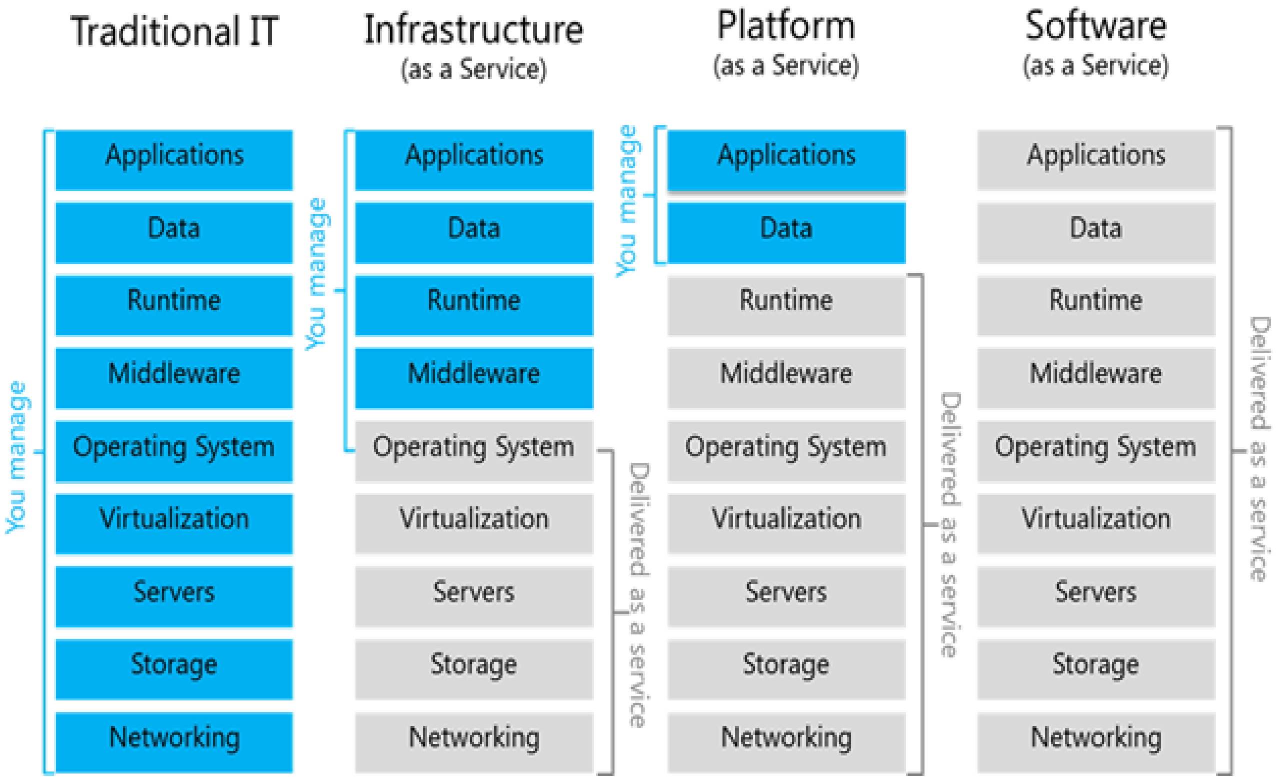


Fig. 1: Management in Service Models

In SaaS, the cloud service provider manages everything, from the infrastructure to the applications themselves. The customer only uses the application and has no responsibility for managing the underlying infrastructure or platform [12]. This means that the customer has very little control over the application, but also has no responsibility for managing it[10]. The cloud service vendor is in charge of ensuring that the application is available and performing well and providing tools for managing the application.

In summary, the cloud service models define what responsibilities are managed by the cloud service provider and what responsibilities are left to the customer. In IaaS, the customer has the most control over the underlying structure and is responsible for managing the OS, data, and applications. In PaaS, the customer has a high level of control over the applications they build and run, but without the overhead of managing the underlying architecture [9]. In SaaS, the cloud service provider manages everything, and the customer has no responsibility for managing the underlying infrastructure or platform.

It's important for customers to understand what responsibilities they have when using cloud services, as well as what responsibilities the cloud service provider has. This understanding can help customers make informed decisions about what service model to use and how to best manage their applications and data in the cloud.

Table 1: Comparison between different Cloud Architecture Models

	IaaS	**Paas**	**Saas**
Definition	Infrastructure as a Service, where a third-party provider offers virtualized computing resources over the internet.	Platform as a Service, where a third-party provider offers a pre-configured platform for developing, testing and deploying applications.	Software as a Service, where a third-party provider offers access to pre-built software applications over the internet.
Example	Microsoft Azure, Compute Engine, Amazon Web Services, Google.	Heroku, Google App Engine, Microsoft Azure	Salesforce, Google Workspace, Dropbox
User Responsibility	Users are responsible for managing applications, data, runtime, middleware, and operating systems.	Users are responsible for managing applications and data, while the provider manages runtime, middleware, operating system, virtualization, servers, and storage.	The provider is responsible for managing everything from the application to the infrastructure, including runtime, middleware, operating system, virtualization, servers, storage, and networking.

Flexibility	Offers maximum flexibility as users have complete control over the infrastructure and can configure and manage it as per their requirements.	Offers moderate flexability as users can focus on developing and deploying applications without worrying about the underlying infrastructure.	Offers limited flexibility as users can only use the pre-built applications provided by the provider without any scope for customization.
Cost	Users pay for the computing resources they use on a pay-as-you-go basis, making it a cost-effective option for businesses with varying computing needs.	Users pay for the platform services they use on a pay-as-you-go basis, which can be more expensive than IaaS but less than SaaS.	Users pay a fixed subscription fee for access to pre-built software applications, which can be more expensive than IaaS and PaaS.
Use Case	Best suited for businesses with fluctuating computing needs or those who require complete control over their infrastructure.	Best suited for businesses looking to develop and deploy applications quickly and efficiently without worrying about infrastructure management.	Best suited for businesses looking for pre-built software applications that can be used out-of-the-box without any customization.

Conclusion:

Cloud computing is a powerful tool that can help businesses of all sizes to improve their efficiency, agility, and scalability. By leveraging the cloud, businesses can access the resources they need on demand, without having to invest in and maintain their own IT infrastructure. However, it is important to choose the right cloud computing service model for your business needs. IaaS provides the most control and flexibility, but it also requires the most management expertise. PaaS is a good option for businesses that want to focus on developing and running applications without having to manage the underlying infrastructure. SaaS is the simplest option, as it requires the least amount of management overhead. No matter which cloud computing service model you choose, make sure to select a reputable cloud provider with a proven track record of reliability and security.

Reference:s

1. Anggraini, V. (2022). Google workspace meningkatkan hasil belajar matematika siswa. *Secondary, 2*(3), 364–369. https://doi.org/10.51878/secondary.v2i3.1411
2. Bandaru, A. (2020). Amazon Web Services. *ResearchGate*. https://www.researchgate.net/publication/347442916_amazon_web_services
3. Garg, V. (2023). Introduction of Cloud Computing. *ResearchGate*. https://www.researchgate.net/publication/368873526_Introduction_of_Cloud_Computing
4. Gomes, D. G., Bittencourt, L. F., & Tolosana-Calasanz, R. (2015). Introduction to the Special Section on "Cloud computing: Current advances and new research trends." *Computers & Electrical Engineering.* https://doi.org/10.1016/j.compeleceng.2015.11.015
5. Javanmard, M. O. M., & Zareiyan, F. (2014). Introduction to cloud computing and examine in the world and Iran. *ResearchGate*. https://www.researchgate.net/publication/278300460_Introduction_to_cloud_computing_and_examine_in_the_world_and_Iran?_sg=0b5X726aCETm9bd_Ro391bv9NAfZxGOLe5zCF3azd9kiZbTPYtkNLO5GRN7PGbSzcOIoQpN_-c7bXnA
6. Pal, D., & Nag, A. (2022). Cloud Computing: A Paradigm Shift in IT Infrastructure. *ResearchGate*. https://www.researchgate.net/publication/361462059_Cloud_Computing_A_Paradigm_Shift_in_IT_Infrastructure
7. Sabale, A., & Ilag, B. N. (2022). Introduction to Azure Virtual Desktop. In *Apress eBooks* (pp. 1–14). Apress. https://doi.org/10.1007/978-1-4842-8063-8_1
8. Sheth, A., Bhosale, S. S., Kadam, H., & Prof, A. (2021). Research Paper on Cloud Computing. *ResearchGate*. https://www.researchgate.net/publication/352477780_Research_Paper_on_Cloud_Computing
9. Shuraida, S., & Titah, R. (2023). An examination of cloud computing adoption decisions: Rational choice or cognitive bias? *Technology in Society, 102284.*
10. Sharma, M., Singh, A., & Daim, T. (2023). Exploring cloud computing adoption: COVID era in academic institutions. *Technological forecasting and social change, 193,* 122613.
11. Tianfield, H. (2011, October). Cloud computing architectures. In *2011 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1394-1399). *IEEE.*

12. Cusumano, M. (2010). Cloud computing and SaaS as new computing platforms. *Communications of the ACM*, *53*(4), 27-29.
13. Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infrastructure as a service (IAAS). *International Journal of Engineering and Information Technology, 2*(1), 60-63.
14. Metz, R. (2011). Understanding the Cloud: An Introduction to the Cloud. *Getting Started with Cloud Computing.*
15. Peng, J., Zhang, X., Lei, Z., Zhang, B., Zhang, W., & Li, Q. (2009, December). Comparison of several cloud computing platforms. In *2009 Second International Symposium on Information Science and Engineering* (pp. 23-27). *IEEE.*

A COMPARISON OF DIFFERENT TYPES OF REAL TIME OPERATING SYSTEM WITH THEIR APPLICATIONS

Sumit Chopra*, Palak, Anchal Nayyar and Simranjot Kaur

GNA University, Phagwara

*Corresponding author E-mail: sumit.chopra@gnauniversity.edu.in

Abstract:

In Real Time Systems, each job has a set deadline by which it must be finished; failing to do so will result in a significant loss, and even if the result is produced, it will be useless. an embedded system-developing real-time operating system (RTOS). Various types of Real-Time Operating System's are Hard or Soft or Firm Real Time Operating System.The Computable and Conditional results that are obtained from real-time operating system analysis are presented here. Results are generated. Important tasks should be executed on a priority basis, minimizing text switching time. Real-time operating systems have been crucial to the expansion of numerous commercial RTOS products over the past two decades. We have chosen a few business RTOSs in various classifications of constant applications and examined their continuous capacities. This paper shows an illustration of various commercial RTOS to help you learn more about RTOS.

Keywords: RTOS, MQX, FreeRTOS, WindowCE, eCos, QNX Neutrino, RTAI**,** VxWorks.

Introduction:

The commercial and non-commercial RTOS i.e, Real Time Operating System is the OS in which each chores perform their functions within "PRESCRIBED DEADLINE" and "FAILURE" to these deadlines cause 'SEVERE CONSEQUENCES', so it has fixed "TIME CONSTRAINTS". 'USER CONSTRAINTS' and 'RESOURCE UTILIZATION' are the secondary concerns for RTOS. It provides the basic assistance for the tasks like Scheduling, Synchronization, Resource Management, Communication, Input/Output and Precise timing etc. Real Time Operating System (RTOS) is an operating system that is mainly chosen for taking up real time embedded applications that develop progressively around processes and controllers. RTOS are present in Embedded systems [1]. As the name implies, embedded is fixed or attached to something. A specialized real-time computer system that is a component of a larger system is this one. The utilization of a Constant Working Framework exists on account of military applications, if you have any desire to drop a rocket, then, at that point, the rocket should be dropped with a specific accuracy, in the event that the timing is off base, it prompts tremendous harm to the encompassing region. The various RTOS are designed primarily to comprehend multiple processes simultaneously and guarantee that processes comprehend events within a predetermined time limit.

Real time Operating Systems make the effective use of various paradigms like: Hard and Soft various Real time Guarantee: careful design and analytics can result in meeting hard deadline environments, The larger more dynamic environment and kernels support the soft real-time systems [2]. When you choose a real time operating system then it will depend on how swift your system is so that it must react fast when the process is executed. For example, hard RTOS is used for medical systems in various organizations and soft RTOS is used for video streaming, Autopilot Systems, Airplane Sensors etc. The controller decides whether to approve new work entered the system or not. Resource cores: either resource reservations and reflective reservations that provide specific pieces of resources to tasks: metadata about your application, or about the functionality and features of different operating systems.

Consider the role of RTOS in mobile phones. Mobile phones have various functions such as call handling, call waiting notification, phone book maintenance, messaging, and other utilities such as web browsers, calculators, games and applications. The RTOS will handle it. Consider each of these mobile functions as a separate task. Let us consider an example that, on a digital phone the person opens the game and starts playing it and suddenly a call arrives. After that the caller ID will start sparking on the mobile’s screen. When the call is over, the person can resume the game from where he left off. In this case, it was observed that the RTOS uses priorities for processing tasks and uses context switching and scheduling for multitasking. This is the uniqueness of the Real Time Operating System [3].

Real Time Operating System Architecture

A real-time operating system's (RTOS) architecture might change depending on how it is set up and the hardware platform it utilizes [4]. There are, nevertheless, some components and ideas that various RTOS designs have in common. Figure 1 shows the basic architecture of Real Time Operating System (RTOS).

The basic architecture of an RTOS typically consists of three main components:

1. **Kernel:** The operating system's basic component, the kernel, is in charge of managing system resources and offering services to user programmes. This covers I/O operations, job scheduling, memory management, and task synchronization. The kernel is often made to be small, effective, and responsive with little latency and little overhead.
2. **Hardware:** This is the lowest layer of the architecture and this is responsible for coordination with the peripheral devices like CPU, Keyboard etc. This layer also handles

the interrupt and manages time.

3. **User Space:** This layer is the application layer of the architecture and it deals with application programs running e.g., games, browser etc

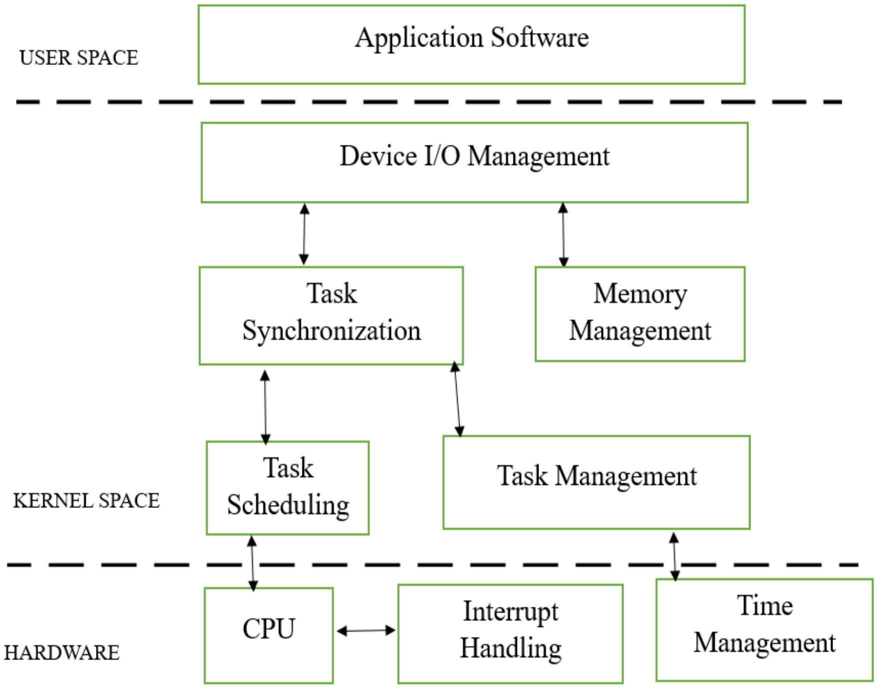


Fig. 1: Architecture of RTOS

Features of Real Time Operating systems (RTOS)

Here are some key facts or features about Real-Time Operating systems (RTOS):

1. RTOS is designed to handle real-time applications where data processing and response time are critical.
2. Unlike general-purpose operating systems, RTOS is optimized for deterministic, predictable behavior and can respond to events in a fixed and predictable amount of time.
3. RTOS supports multiple tasks, each with its own priority and time-slicing algorithm.
4. RTOS provides mechanisms to protect shared resources and handle inter-task communication.
5. RTOS has a small kernel and uses minimal resources, making it suitable for use in embedded systems and other resource-constrained environments.
6. Debugging and testing of RTOS-based systems can be challenging as they have a deterministic behavior and often involve complex interactions between tasks.

Real Time Operating System Examples

Summary of Some Real time operating system that are enumerated as top RTOS by the Embedded System Design:

a) **Message Queue eXecutive (MQX):** MQX - the real-time operating system developed by NXP Semiconductors formerly known as Freescale Semiconductor provides an exceptional level of determinism and reliability within embedded systems. As a result of its impressive performance record- developers have been able to access a highly scalable architecture capable of meeting unique specificities required at any point in time. With compatibility across multiple processor platforms ranging from ARM to PowerPC, ColdFire amongst other popular archetypes; it's no wonder why many experts consider MQX as the preferred choice among top programmers. With regards to its defining characteristics, MQX is particularly notable for its real-time capabilities which enable the operating system to promptly and predictably respond to both events and interrupts. This results in ensuring that time-sensitive tasks are completed accurately, making it an ideal choice for industrial control systems, automotive applications and medical devices [5].

b) **Free RTOS:** It is a RTOS kernel based Operating System that is used in the Embedded System that was originally developed by Richard Barry around 2003 (20 years ago), at Real Time Engineers Ltd. It was an open source model. It was mostly drafted in C Programming Language for easy maintenance. FreeRTOS allows tasks to communicate and synchronize with each other using the queues [6].

FreeRTOS provides the functionality like: Low memory usage, Royalty Free, Low overhead, Powerful execution trace functionality, Multi-threading, Synchronization, Efficient Software timers, Reducing Clock Ticking in low power mode, Task priority, Quad memory allocation scheme, Open source, Large users and developers community, Designed to be simple and easy to use.

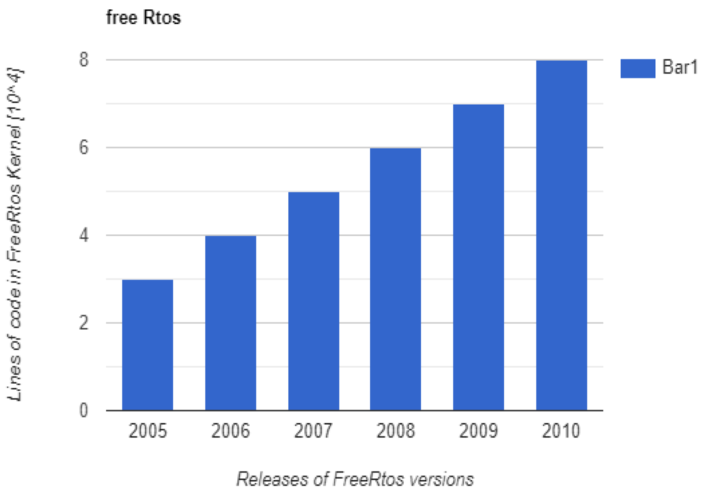


Fig. 2: Versions of FreeRTOS

c) **VxWorks:** The most compatible RTOS, VxWorks is a commercial RTOS developed in the late 1980s for commercial use. Wind River's commercial VxWorks RTOS has the most legitimate use in the embedded industry (used on the International Space Station, for example). It was published in 1987 (36 years ago). Embedded systems come with their unique requirements for reliability and safety - this is precisely where the real-time operating system (RTOS), VxWorks developed by Wind River Systems comes into play. Owned by Intel Corporation, this OS delivers outstanding reliability coupled with optimum security features in a wide range of industrial settings including but not limited to the defense industry, aerospace domain or even automotive manufacture. And what better attribute could it have than being scalable enough to go from small-scale applications up until managing intricate embedded processes quite effortlessly [7]. The available range of processor architectures that can be operated on the system is considerable; it consists of ARM, Intel x86, MIPS, and PowerPC among others.

d) **Window CE:** Window Embedded CE was originally developed by Microsoft in late 1990’s and mainly written in C language. Its application goes from mobile devices to industrial automation systems. It is a real time, lightweight, multithreaded and component-based OS. It is a closed source model of hybrid type kernel [8]. Its initial release was on November 16, 1996 (26 years ago).

Windows CE progression stage moreover has a couple of practical and vital resources for affirm in case the system is meeting nonstop necessities and better examine the structure. For these, the OSBench, ILTiming, and Kernel Tracker are extremely helpful. Windows CE possesses the following distinct features: The RTOS maintained good frequency measurement stability even in the worst conditions; Supported a variety of programming languages, including C/C++, VisualBasic, and C#. Windows CE is a simplified operating system developed by Microsoft for handheld devices and installed systems [9]. The goal of Window CE was to provide a user experience that was comparable to that of workstation versions of Windows, but with a much smaller appearance and less functionality. Windows CE was a well-known choice for manufacturers of PDAs, cell phones, and GPS devices because it supported a wide range of processors, including ARM, MIPS, and x86. Despite its fundamental success, Windows CE has been substantially eliminated from other operating systems like Windows Mobile and Windows Phone. In spite of this, it is as yet a huge piece of Microsoft's set of experiences since it prepared for further developed portable working frameworks to be created [10].

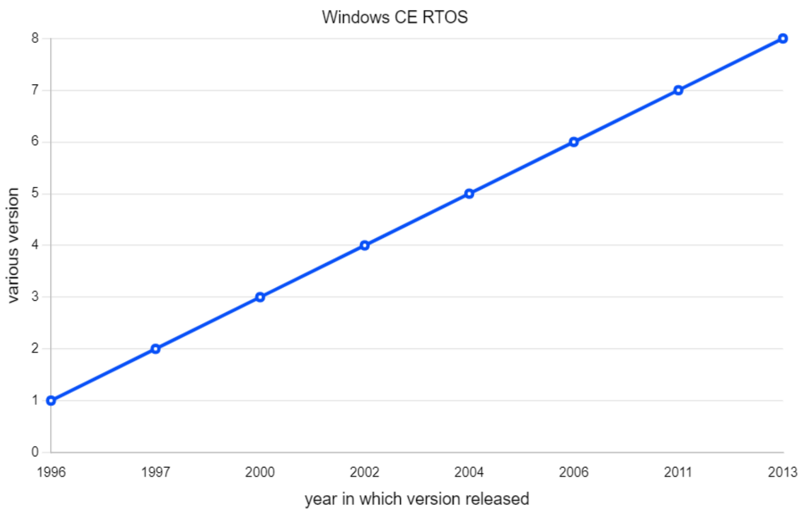


Fig. 3: Versions of Window CE RTOS

The 1.0 form is delivered on November 19,1996 and the Codename is "Pegasus" and "Birch" withe the base ROM of 4MB and least Smash of 2 MB ; Version 2.0 with the codename "Birch," the device name "Palm-size PC," and a 32-bit color screen is released on September 29, 1997; The subsequent version 3.0, codenamed "Cedar" and "Galileo," was released on June 15, 2000. The 4.x version, codenamed "Talisker/McKendrick," was released on January 7, 2002, and it included support for Bluetooth, Flash Memory, and the.Net Compact Framework. The adaptation 5.x was delivered in August 2004 with Codename "Macallan" with Distant Work area Protocol (RDP) support, added Programmed revealing for producers. Version 6.0, codenamed "Yamazaki," was released in September 2006; version 7.0, with Multi-Core CPU support (SMP) and the Media transfer protocol, was released in March 2011. The most recent delivery in June 2013 with DHCPv6 client with stateful/stateless location setup [11].

e) **eCos**: ECOS, or the Embedded Configurable Operating System, is a continuous working framework intended for use in installed frameworks. It is the free and open source constant working framework that was created by eCos people group planned for inserted frameworks and applications which need just a single interaction with numerous strings. It was implemented in C and C++ programming languages, the most important of which was the Assembly language. It was at first delivered in September 1998 (24 years ago).It was at first evolved by eCosCentric Restricted and delivered as open source programming under the GNU Overall population Permit. ECOS supports numerous architectures, including ARM, MIPS, and PowerPC, and offers a wide range of features [12]. Because it is so easily modifiable, developers can select only the components they require for their particular applications, resulting in a smaller memory footprint and lower costs. ECOS has been utilized in a great many applications, including set-top boxes, network switches, and car frameworks. It is known for its dependability, little size, and adaptability, pursuing it a famous decision among inserted frameworks engineers [13].

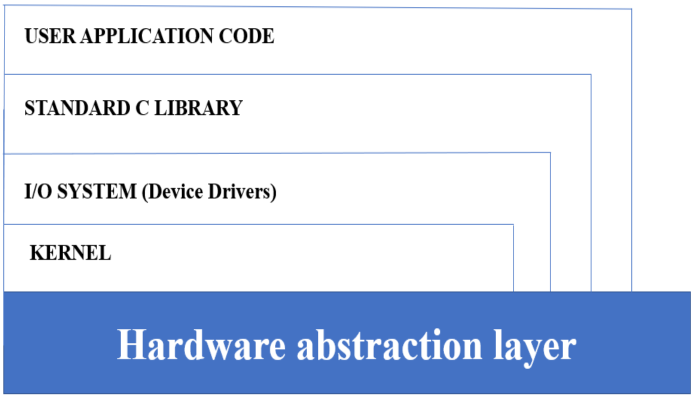


Fig. 4: Ecos layered structure

f) **QNX Neutrino:** QNX Neutrino is a real-time operating system (RTOS) developed by QNX Software Systems, a subsidiary of BlackBerry Limited. It is designed for use in embedded systems, such as automotive infotainment systems, industrial automation systems, medical devices, and aerospace applications. One of the key features of QNX Neutrino is its microkernel architecture, which enables the system to be highly modular and scalable [14]. The microkernel provides only the most essential services, such as process scheduling and inter-process communication, while other services, such as file systems and networking, run as separate processes in user space. This approach allows for greater flexibility, reliability, and security [15]. QNX Neutrino also includes a number of other features that are important for embedded systems, such as support for symmetric multiprocessing (SMP), real-time scheduling, and a range of file systems, including a high-performance flash file system [16]. It also supports a variety of programming languages, including C, C++, Java, and Python.

g) **RTAI:** RTAI is an extension for Real-Time Application Interface. It's the most normally and prevalently utilized RTOS and an ongoing expansion to the Linux portion. With this, you can create programs within the strict time constraints of Linux. Controlling robots, data collection systems, manufacturing facilities, and other time-sensitive machines and equipment are all made possible by it [17]. Additionally, it improves application-usable hard real-time scheduling capabilities and primitives. After that, RTAI was ported to ADEOS, which enabled the RTAI development team to eliminate kernel patching and add real-time There are primarily two parts to RTAI: a Linux kernel patch based on Adeo that adds a hardware abstraction layer and a wide range of services to make real-time programmer's lives easier.

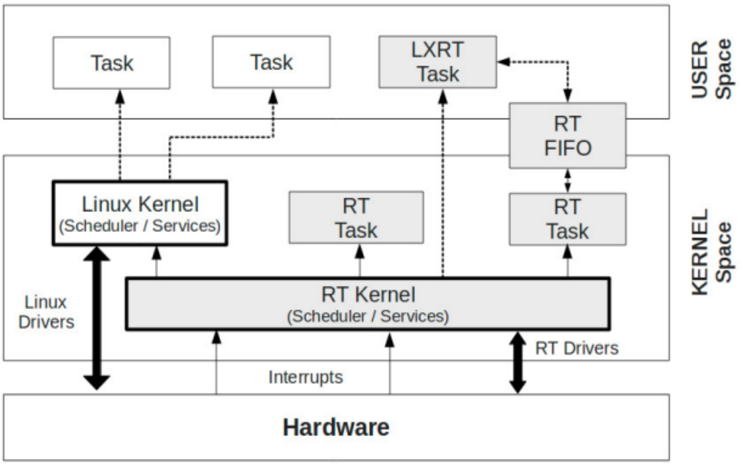


Fig. 5: Linux Real Time Application Interface [18]

Table 1: Comparison Between Different RTOS

RTOS	Scheduling	Memory Management	Interrupt Latency Handler
FreeRTOS	Round-Robin Scheduling	Heap Memory	Application Controlled Deferred Interrupt Handling
eCos	Multi-Level Queue	Static and Dynamic Memory	Interrupt Service Routine & Deferred Service Routine
VxWorks	Round-Robin Scheduling	Supports Virtual Memory	Interrupt Service Routine
RT-Linux	Earliest deadline first scheduler	Real - Addressing Technique	Linux-Kernel handles the interrupt
Window CE	Priority based Time Slice algorithm	Supports Virtual Memory	Interrupt service Routine Library
MQX	Preemptive scheduling	Supports the use of multiple memory pool	Interrupt Service Routine
RTAI	Priority based Scheduling	Real - time memory management	Interrupt handle layer of Standard Linux Kernel

RTAI Linux solves the problem by allowing the development of real-time tasks, using the RTAI Linux tackles the issue by permitting the improvement of ongoing errands, utilizing the RTAI'S hard-constant framework calls from inside the memory - safeguarded space of Linux and under a firm "Continuous help". The RT Kernel's goals are determinism, consistent response time, and low latency. The real time kernel is also known as kernel-rt or preempt-rt [19].

Conclusion:

Real time applications have become popular these days due to the complexity that are present in the system. A RTOS is mainly designed and developed to use in real-time applications where response times are critical, such as in embedded systems, robotics, aerospace, and industrial automation. In comparison to other types of operating systems, RTOS systems have a number of advantages for time-sensitive applications. They offer determinism, scalability, minimal latency, real-time responsiveness, and effective resource management. They might not be appropriate for all applications, but they are a vital tool in many sectors that need high standards of performance and dependability. Real-time responsiveness refers to the ability of a system to execute operations with a guaranteed response time and complete them by set deadlines without interruption from other tasks. For applications like robotics or medical equipment, where quick responses are crucial, minimal latency is necessary. For managing a large number of small, time-sensitive tasks, effective resource management is necessary, and scalability is crucial for embedded systems with constrained hardware resources. For safety-critical systems, where even a minor inaccuracy might have disastrous results, determinism is essential. The developer has the option of using a commercial RTOS (which is used by 44% of developers), an open source RTOS (which is used by 20% of developers), or an internally developed RTOS (which is used by 17% of developers).

References:

1. Chandane, M. P. (2016). Real-time operating systems: A complete overview. *International Journal of Electrical and Electronics Engineering (IJEEE)[e-journal], 8*(1), 361-365.
2. Baskiyar, S., & Meghanathan, N. (2005). A Survey of Contemporary Real-time Operating Systems. *Informatica (03505596), 29*(2).
3. Aroca, R. V., Caurin, G., & Carlos-SP-Brasil, S. (2009). A real-time operating systems (RTOS) comparison. In *WSO-Workshop de Sistemas Operacionais (Vol. 12).*
4. Medina, S., Puig, M. P., & Paniego, J. M. (Year not provided). Real-Time Operating Systems evaluation over Microcontrollers. *Computer Science & Technology Series, 251.*

5. Hambarde, P., Varma, R., & Jha, S. (2014, January). The survey of real-time operating system: RTOS. In *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies* (pp. 34-39). *IEEE.*
6. Chandane, M. P. (2016). Real-time operating systems: A complete overview. *International Journal of Electrical and Electronics Engineering (IJEEE)[e-journal], 8*(1), 361-365.
7. Tao, Z., & Deqing, L. (2013, December). Analysis on task scheduling operating mechanism of embedded real-time operating system MQX. In *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)* (pp. 1844-1847). *IEEE.*
8. Chopra, S., Kumar, D., Singh, G., & Nayyar, A. (July, 2023). User Interface: An Overview, *Dr. Rohit Srivastava, Research Trends in Science and Technology, Vol-I,* pp. 1-9, Bhumi Publishing House, India.
9. Chopra, S., Kaur, L., Nayyar, A., & Singh, G. (July, 2023). Review on Application Programming Interfaces for Internet Protocol. *Dr. Alok Ranjan Sahu, Research Trends in Science and Technology, Vol. III,* pp.54-63, Bhumi Publishing House, India.
10. Netter, C. M., & Baceller, L. F. (2001, May). Assessing the real-time properties of Windows CE 3.0. In *Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. ISORC 2001* (pp. 179-184). *IEEE.*
11. Lohmann, D., Scheler, F., Tartler, R., Spinczyk, O., & Schröder-Preikschat, W. (2006, April). A quantitative analysis of aspects in the eCos kernel. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006* (pp. 191-204).
12. She, S., Lotufo, R., Berger, T., Wąsowski, A., & Czarnecki, K. (2011, May). Reverse engineering feature models. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 461-470).
13. Kaur, B., Chopra, S., & Singh, G. (May, 2023). Comparison of Windows and Linux and their Server Configuration. *Dr. Vanitha G. Ramesh, Trends in Interdisciplinary Research Volume II,* pp. 68-76, Bhumi Publishing House, India.
14. Sygulla, F., Wittmann, R., Seiwald, P., Berninger, T., Hildebrandt, A. C., Wahrmann, D., & Rixen, D. (2018, August). An EtherCAT-based real-time control system architecture for humanoid robots. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)* (pp. 483-490). *IEEE.*
15. Netter, C. M., & Baceller, L. F. (2001, May). Assessing the real-time properties of Windows CE 3.0. In *Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. ISORC 2001* (pp. 179-184). *IEEE.*

16. Chopra, S., Rani, D., Sharma, R., & Kaur, S. (July, 2023). Process Management in Operating System. *Dr. Alok Ranjan Sahu, Research Trends in Science and Technology, Vol. III,* pp. 64-74, Bhumi Publishing House, India.
17. Hambarde, P., Varma, R., & Jha, S. (2014, January). The survey of real-time operating system: RTOS. In *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies* (pp. 34-39). *IEEE.*
18. Arm, J., Bradac, Z., & Kaczmarczyk, V. (2016). Real-time capabilities of Linux RTAI. *IFAC-PapersOnLine, 49*(25), 401-406.
19. Kumar, P. L. (2012). An Adaptive Framework for the Selection of Embedded Operating Systems. *International Journal of Scientific and Research Publications, 2*(4).

ANALYSIS OF COMPLETELY RANDOMIZED DESIGN (CRD) AND RANDOMIZED BLOCK DESIGN(RBD) USING MS- EXCEL

Malati C. Yeola

Department of Statistics

Ahmednagar College, Ahmednagar, Maharashtra, India

Corresponding author E-mail: mcyeola@gmail.com, malati.yeola@aca.edu.in

Abstract:

In this book chapter, how to solve ANOVA for CRD and RBD using MS-EXCEL's Data Analysis tool is explained in detail. Step by step procedure to perform ANOVA for CRD and RBD explained by taking one example for each of the design.

Keywords: Analysis of Variance (ANOVA), Completely Randomized Design (CRD), Randomized Block Design (RBD)

Introduction:

1. Analysis of Completely Randomized Design (CRD) Using MS- EXCEL

The simplest and most flexible design is completely randomized design. It is based on the prin ciples of randomization and replication. In this design treatments are allocated at random to the experimental units over the entire experimental material. Let us suppose that we have t treatments, the i^{th} treatment being replicated n_i times, $i = 1, 2, \cdot\cdot\cdot t$. Then the whole experimental material is divided into $_{n} = \mathrm{P}^{t}_{i=1}\, n_i$ experimental units and the treatments are distributed completely at random over the units subject to the condition that the i^{th} treatment occurs n_i times.

Analysis of CRD using MS-EXCEL

1. Enter the data.

2. Go in the tab data then data analysis as shown in the following figure 1.

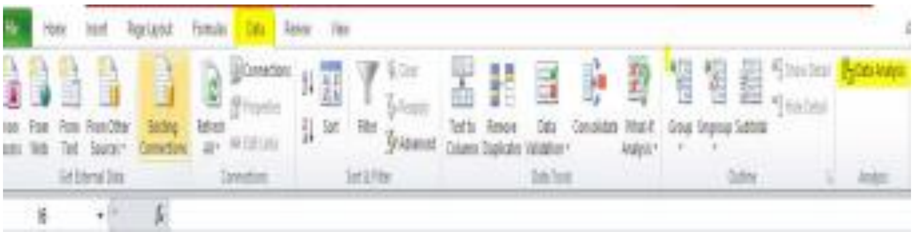


Figure 1

3. In data analysis select the analysis tool as ANOVA: Single Factor as shown in figure 2a. click on ok. you will get window like this (as shown in the following figure 2b)

4. Select the data in the window Input Range(2c)

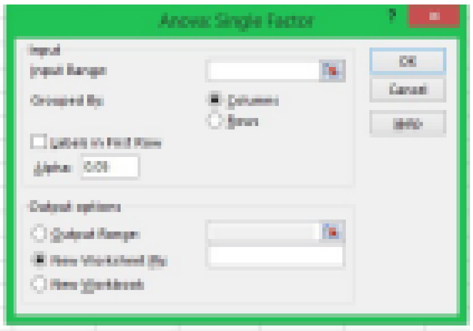


Figure 2 (a)

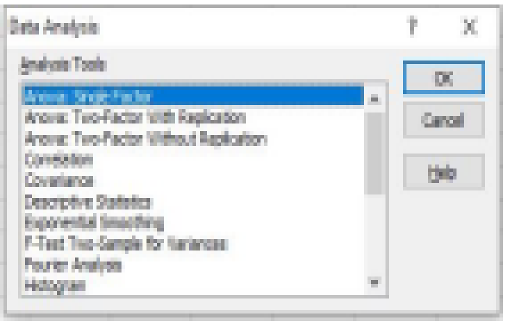


Figure 2 (b)

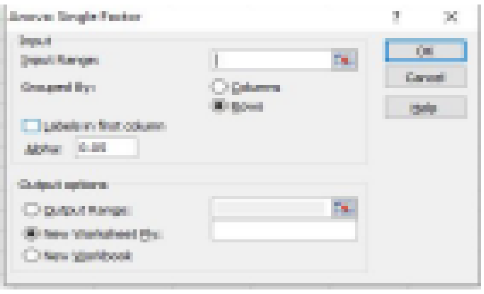
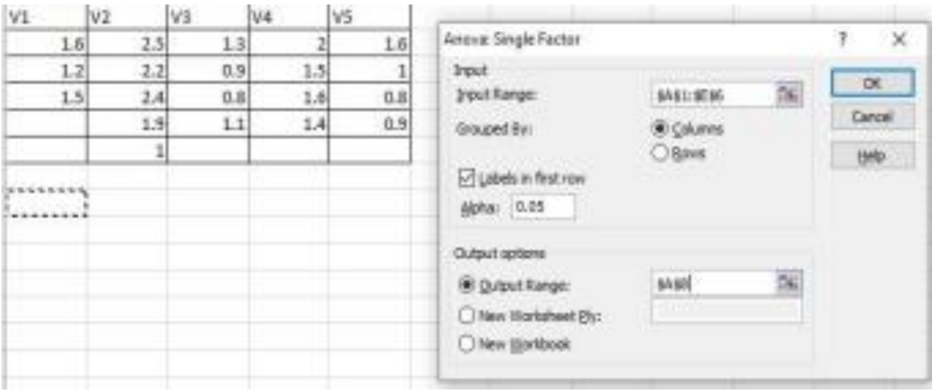


Figure 2 (c)

5. If data entered in the excel sheet treatments column wise then in the above appeared window, in front of Grouped By: select Columns as shown in figure 2b otherwise select Rows as shown in figure 2c.

6. While selecting data if its labels are also selected then click the box Labels in the first row as shown in the following figure 3



V1	V2	V3	V4	V5
1.6	2.5	1.3	2	1.6
1.2	2.2	0.9	1.5	1
1.5	2.4	0.8	1.6	0.8
	1.9	1.1	1.4	0.9
	1			

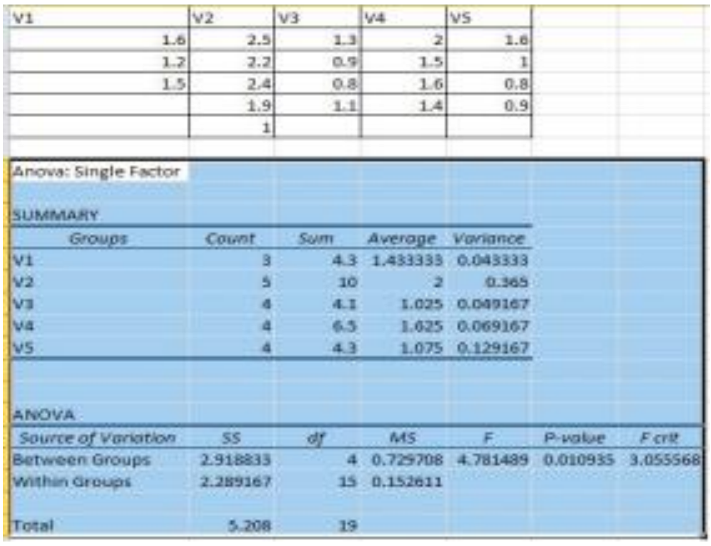
Figure 3

7. Also select the output range if we want output on the same sheet. By default it will be selected new worksheet.

8. After that click on OK. We will get the output as shown in the figure 4

9. Column 2 shows sum of squares(SS). Here between groups is treatment sum of squares and within groups is error sum of squares. Total shows Total sum of squares.

10. By observing the table in figure 3 (column of p value) we conclude that reject H_0 at 5% level of significance because p-value is less than 0.05.



V1	V2	V3	V4	V5
1.6	2.5	1.3	2	1.6
1.2	2.2	0.9	1.5	1
1.5	2.4	0.8	1.6	0.8
	1.9	1.1	1.4	0.9
	1			

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
V1	3	4.3	1.433333	0.043333
V2	5	10	2	0.365
V3	4	4.1	1.025	0.049167
V4	4	6.5	1.625	0.069167
V5	4	4.3	1.075	0.129167

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	2.918833	4	0.729708	4.781489	0.010935	3.055568
Within Groups	2.289167	15	0.152611			
Total	5.208	19				

Figure 4

2. Analysis of Randomized Block Design (RBD) Using MS EXCEL

When an experimental area (material) is not homogeneous, then variability of the experimental material is controlled by grouping or stratifying the whole material into relatively homogeneous strata or subgroups (known as blocks or replicates). If the treatments are applied at random to rel atively homogeneous units within each strata or block and replicated overall the blocks, the design is a Randomized Block Design (R.B.D).

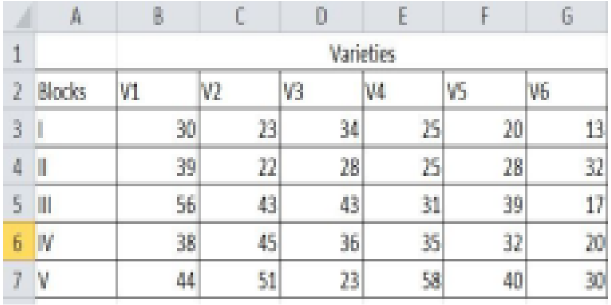
Example A varietal trial was conducted at research station. The design adopted for the same was five randomized blocks of 6 plots each. The yields in lb. per plot (of 1/20 th of an acre) obtained from the experiment are as given in the following table:

Blocks	**Varieties** V_1 V_2 V_3 V_4 V_5 V_6
I	30 23 34 25 20 13
II	39 22 28 25 28 32
III	56 43 43 31 39 17
IV	38 45 36 35 32 20
V	44 51 23 58 40 30

Analyse the data completely.

1. Enter the data as shown in the following figure 5a

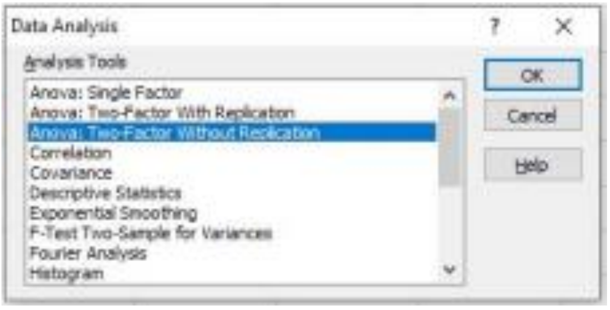
2. Go in the tab data then data analysis as shown if figure 5b.



	A	B	C	D	E	F	G
1		Varieties					
2	Blocks	V1	V2	V3	V4	V5	V6
3	I	30	23	34	25	20	13
4	II	39	22	28	25	28	32
5	III	56	43	43	31	39	17
6	IV	38	45	36	35	32	20
7	V	44	51	23	58	40	30

Figure 5 (a) **Figure 5 (a) (b)**

3. In data analysis select the analysis tool as ANOVA:Two- Factor Without Replication. Click on ok. you will get window like this (as shown in the following figure 6)



Data Analysis

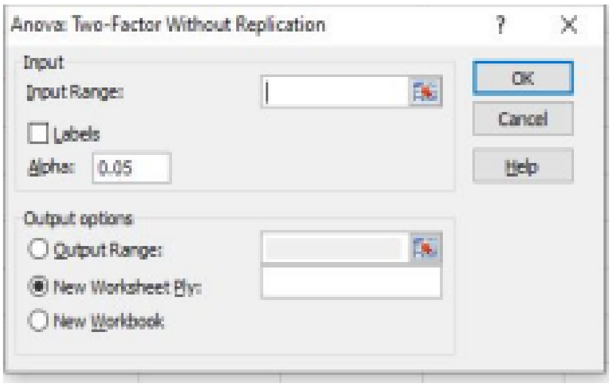
Analysis Tools

- Anova: Single Factor
- Anova: Two-Factor With Replication
- Anova: Two-Factor Without Replication
- Correlation
- Covariance
- Descriptive Statistics
- Exponential Smoothing
- F-Test Two-Sample for Variances
- Fourier Analysis
- Histogram

Figure 6

4. Select the entered data in the input range as shown in figure 7.

5. If you want any other level of significance other than 0.05, change the value in front of Alpha. Otherwise keep it as it is as shown in figure 7.



Anova: Two-Factor Without Replication

Input

Input Range:

□ Labels

Alpha: 0.05

Output options

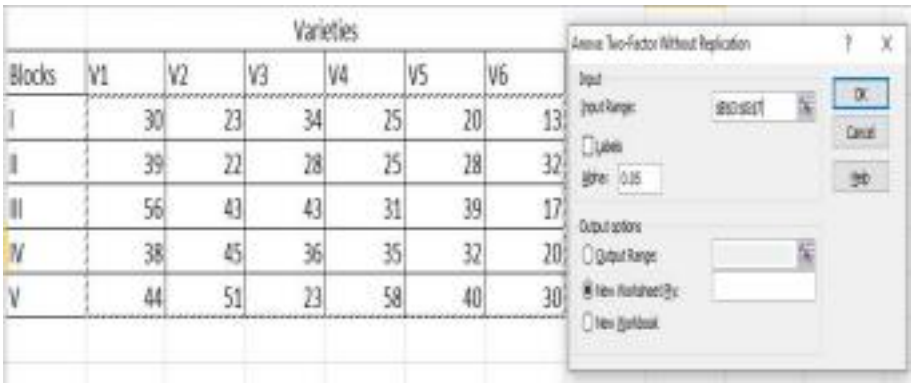
○ Output Range:

● New Worksheet Ply:

○ New Workbook

Figure 7

6. Select the output range by clicking in the any empty cell of the same sheet. If not selected any empty cell, by default it will show output on New worksheet as shown in figure 8.



	Varieties					
Blocks	V1	V2	V3	V4	V5	V6
I	30	23	34	25	20	13
II	39	22	28	25	28	32
III	56	43	43	31	39	17
IV	38	45	36	35	32	20
V	44	51	23	58	40	30

Anova: Two-Factor Without Replication

Input

Input Range: [illegible]

☐ Labels

Alpha: 0.05

Output options

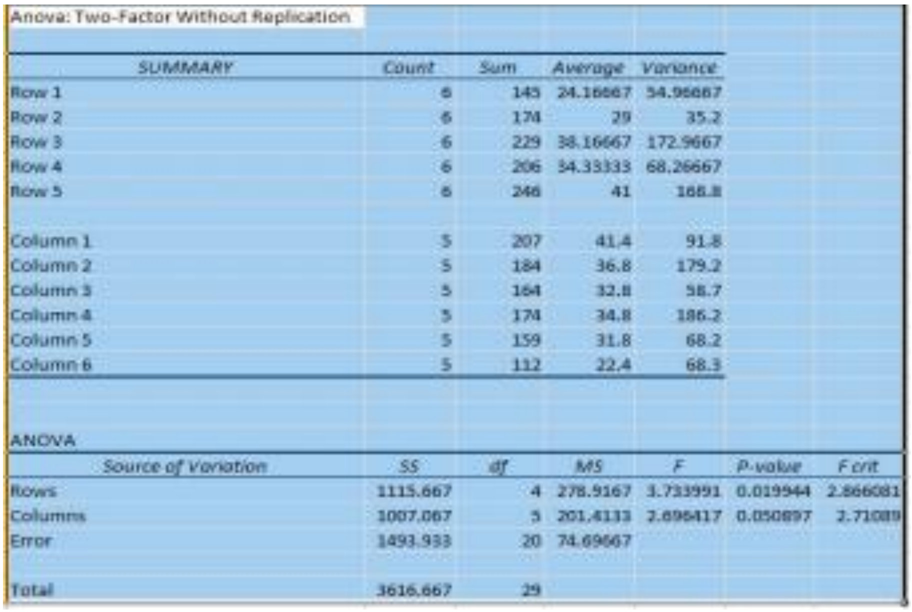
○ Output Range:

● New Worksheet Ply:

○ New Workbook

Figure 8

7. In ANOVA table Rows denotes for this example as blocks therefore here Rows represent as blocks. And columns as treatments. The output will be appeared as shown in figure 9



Anova: Two-Factor Without Replication

SUMMARY	Count	Sum	Average	Variance
Row 1	6	145	24.16667	54.96667
Row 2	6	174	29	35.2
Row 3	6	229	38.16667	172.9667
Row 4	6	206	34.33333	68.26667
Row 5	6	246	41	168.8
Column 1	5	207	41.4	91.8
Column 2	5	184	36.8	179.2
Column 3	5	164	32.8	58.7
Column 4	5	174	34.8	186.2
Column 5	5	159	31.8	68.2
Column 6	5	112	22.4	68.3

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Rows	1115.667	4	278.9167	3.733991	0.019944	2.866081
Columns	1007.067	5	201.4133	2.696417	0.050897	2.71089
Error	1493.933	20	74.69667			
Total	3616.667	29				

Figure 9

8. Applying the decision rule of reject H_0 if p-value less than level of significance otherwise accept H_0, the decision may be taken accordingly.

References:

1. Gupta S.C. and Kapoor V.K. (2006). Fundamentals of Applied Statistics, S. Chand Sons, New Delhi

2. Montgomery, D.C. (2012). Design and Analysis of Experiments, John Wiley and sons Inc., New Delhi.

EXPLORING RESEARCH METHODOLOGY IN COMPUTER ENGINEERING

Suraj Shankarrao Damre[1] and Bharat Madhavrao Pawar[2]

[1]Department of Computer Engineering

M. S. Bidve Engineering College, Barshi Rd, Pakharsangvi, Latur, Maharashtra

[2]Department of Computer Engineering

Vilasrao Deshmukh Foundation Group of Institutions,

Near to Manjara Sugar, Additional MIDC, Barshi Rd, Latur, Maharashtra

Corresponding author E-mail: surajdamre@gmail.com, bharatmpawar0206@gmail.com

Abstract:

Research methodology serves as the cornerstone of systematic inquiry in computer engineering, guiding scholars and practitioners in their pursuit of innovative solutions and advancements. This paper provides an in-depth exploration of research methodology within the realm of computer engineering, shedding light on its fundamental concepts, various approaches, and practical application. By delving into the intricacies of research design, data collection, analysis, and ethical considerations, this paper equips researchers with a comprehensive understanding of how to navigate the complex landscape of computer engineering research.

Keywords: Research methodology, Computer engineering, Research process, Types of research, Quantitative methods, Qualitative methods, Data collection, Data analysis, Ethical considerations

Introduction:

Research methodology forms the cornerstone of advancements in computer engineering, guiding scholars and practitioners in the systematic pursuit of knowledge and innovation. This chapter delves into the intricacies of research methodology, highlighting its significance, key components, and application within the realm of computer engineering.

The Research Process: A Comprehensive Exploration

Understanding the research process is paramount for conducting meaningful studies in computer engineering. This section elucidates the stages of research, from problem identification and literature review to hypothesis formulation, data collection, analysis, and interpretation. It also underscores the iterative nature of research and the importance of refining methodologies as insights unfold.

The research process constitutes the systematic journey undertaken by scholars and practitioners in the realm of computer engineering to uncover new knowledge, address questions, and contribute to the ever-evolving field. This multifaceted endeavor involves a series of interconnected stages, each playing a pivotal role in shaping the outcomes and significance of the research study. The following discussion provides an in-depth expansion of the research process, elucidating its key stages and their significance within the context of computer engineering research.

1. Problem Identification and Formulation:

At the inception of the research process lies the critical task of identifying a research problem that holds relevance and significance within the domain of computer engineering. This entails recognizing gaps in existing knowledge, technological challenges, or areas where innovations are sought. The problem statement serves as the compass guiding the entire research journey, defining the scope, objectives, and boundaries of the study.

2. Literature Review:

A thorough literature review is an indispensable stage that involves an exhaustive exploration of existing research, scholarly works, and technological advancements relevant to the chosen problem. This step aids researchers in understanding the current state of knowledge, identifying key contributors, and discerning gaps or opportunities for further investigation. By building on existing insights and identifying areas where the study can contribute, the literature review shapes the research's foundation.

3. Hypothesis Formulation or Research Questions:

With insights garnered from the literature review, researchers proceed to formulate hypotheses or research questions that encapsulate the focal points of the study. Hypotheses represent informed assumptions or predictions about relationships between variables, while research questions seek to explore specific aspects of the research problem. These hypotheses or questions guide subsequent data collection, analysis, and interpretation.

4. Data Collection:

Data collection involves the systematic gathering of relevant information, measurements, observations, or experiences that pertain to the research problem. In computer engineering, this may encompass data from simulations, experiments, surveys, interviews, or sensor networks. Rigorous data collection ensures that the study is grounded in empirical evidence, enabling researchers to draw meaningful conclusions and insights.

5. Data Analysis:

The amassed data undergoes rigorous analysis to unearth patterns, trends, and relationships. In computer engineering, data analysis often involves applying statistical methods, algorithms, and software tools to extract meaningful insights from complex datasets. This stage converts raw data into actionable information, enabling researchers to address their research questions or test hypotheses.

6. Interpretation and Conclusion:

The interpretation phase involves making sense of the analyzed data, drawing connections between findings and the research problem. Researchers discern implications, draw conclusions, and offer insights that contribute to the broader knowledge landscape. This phase may also involve reflecting on the limitations of the study, potential sources of bias, and avenues for further exploration.

7. Iterative Nature of Research:

Importantly, the research process is iterative, not linear. Researchers often revisit and refine various stages based on emerging insights, new data, or unexpected findings. This iterative nature allows for the dynamic evolution of the study, fostering deeper understanding and refinement of the research objectives.

Types of Research in Computer Engineering: An In-depth Exploration

Computer engineering research encompasses diverse methodologies, each tailored to specific objectives. This segment elucidates various research paradigms, such as experimental, exploratory, descriptive, and case study research. Additionally, it delves into qualitative and quantitative research methods, highlighting their applicability and benefits.

1. Experimental Research:

Experimental research involves the systematic manipulation of variables to establish cause-and-effect relationships. In computer engineering, this may entail conducting controlled experiments to test the impact of specific interventions, algorithms, or configurations on system performance. Experimental research contributes to a deeper understanding of how changes in variables influence outcomes, providing valuable insights for optimizing computer systems and technologies.

2. Exploratory Research:

Exploratory research aims to delve into novel or uncharted territories within computer engineering. Researchers use this approach to gain preliminary insights, generate hypotheses, and identify potential research directions. Exploratory studies are particularly valuable for identifying emerging trends, innovative technologies, and new avenues for investigation in rapidly evolving areas of computer engineering

3. Descriptive Research:

Descriptive research seeks to provide a comprehensive and detailed account of specific phenomena within computer engineering. This type of research often involves surveys, observations, or case studies to capture and analyze real-world scenarios, trends, or patterns. Descriptive research contributes to the documentation and analysis of existing practices, enabling researchers to draw informed conclusions and make data-driven recommendations.

4. Case Study Research:

Case study research entails an in-depth analysis of a specific case, system, or phenomenon within computer engineering. Researchers closely examine real-world instances to gain a deeper understanding of complex interactions, challenges, and outcomes. Case studies offer valuable insights into practical applications, problem-solving approaches, and the contextual factors that shape computer engineering practices.

5. Qualitative Research:

Qualitative research focuses on understanding the underlying motivations, perspectives, and experiences of individuals or groups within the realm of computer engineering. Techniques such as interviews, focus groups, and content analysis are employed to uncover nuances, values, and contextual factors that influence behaviors and decisions. Qualitative research adds depth and richness to the exploration of human aspects in technology development and adoption.

6. Quantitative Research:

Quantitative research involves the collection and analysis of numerical data to quantify relationships, trends, and patterns within computer engineering. Researchers use statistical methods to derive objective insights, validate hypotheses, and make data-driven predictions. Quantitative research is instrumental in assessing the performance, scalability, and efficiency of computer systems and algorithms.

7. Action Research:

Action research bridges the gap between theoretical knowledge and practical application in computer engineering. Researchers collaborate with practitioners to identify challenges, implement interventions, and assess outcomes in real-world settings. This iterative process fosters continuous improvement, enhances problem-solving capabilities, and contributes to the refinement of computer engineering practices.

Selecting Research Methods in Computer Engineering: A Strategic Approach

The selection of appropriate research methods is a critical decision in computer engineering research. This section outlines the criteria for choosing methodologies based on research goals, nature of the problem, available resources, and ethical considerations. It explores the use of surveys, experiments, simulations, interviews, and observations, among others, as tools for data collection.

The process of selecting research methods in computer engineering is a crucial and intricate endeavor, shaping the trajectory and outcomes of the entire research study. This section delves into the nuanced considerations, criteria, and strategic thinking involved in the selection of research methods, equipping researchers with insights to make informed decisions tailored to their specific research objectives and contexts.

1. Alignment with Research Objectives:

Central to the selection of research methods is a clear alignment with the overarching research objectives. Researchers must critically assess the goals of their study, whether they aim to explore new phenomena, validate hypotheses, or provide practical solutions within computer engineering. A careful examination of research objectives ensures that the chosen methods are attuned to the desired outcomes.

2. Nature of the Research Problem:

The complexity, scope, and nature of the research problem play a pivotal role in determining the most suitable research methods. For instance, if the research problem involves understanding user experiences with a new interface design, qualitative methods such as interviews or usability testing may be preferred. Conversely, if the problem involves analyzing large datasets for performance optimization, quantitative methods like data analytics or simulations may be more appropriate.

3. Availability of Resources:

Researchers must consider the resources available, including time, funding, equipment, and expertise, when selecting research methods. Some methods may require specialized software, hardware, or personnel, which may influence the feasibility of their adoption. Balancing the available resources with the chosen methods ensures a pragmatic and realistic research approach.

4. Ethical Considerations:

Ethical implications and considerations are integral to research methodology selection. Researchers must ensure that the chosen methods align with ethical guidelines and respect the rights and privacy of participants, especially when involving human subjects. Ethical transparency and compliance are paramount in maintaining the integrity and credibility of the research study.

5. Data Collection and Analysis Techniques:

The nature of data to be collected and analyzed influences the selection of methods. For instance, if the research involves capturing user interactions with a software application, methods such as user tracking, logging, or surveys may be employed. Similarly, the analysis of qualitative data may involve content analysis or thematic coding, while quantitative data may require statistical analysis.

6. Validity and Reliability:

The validity and reliability of research findings hinge on the robustness of the chosen methods. Researchers must assess the extent to which the selected methods can yield accurate and consistent results. Employing methods with established validity and reliability enhances the trustworthiness of the study's outcomes.

7. Practicality and Feasibility:

Practicality and feasibility considerations involve evaluating the ease of implementing selected research methods. Researchers need to assess whether the chosen methods can be executed within the available timeframe, resources, and constraints. A method's practicality ensures that the study's objectives can be achieved effectively.

8. Innovation and Exploration:

Innovative research often calls for unconventional or emerging research methods. Researchers exploring uncharted territories within computer engineering may opt for methods that enable creative problem-solving, facilitate groundbreaking insights, or capitalize on cutting-edge technologies.

Data Collection and Analysis in Computer Engineering Research: An In-depth Exploration

Data collection and analysis form the crux of empirical inquiry in computer engineering research, enabling scholars to transform raw information into meaningful insights and informed conclusions. This section delves into the intricacies of data collection and analysis within the context of computer engineering, elucidating the methodologies, techniques, and considerations that underpin this crucial phase of the research process.

1. Data Collection Methodologies:

Data collection methodologies encompass a spectrum of techniques employed to gather relevant information for analysis. In computer engineering research, data collection methods vary based on the research objectives and the nature of the data sought. Common methodologies include surveys, experiments, simulations, observations, interviews, focus groups, and sensor networks. Each methodology serves as a lens through which researchers gain access to diverse types of data, ranging from user interactions and system performance to software behavior and network traffic.

2. Instrumentation and Measurement:

Instrumentation involves the design and implementation of tools, sensors, or instruments to capture data accurately and consistently. In computer engineering, this may entail creating software interfaces, deploying hardware sensors, or developing specialized data collection platforms. Measurement techniques ensure that data is quantified, standardized, and suitable for analysis. Precise instrumentation and measurement practices are essential to generating reliable and valid data for subsequent analysis.

3. Big Data and Analytics:

The advent of big data has revolutionized data collection and analysis in computer engineering research. Massive datasets, generated from sources like social media, IoT devices, and online platforms, offer unprecedented insights when subjected to advanced analytics techniques. Data mining, machine learning, and artificial intelligence are instrumental in extracting patterns, trends, and correlations from large-scale data, enriching computer engineering research with valuable insights.

4. Qualitative Data Collection and Analysis:

Qualitative data collection methods, such as interviews and focus groups, facilitate the exploration of human experiences, behaviors, and perceptions in computer engineering contexts. Qualitative analysis techniques, including thematic analysis and content analysis, help researchers uncover underlying themes, motivations, and contextual factors that influence technology adoption, user preferences, and system usability.

5. Quantitative Data Collection and Analysis:

Quantitative data collection employs numerical measurements to quantify relationships, trends, and patterns. Surveys, experiments, and simulations generate structured data suitable for statistical analysis. Quantitative analysis techniques involve applying statistical tests, correlation analyses, regression models, and data visualization to derive meaningful insights and test hypotheses within computer engineering research.

6. Ethical Considerations in Data Collection:

Ethical considerations are paramount in data collection, especially when involving human participants. Researchers must ensure informed consent, privacy protection, and data anonymization to uphold ethical standards. Transparent communication with participants and adherence to ethical guidelines safeguard the rights and well-being of those contributing to the research.

7. Data Validation and Reliability:

Data validation encompasses processes to ensure the accuracy, completeness, and integrity of collected data. Validation techniques range from cross-referencing data sources to implementing error checks during data collection. Reliability checks ascertain the consistency and stability of collected data over time and across different contexts, enhancing the robustness of research findings.

8. Data Interpretation and Visualization:

Interpreting and making sense of collected data is a pivotal phase. Researchers employ various techniques to analyze and synthesize data, revealing trends, outliers, and relationships. Data visualization tools, such as graphs, charts, and heatmaps, aid in conveying complex findings intuitively, enabling researchers to communicate insights effectively.

Detailing the collection and analysis of data is crucial for robust research outcomes. This part expounds on techniques for gathering data, including sensor networks, instrumentation, and data mining. It also discusses data preprocessing, statistical analysis, and software tools used for interpreting results.

Ethical Considerations in Computer Engineering Research: Navigating the Ethical Landscape

Ethics underpin responsible research practices. This section delves into ethical considerations related to data privacy, intellectual property, human subjects, and transparency in computer engineering research. It emphasizes the significance of adhering to ethical guidelines to maintain the integrity and credibility of research endeavours. Ethical considerations hold paramount significance in the realm of computer engineering research, guiding scholars and practitioners in upholding integrity, protecting participants' rights, and ensuring responsible conduct. This section delves into the multifaceted dimensions of ethical considerations in computer engineering research, elucidating the principles, challenges, and strategies that underpin ethical decision-making.

1. Informed Consent:

Informed consent is a cornerstone of ethical research practice. Researchers must ensure that participants fully comprehend the purpose, procedures, potential risks, and benefits of the study before providing consent to participate. In computer engineering research, this involves transparently explaining how data will be collected, used, and stored, especially when involving user interactions, personal information, or sensitive technologies.

2. Privacy and Data Protection:

Protecting participants' privacy and data security is imperative, particularly in the era of digitization and data sharing. Researchers must implement robust measures to safeguard sensitive information, employ encryption techniques, and adhere to data protection regulations. In computer engineering, these entails addressing concerns related to data breaches, unauthorized access, and ensuring that participants' identities remain confidential.

3. Minimization of Harm:

Researchers have an ethical obligation to minimize potential harm to participants. In computer engineering, this extends to minimizing risks associated with technology use, software testing, or system experimentation. Mitigating harm may involve rigorous testing, implementing safety measures, and providing clear guidelines to participants on using technology safely and responsibly.

4. Beneficence and Non-Maleficence:

The principles of beneficence (maximizing benefits) and non-maleficence (minimizing harm) guide researchers in promoting the well-being of participants. In computer engineering research, this translates to developing technologies that enhance user experiences, contribute to societal betterment, and prioritize user safety. Researchers should also anticipate and address potential negative impacts of technology and take proactive measures to prevent harm.

5. Transparency and Full Disclosure:

Transparent communication with participants is essential to ethical research. Researchers should provide comprehensive and accurate information about the study's objectives, methodologies, potential risks, and potential benefits. In computer engineering, this involves clearly articulating how data will be collected, processed, and utilized, enabling participants to make informed decisions.

6. Research Integrity and Honesty:

Maintaining research integrity and honesty is paramount in upholding ethical standards. Researchers must accurately represent their findings, avoid fabricating or manipulating data, and acknowledge sources appropriately. In computer engineering, this entails ensuring the accuracy of algorithms, models, and experimental results, and acknowledging any limitations or uncertainties.

7. Fair Treatment and Equity:

Researchers must ensure equitable treatment of participants, avoiding discrimination, bias, or undue influence. In computer engineering, this involves promoting inclusivity in technology design, considering diverse user needs, and avoiding the reinforcement of existing inequalities or biases in algorithms or software systems.

8. Institutional Review Board (IRB) Approval:

Many research institutions require ethical oversight through an Institutional Review Board (IRB). Researchers seeking approval for their studies must demonstrate compliance with ethical guidelines and obtain IRB approval before commencing research activities. In computer engineering, this ensures that ethical considerations are rigorously evaluated and upheld throughout the research process.

Case Studies in Computer Engineering Research: A Deeper Exploration

Case studies serve as powerful tools in computer engineering research, offering researchers an opportunity to delve into real-world scenarios, analyze complex interactions, and gain in-depth insights into technological phenomena. This section delves into the multifaceted dimensions of case studies within computer engineering research, shedding light on their characteristics, methodologies, and significance in advancing knowledge and innovation.

1. Defining Case Studies:

A case study involves an intensive and in-depth investigation of a specific instance, system, process, or phenomenon within the realm of computer engineering. It seeks to explore, understand, and explain the intricacies, challenges, and dynamics of real-world situations, often involving interactions between technology, users, and the environment.

2. Types of Case Studies:

Case studies in computer engineering encompass a wide range of contexts and focus areas. They can range from single-case studies, which deeply explore a singular instance, to multiple-case studies, which compare and contrast multiple instances or variations. Case studies can be illustrative, explanatory, exploratory, or evaluative, depending on the research objectives.

3. Methodological Approaches:

Researchers can adopt various methodological approaches when conducting case studies in computer engineering research. These approaches may include:

- Exploratory Case Studies: Uncover new insights, trends, or patterns in emerging technologies, user behaviors, or system interactions.
- Explanatory Case Studies: Seek to explain causal relationships, often by investigating how and why certain phenomena occur.
- Descriptive Case Studies: Document and provide a comprehensive account of specific technological implementations, user experiences, or system behavior.
- Comparative Case Studies: Compare and contrast multiple cases to identify similarities, differences, or patterns across different contexts.

4. Data Collection Methods:

Data collection in case studies often involves a combination of techniques. Researchers may employ interviews, observations, surveys, documentation analysis, user feedback, and performance metrics to gather a rich and diverse set of data. Data collection methods are chosen based on the research questions, the nature of the case, and the desired depth of analysis.

5. Data Analysis and Interpretation:

Data analysis in case studies is typically qualitative in nature, aiming to uncover underlying themes, relationships, and insights. Researchers engage in coding, categorization, pattern recognition, and content analysis to derive meaningful interpretations. The goal is to develop a comprehensive understanding of the case's intricacies and extract knowledge that contributes to the field.

6. Value and Significance:

Case studies hold several key advantages in computer engineering research. They provide researchers with a deep and contextualized understanding of complex phenomena, allowing for the exploration of dynamic interactions between technology and users. Case studies also facilitate the testing and refinement of theoretical frameworks, the identification of practical implications, and the generation of hypotheses for further investigation.

7. Practical Applications:

Case studies in computer engineering research have practical applications across various domains. They inform the design and development of user-centric technologies, guide the optimization of system performance, and contribute to the resolution of real-world challenges in areas such as cyber security, human-computer interaction, software engineering, and network analysis.

8. Challenges and Considerations:

While powerful, case studies also present challenges, including potential bias, limited generalizability, and subjectivity in data interpretation. Researchers must address these challenges through rigorous data collection, transparency in methodologies, and triangulation of findings to enhance the credibility of their research.

Illustrating the application of research methodologies, this section presents case studies from diverse subfields of computer engineering. These cases showcase how specific methodologies were employed to address research questions, highlighting the challenges faced and insights gained.

Future Trends in Computer Engineering Research Methodology: Anticipating Innovation and Advancement

As computer engineering continues to evolve, so does its research methodology. This section offers insights into emerging trends such as interdisciplinary research, machine learning, and big data analytics. It explores how these trends reshape research practices and pave the way for innovative approaches. The landscape of computer engineering research methodology is poised for transformative shifts, driven by technological advancements, evolving research paradigms, and emerging challenges. This section delves into the potential future trends in computer engineering research methodology, shedding light on the directions that researchers are likely to explore in the pursuit of cutting-edge knowledge and innovation.

1. Interdisciplinary Collaborations:

Future trends in computer engineering research methodology are expected to emphasize interdisciplinary collaborations that bridge the gap between computer science, engineering, and other domains. Researchers will increasingly collaborate with experts in fields such as psychology, sociology, medicine, and environmental science to address complex societal challenges and develop holistic technological solutions.

2. Integration of Ethical AI and Bias Mitigation:

As artificial intelligence (AI) continues to shape the computer engineering landscape, future research methodology will prioritize the integration of ethical considerations and bias mitigation techniques. Researchers will focus on developing robust methods to ensure fairness, transparency, and accountability in AI algorithms, aiming to prevent unintended biases and discriminatory outcomes.

3. Human-Centered Design and User Experience:

The future of computer engineering research methodology will place greater emphasis on human-centered design principles and user experience evaluation. Researchers will adopt methodologies to deeply understand user needs, preferences, and behaviors', ensuring that technology design aligns with the expectations and well-being of users.

4. Embracing Mixed-Methods Approaches:

Future trends in computer engineering research methodology will witness the integration of mixed-methods approaches that combine quantitative and qualitative techniques. Researchers will leverage the strengths of both methodologies to gain comprehensive insights, triangulate findings, and validate research outcomes across diverse dimensions.

5. Emphasis on Explainable AI and Trustworthiness:

As AI systems become more complex and integrated into various applications, research methodology will evolve to focus on explainable AI (XAI) and trustworthiness assessment. Researchers will develop methodologies to interpret AI decisions, generate human-understandable explanations, and ensure that AI technologies are transparent, accountable, and reliable.

6. Real-Time Data Analytics and Edge Computing:

With the rise of edge computing and the Internet of Things (IoT), future research methodology in computer engineering will explore real-time data analytics techniques.

Researchers will develop methodologies to process and analyze data at the edge of networks, enabling timely insights, efficient resource utilization, and reduced data transfer overhead.

7. Virtual and Augmented Reality Research Paradigms:

The incorporation of virtual and augmented reality (VR/AR) technologies will drive new research methodologies. Future trends will focus on immersive research approaches that explore the potential of VR/AR in education, training, healthcare, and entertainment, demanding innovative data collection and analysis techniques.

8. Quantum Computing Methodologies:

As quantum computing emerges as a disruptive technology, future research methodology will delve into the complexities of quantum algorithms, programming languages, and error correction techniques. Researchers will pioneer methodologies to harness the potential of quantum computing for solving complex computational problems.

9. Ethical Hacking and Cyber security Research:

Given the escalating challenges in cyber security, future trends in computer engineering research methodology will encompass ethical hacking methodologies. Researchers will adopt simulated attack scenarios to assess system vulnerabilities, test defense mechanisms, and develop resilient cyber security strategies.

Conclusion:

Research methodology is the compass that guides the exploration of uncharted territories in computer engineering. This chapter underscores the importance of a robust research framework, equipping researchers with the tools and insights needed to advance the boundaries of knowledge and contribute to the dynamic field of computer engineering.

References:

1. Creswell, J. W., & Creswell, J. D. (2017). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage Publications.
2. Yin, R. K. (2017). Case Study Research and Applications: Design and Methods. Sage Publications.
3. Robson, C. (2016). Real World Research. John Wiley & Sons.
4. Saunders, M., Lewis, P., & Thornhill, A. (2019). Research Methods for Business Students. Pearson.
5. Bryman, A., & Bell, E. (2015). Business Research Methods. Oxford University Press.
6. Cohen, L., Manion, L., & Morrison, K. (2018). Research Methods in Education. Routledge.

7. Easterbrook, S. M., Singer, J., Storey, M. A., & Damian, D. (2014). Selecting Empirical Methods for Software Engineering Research. In Guide to Advanced Empirical Software Engineering (pp. 285-311). Springer.
8. Kitchenham, B., & Pfleeger, S. L. (2002). Principles of Survey Research Part 1: Turning Lemons into Lemonade. ACM SIGSOFT Software Engineering Notes, 27(5), 19-20.
9. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in Software Engineering. Springer Science & Business Media.
10. Shull, F., Singer, J., Sjøberg, D. I., & Storey, M. A. (2008). Evaluating software engineering research articles: A sentiment analysis approach. Empirical Software Engineering, 13(1), 3-36.
11. Seaman, C. B., & Shull, F. (2008). "Guidelines for empirical research in software engineering". Empirical software engineering, 13(4), 361-371.
12. Runeson, P., Höst, M., Rainer, A., & Regnell, B. (2012). Case study research in software engineering: Guidelines and examples. John Wiley & Sons.

THE MATHEMATICAL EXPRESSION DENOTING THE SELLING PRICE OF A DECAYING INVENTORY, CHARACTERIZED BY A LINEAR DETERIORATION RATE, CONSTANT HOLDING COST, AND DEMAND-DEPENDENT SELLING PRICE, IS A POLYNOMIAL OF DEGREE FIVE

Ankit Sharma[1], Ram Asrey Rajput[2] and Ajay Kumar Sharma*[3]

[1]Department of Mathematics, Sanskriti University Chhatta, Mathura-281401 (UP), India

[2]Department of Mathematics, Bundelkhand college Jhansi -284001 (UP), India

[3]Department of Mathematics, Mahatma Gandhi College of Science Gadchandur,
Ta-Korpana, Dist- Chandrapur- 442908 (MS), India

Corresponding author E-mail: sharmaankit199697@gmail.com, ramasreyrajput@yahoo.in, *ajaykumarsharma101010@gmail.com

Abstract:

The paper proposes the utilization of a deterministic inventory model in contrast to a probabilistic approach. When only considering odd powers, the demand rate of the model can be represented as a fifth-degree polynomial function in terms of the selling price. It is anticipated that the cost of maintenance will remain constant. The passage of time influences the rate at which degradation occurs. Both the backlog and shortages are deemed acceptable. The inventory model employed a univariate inventory system. The absence of lead time is presumed. The solution of the differential equation was employed in this work to ascertain the optimal values. The numerical validation of the model is conducted, and convexity is visually shown as a two-dimensional graph in a maple application.

Keywords: Demand, Polynomial Function, Holding Cost, and Selling Price. Inventory Control.

Introduction:

A working business will employ some sort of inventory model to keep track of stock. Let's pretend we're interested in monitoring our stock of table salt. Because of its ubiquitous use in the kitchen, salt is a classic product that will never go out of style. In any era, salt is an essential component of the foods we eat every day. The need for salt persists. The need for salt persists. This continual need can be incorporated into a model. In a similar vein, the demand for some products fluctuates with changes in their selling prices. Consider dry fruits, which have a reasonable selling price for common people and are utilized whenever sweets are required. For people in the business class, dry fruits must be used in sweets, whereas the use of dry fruits in sweets for people in the lower and middle classes is governed by the selling price. When the market price of these commodities is high, they try to consume less of them. The inventory model for items that depreciate over time uses a five-degree polynomial with a linear degradation rate and a constant holding cost:

A polynomial of degree 5 that represents the selling price, a linear rate of degradation, and a fixed holding cost are used in this straightforward inventory model for deteriorating products. More complex models can be created by loosening some of the assumptions, such as allowing for shortages or stochastic demand.

Assumptions:

- Periodically, the inventory is managed.
- The item's demand is predictable and deterministic.
- A polynomial of degree five at the inventory level determines the item's selling price.
- The item's rate of deterioration is linear.
- The keeping expense is fixed.
- No shortages exist.

The best inventory levels can then be utilized to determine an optimal ordering policy.

Another example is:

The following assumptions can be used to develop an inventory model for degrading items with a demand-dependent selling price expressed as a polynomial of degree 5, a linear deterioration rate, and a constant holding cost:

Assumptions:

- The product's demand is determined by the selling price and is represented by a polynomial of degree five. The selling price is supposed to be a function of the quantity in demand.
- The items deteriorate at a constant linear pace, hence the amount of the items decreases linearly over time.
- The holding cost is constant and is incurred for each unit of the item retained in inventory.

Let us label the following variables:

- T: Considered time period (in days, months, or years).
- D: The item's demand quantity.
- P: The item's selling price.
- H: The item's holding cost per unit.
- r: The rate of deterioration per unit of time.

The goal is to discover the ideal order quantity (Q) and order timing to minimize total cost, which includes ordering cost, holding cost, and missed revenues due to stock outs.

The following is a definition of the total cost function: Total cost = order cost + holding cost + lost sales cost

- **Ordering Cost:** When an order is placed, the ordering cost is incurred. Let's call the ordering cost C. The ordering cost is proportional to the number of orders placed, which is calculated by dividing the total demand by the order quantity. Cost of Ordering = C * (D / Q)
- **Holding Cost:** The holding cost is incurred for each unit of inventory held. Let's call the total inventory level I. The holding cost is (Q/2) + (r * T / 2), where Q/2 is the average inventory during the replenishment cycle and r * T / 2 is the average missed sales during the replenishment cycle. H * [(Q/2) + (r * T / 2)] = Holding Cost
- **Lost Sales Cost:** The cost of lost sales is incurred when there is a stockout, which occurs when demand exceeds available inventory. If (D > Q) = 0, the cost of lost sales is P * (D - Q).

The goal is to reduce the total cost: Total cost = order cost + holding cost + lost sales cost

We can find the ideal order quantity (Q) and order timing by minimizing the total cost function.

To identify the ideal solution that reduces the total cost, this model can be solved using optimization techniques such as dynamic programming or numerical approaches. The model can be expanded by including additional elements such as lead time, backorders, and maximum inventory levels.

Literature Review:

Several academics collaborated to design a demand rate that is proportional to the selling price of the product. Some scholars, after doing a review of the relevant published material on inventory models, arrived at the following conclusions:

In 2008, Ajanta Roy created solutions for the inventory model with demand rate as a linear function of selling price, time-dependent holding cost, and time-variable deterioration rate. [1] These solutions were devised to account for the fact that these factors change over time. Timothy H. Burwell, Dinesh S. Dave, Kathy E. Fitzpatrick, and Melvin R. Roy (1997) [2] employed incorporated quantity and foreign discounts in their decision-making process. They also assumed that the demand rate was affected by price. T. Chakrabarty, B. C. Giri, and K. S. Chaudhuri (1997) [3] investigated the problem of instantaneous supply under the assumption that the demand was linearly dependent on time. Work on inventory renewal challenges while considering a limited time horizon and a demand that varies linearly with time.rate was completed in the year 1997 by T. Chakrabarti and K.S. Chaudhuri [4]. He took into account the equal phases of rejuvenation as well as the deficits that each cycle presented.

Kun-Jen Chung and Pin-Shou Ting (1993) [5] framed solutions on deterministic demand with positive linear drift. The rate of degradation per unit time was taken into consideration when formulating these solutions. The Weibull distribution function was utilized to determine the rate of deterioration in the inventory model that was developed by Richard P. Covert and George C. Philip (1973) [6]. The inventory model that Upendra Dave and L. K. Patel established in 1981 [7] included a time-dependent deterioration rate as well as a constant deterioration rate.

Assumptions:

The demand rate is selling price depend as a polynomial function degree five

$.D(s) = (bs^3 + as^5 + c), where\ s\ is\ the\ selling\ price\ per\ unit\ time. (a > 0, b > 0, c > 0)$

The order quantity per cycle is L.

Holding cost per unit time $H(t) = h, where\ h \geq 0$.

Shortages are allowed and are fully backlogged.

The cost of an item is C.

The deterioration rate $\boldsymbol{\theta}(\boldsymbol{t}) = \lambda \boldsymbol{t}, \mathbf{0} < \lambda < 1$.

The shortage cost per unit time is C_D

The lead time is zero.

In interval $(0, \vartheta)$inventory is positive and in the interval $(\vartheta, T)$the inventory level is negative.

Notations:

A-The ordering cost

D- Demand Rate

C- Purchase cost per unit

H(t)- Holding cost

L- Order Quantity

$\boldsymbol{\vartheta}$- The time when inventory level reached zero.

T-The length of a cycle time

$\boldsymbol{\theta - detetrioration\ rate}$

Model:

Period includes positive and declining inventory levels. When the inventory level gets close to zero, a shortage begins, and the inventory turns negative for the duration of the period. Differential

equation for inventory level at time t is as follows: -

$$\frac{dI(t)}{dt} + \lambda t I(t) = -(bs^3 + as^5 + c), 0 \leq t \leq \vartheta\,. \;\dots\dots(1)$$

$$\frac{dI(t)}{dt} = -(bs^3 + as^5 + c), \vartheta \leq t \leq T. \dots\dots(2)$$

On solving equation (1) and (2) and neglecting higher powers of t, we get

$$I(t) = (bs^3 + as^5 + c)(\vartheta - t) + \lambda\left(\frac{\vartheta^2}{6} + \frac{t^3}{3} - \frac{\vartheta t^2}{2}\right) + \lambda^2\left(\frac{\vartheta^5}{40} - \frac{t^5}{15} - \frac{t^2\vartheta^3}{12} + \frac{\vartheta t^4}{8}\right), 0 \leq t \leq \vartheta$$

$$I(t) = -(bs^3 + as^5 + c)(t - \vartheta), \vartheta \leq t \leq T$$

The stock loss due to deterioration is given as

$$D' = (bs^3 + as^5 + c)\int_0^{\vartheta} e^{\frac{\lambda t^2}{2}} - (bs^3 + as^5 + c)\int_0^{\vartheta} dt = (bs^3 + as^5 + c)\left(\frac{\lambda\vartheta^3}{6} + \frac{\lambda^2\vartheta^5}{40}\right)$$

$$L = (bs^3 + as^5 + c)\left(\frac{\lambda\vartheta^3}{6} + \frac{\lambda^2\vartheta^5}{40}\right) + (bs^3 + as^5 + c)T \;\dots\dots..(3)$$

Holding Cost is given as

$$H(t) = \int_0^{\vartheta} he^{\frac{-\lambda t^2}{2}}[\int_t^{\vartheta}(bs^3 + as^5 + c)e^{\frac{\lambda u^2}{2}}du]\,dt$$

$$H(t) = (bs^3 + as^5 + c)\int_0^{\vartheta} h\left(1 - \frac{\lambda t^2}{2} + \frac{\lambda^2 t^2}{8}\right).[\int_t^{\vartheta}(1 + \frac{\lambda u^2}{2} + \frac{\lambda^2 u^4}{8})du]dt$$

$$H(t) = h(bs^3 + as^5 + c)\left[\frac{\vartheta^2}{2} + \frac{\lambda\vartheta^4}{12} + \frac{\lambda^2\vartheta^6}{19} - \frac{\lambda^3\vartheta^8}{960}\right] \;\dots\dots\dots\dots(4)$$

The shorttages during the cycle is given as:

$$S = -\int_0^{\vartheta} -(bs^3 + as^5 + c)(\vartheta - t)dt$$

$$S = \frac{1}{2}(bs^3 + as^5 + c)(\vartheta - t)^2 \;\dots\dots\dots.(5)$$

Total cost per unit is as follows:

$$P = \frac{1}{T}[A + C_D L + H + C_s S]$$

$$P(T,s)=\frac{1}{T}\left[A+C_D(bs^3+as^5+c)\left(T+\frac{\lambda\vartheta^3}{6}+\frac{\lambda^2\vartheta^5}{40}\right)\right.$$
$$+h(bs^3+as^5+c)\left[\frac{\vartheta^2}{2}+\frac{\lambda\vartheta^4}{12}+\frac{\lambda^2\vartheta^6}{19}-\frac{\lambda^3\vartheta^8}{960}\right]$$
$$\left.+\frac{1}{2}C_s(bs^3+as^5+c)(\vartheta-t)^2\right]\ldots\ldots\ldots.(6)$$

For optimal value of s and T, we have

$$\frac{\partial P}{\partial s}=0\ \ldots\ldots\ldots.(7)$$

and

$$\frac{\partial P}{\partial T}=0\ \ldots\ldots\ldots.(8)$$

the minimum cost per unit of time $P(T,s)$ *as a whole satisfies the necessary criterion.*

$$\frac{\partial^2 P}{\partial T^2}>0\ \ldots\ldots\ldots.(9)$$

$$and\quad \frac{\partial^2 P}{\partial s^2}>0\ \ldots\ldots\ldots.(10)$$

$$\frac{\partial^2 P}{\partial T^2}*\frac{\partial^2 P}{\partial s^2}-\frac{\partial^2 P}{\partial s\partial T}>0\ldots\ldots\ldots\ldots.(11)$$

By resolving equations (7) and (8) and entering the values into equation (6), we can determinethe value of s and T as well as the least cost per unit of time for the values that satisfy conditions (9), (10) and (11).

Numerical Illustrations:

Parameters	Example-1	Example-2	Parameters	Example 1	Example 2
A	88888	98888	λ	1.48	1.48
B	888	900	ϑ	2	2
Γ	-8888	-999	Optimal Value of T	23.80541477	16.54472772
A (ordering-cost)	9877.99	8999.99	Optimal Value of s	0.627791521	0.3952489827
C S	108.99	18.99	Total optimalcost per unittime	414.9476065	540.6116974
C_D	935.991	95.99	-	-	-
h	6.99	6.99	-	-	-

The above illustration is graphically presented as follows:

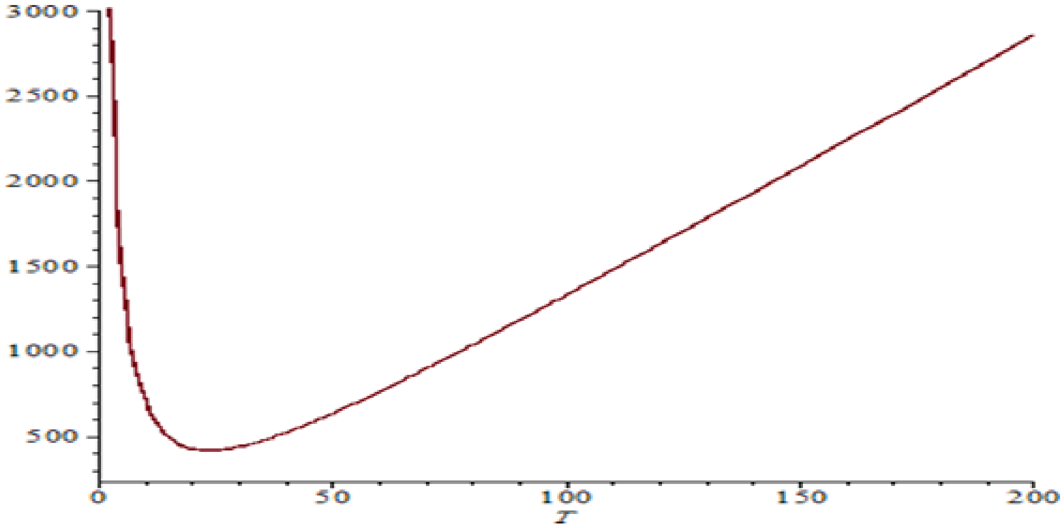


Fig. 1: Total cost function verses T

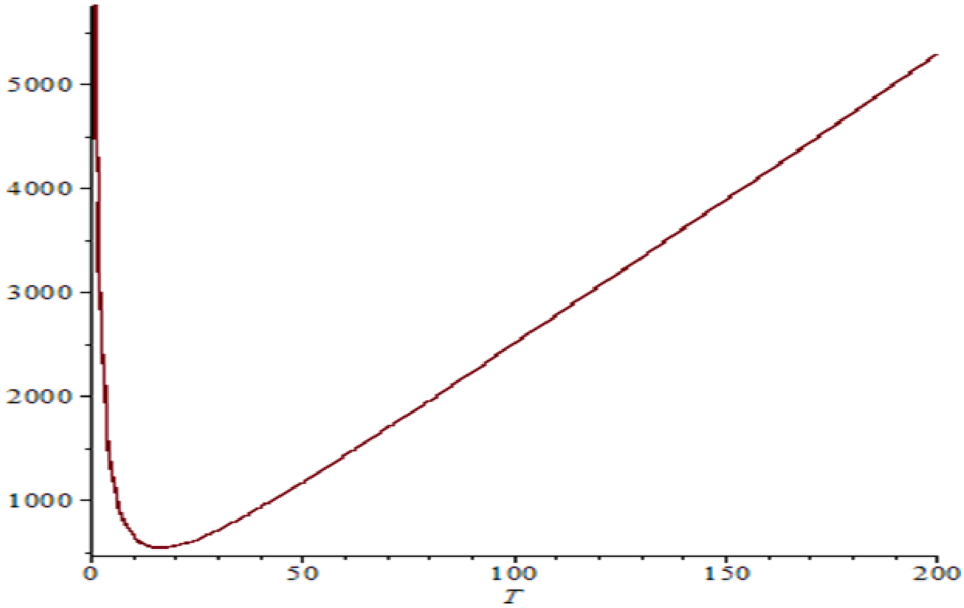


Fig. 2: Total cost function verses T

Conclusion:

In this paper's deterministic inventory model, the demand rate is determined by a five-fold increase in the selling price, a linear decline rate, and a constant holding cost. The total optimal cost has been computed for the values of and T, satisfying the condition. The model was validated using numerical and pictorial illustrations. Given that the price of items is increasing on a daily basis, this model fits the one in which the demand rate is influenced by the even powers of the selling price rather well. As a result, the concept will be extremely beneficial to businesses and society in the future. A deterministic inventory model is built in this study employing time raised to the power of p by two, a constant decline rate, and a constant holding cost. The total optimal cost has been computed for the values of and T, satisfying the condition. The model was tested using both numerical and graphical diagrams. This model is similar to the one inas commodity prices grow on a daily basis, demand swings modestly over time. The concept will greatly benefit future cities and businesses.

References:

1. Chakrabarty, T., Giri, B. C., & Chaudhuri, K. S. (1998). An EOQ model for items with Weibull distribution deterioration, shortages and trended demand: an extension of Philip's model. Computers & Operations Research, 25(7-8), 649-657.
2. Burwell, T. H., Dave, D. S., Fitzpatrick, K. E., & Roy, M. R. (1997). Economic lot size model for price-dependent demand under quantity and freight discounts. International Journal of Production Economics, 48(2), 141-155.
3. Chakrabarti, T., & Chaudhuri, K. S. (1997). An EOQ model for deteriorating items with a linear trend in demand and shortages in all cycles. International Journal of Production Economics, 49(3), 205-213.
4. Chung, K. J., & Ting, P. S. (1993). A heuristic for replenishment of deteriorating items with a linear trend in demand. Journal of the operational research society, 44(12), 1235-1241.
5. Fujiwara, O., &Perera, U. L. J. S. R. (1993). EOQ models for continuously deteriorating products using linear and exponential penalty costs. European Journal of Operational Research, 70(1), 104-114.
6. Goh, M. (1994). EOQ models with general demand and holding cost functions. European Journalof Operational Research, 73(1), 50-54.
7. Giri, B. C., & Chaudhuri, K. S. (1997). Heuristic models for deteriorating items with shortages and time-varying demand and costs. International journal of systems science, 28(2), 153-159.
8. Hariga, M. A., &Benkherouf, L. (1994). Optimal and heuristic inventory replenishment models for deteriorating items with exponential time-varying demand. European Journal of Operational Research, 79(1), 123-137.
9. Jalan, A. K., & Chaudhuri, K. S. (1999). Structural properties of an inventory system with deterioration and trended demand. International Journal of Systems Science, 30(6), 627-633.

ON THE INEQUALITY $\frac{\psi(\varphi(n))}{n} \geq \frac{1}{2}$

Bhabesh Das

Department of Mathematics,

Arya Vidyapeeth College (Autonomous), Assam

Corresponding author E-mail: mtbdas99@gmail.com

Abstract:

In 1988 J. Sandor [3] conjectured that for any positive integer n greater than 1,

$$\frac{\psi(\varphi(n))}{n} \geq \frac{1}{2}, \tag{*}$$

where φ is the Euler's totient function and ψ is the Dedekind arithmetic function. In this paper, we present a new class of number for which this conjecture is valid.

Keywords: Arithmetic functions, Euler's totient function , Dedekind arithmetic function, Sandor conjecture

AMS Classification: 11A25, 11N37

1. Introduction: An arithmetic function or number theoretic function f is one whose domain is the set of positive integers and whose range is a subset of the complex numbers. Arithmetic function f is said to be multiplicative, if $f(mn) = f(m)f(n)$ for g.c.d$(m,n)=1$. Well known Euler's totient function φ and Dedekind arithmetic function ψ are defined as

$$\varphi(n) = n\prod_{p|n}(1-\frac{1}{p})$$

$$\text{and } \psi(n) = n\prod_{p|n}(1+\frac{1}{p})$$

respectively for any positive integer n, where p runs through the distinct prime divisors of n. These two functions are multiplicative function.

Let P^* be the set of all odd primes $P^{\#} = \{p \in P^* : p = 2^{\alpha}m+1, \alpha > 1, m \geq 1, (m,2)=1\}$. We prove that if $n \in W(P^{\#})$, then J.Sandor conjecture (*) is true, where $W(P^{\#}) = \{n = \prod_{j=1}^{r} p_j : p_j = 2^{\alpha_j}m_j + 1, \alpha_j > 1, m_j \geq 1\}$, where $(m_i, m_j) = 1$;for $i \neq j$; i, j=1,2,3....r .

We use the following results extensible to prove our result.

1.1Lemma: (Sandor's inequalities (see [3])) For all positive integers $a,b \ge 1$, we have

$$\psi(ab) \ge a\psi(b)$$

and for all positive integers a,b such that there is at least one prime q such that $q \mid a$ and q does not divide b, then

$$\psi(ab) \ge (a+1)\psi(b)\,.$$

2. Main Result:

Theorem: For any $n \in W(P^{\#})$, we have

$$\frac{\psi(\varphi(n))}{n} \ge \frac{1}{2} \tag{2.1}$$

moreover if $k = 2n$, $n \in W(P^{\#})$, then (2.1) is also true.

Proof: Let $n \in W(P^{\#})$, then $n = p_1 p_2 .... p_k .$,where p_j are odd primes of the form

$$p_j = 2^{\alpha_i} m_i + 1; (2, m_i) = 1; m_i > 1,\ \alpha_i \ge 1; i = 1, 2, ..., k \tag{2.2}$$

By the well-known property of the Euler's arithmetic function φ and (2.2) it follows that

$$\varphi(n) = \varphi(p_1 p_2 .... p_k) = (p_1 - 1)(p_2 - 1)....(p_k - 1) = 2^{\alpha_1 + \alpha_2 + ... + \alpha_k} m_1 m_2 ......m_k \tag{2.3}$$

From (2.3) and the multiplicative property of the function ψ we obtain

$$\psi(\varphi(n)) = 3.2^{\alpha_1 + \alpha_2 + ... + \alpha_k - 1} \varphi(m_1 m_2 ......m_k) \tag{2.4}$$

We prove the inequality (2.1) by induction with respect to $k = \omega(n)$, where $\omega(n)$ denotes the number of distinct primes in the number $n \in W(P^{\#})$.

For $k = 1$, $n = p_1 = 2^{\alpha_1} m_1 + 1$,then $\varphi(n) = 2^{\alpha_1} m_1$ and since $\psi(m_1) \ge m_1$, so

$\psi(\varphi(n)) = 3.2^{\alpha_1 - 1} \psi(m_1) > 2^{\alpha_1 - 1} m_1 = \dfrac{n-1}{2}$, implying that $\dfrac{\psi(\varphi(n))}{n} \ge \dfrac{1}{2}$.

Suppose that (2.1) holds for all $n \in W(P^{\#})$ with $k \le r$, then for any $n = p_1 p_2 .... p_r$

$$\frac{\psi(\varphi(p_1 p_2 .... p_r))}{p_1 p_2 .... p_r} = \frac{3.2^{\alpha_1 + \alpha_2 + ... + \alpha_r - 1}}{(2^{\alpha_1} m_1 + 1)(2^{\alpha_2} m_2 + 1).......(2^{\alpha_r} m_r + 1)} \psi(m_1 m_2 ......m_r) \ge \frac{1}{2} \tag{2.5}$$

Now, let $m \in W(P^{\#})$ and $k = \omega(m) = r + 1$, so

$$m = p_1 p_2 .... p_r p_{r+1} = (2^{\alpha_1} m_1 + 1)(2^{\alpha_2} m_2 + 1).......(2^{\alpha_r} m_r + 1)(2^{\alpha_{r+1}} m_{r+1} + 1) \tag{2.6}$$

From (2.6) and the property of arithmetic function φ we get

$$\varphi(m) = \varphi(p_1 p_2 .... p_{r+1}) = (p_1 - 1)(p_2 - 1)....(p_{r+1} - 1) = 2^{\alpha_1 + \alpha_2 + ... + \alpha_{r+1}} m_1 m_2 ......m_{r+1} \tag{2.7}$$

By (2.7) and the fact that $(2, m_i) = 1$, $i = 1, 2......., r+1$ and multiplicative property of the function ψ it follows that $\psi(\varphi(n)) = \psi(2^{\alpha_1+\alpha_2+.......+\alpha_{r+1}})\psi(m_1 m_2......m_{r+1}) = 3.2^{\alpha_1+\alpha_2+.......+\alpha_{r+1}-1}\psi(m_1 m_2......m_{r+1})$

(2.8)

By the definition of the set $W(P^{\#})$, it follows that $(m_i, m_j) = 1$,for all distinct $i, j = 1, 2, ..., r+1$ and hence we have

$$(m_1 m_2......m_r, m_{r+1}) = 1 \tag{2.9}$$

From (2.8),(2.9),lemma 1.1 and the multiplicative property of the function ψ we obtain

$$\psi(\varphi(m)) \geq 2^{\alpha_1+\alpha_2+.......+\alpha_{r+1}-1}\psi(m_1 m_2......m_r)(m_{r+1}+1) \tag{3.0}$$

Then by using (3.0)

$$\frac{\psi(\varphi(m))}{m} \geq \frac{2^{\alpha_1+\alpha_2+.......+\alpha_{r+1}-1}\psi(m_1 m_2......m_r)}{(2^{\alpha_1} m_1+1)(2^{\alpha_2} m_2+1).......(2^{\alpha_{r+1}} m_{r+1}+1)}(m_{r+1}+1) \tag{3.1}$$

From (3.1) and (2.5), we obtain

$$\frac{\psi(\varphi(m))}{m} \geq \frac{1}{2}\left(\frac{2^{\alpha_{r+1}}(m_{r+1}+1)}{2^{\alpha_{r+1}} m_{r+1}+1}\right) \geq \frac{1}{2}$$

and the proof of the theorem is completed.

References:

1. Cohen, G. L. (1997). On a conjecture of Makowski and Schinzel. *Colloq. Math., 74,* 18.
2. Guy, R. K. (1994). *Unsolved Problems in Number Theory.* Springer-Verlag.
3. Sandor, J. (2005). On the composition of some arithmetic functions II. *jipam, 6*(3), Article 73.
4. Sandor, J. (1989). On the composition of some arithmetic functions. *Studia Babes-Bolyai University, Mathematica, 34,* 7–14.
5. Sandor, J. (2004). *Handbook of Number Theory - II.* Springer Verlag.

Research Trends in Computer Science, Mathematical Science and Information Technology

(ISBN: 978-93-88901-91-8)

About Editors

Mrs. Rimpi Rani is currently working as Assistant Professor in Punjabi University, TPD Malwa College, Rampura Phul, Dist. Bhatinda (Punjab). She has 20 years of teaching experience. She is pursuing her Ph.D. She is continuously contributing in research. She has attended and presented her research work in various national and international level conferences, seminars, workshops, etc. She also served as resource person in various events.

Mr. Somnath Shivram Sanap is working as Assistant Professor at Department of Mathematics, A.V College of Arts K.M College of Commerce and E.S A College of Science Vasai Road, Palghar, (Maharashtra) India. He has completed M.sc Mathematics from K.T.H.M. College Nasik. He also qualified SET exam in Mathematics. He has an experience of more than 11 years of teaching to under graduate level. He taught various subjects of Engineering and B.Sc at UG & PG level. He has published 02 book chapters in different books and presented two research papers at national and international conference.

Mr. Apurav Radhakrishna Gupta is working as Assistant Professor at Department of Mathematics, Late Pundalikrao Gawali Arts and Science Mahavidyalaya, Shirpur (Jain), M.S. India. He has completed M. Phil. in Mathematics from D.B.R.A University Agra, Agra in 2005. Mr. Gupta also completed M.C.A. from IGNOU Delhi in 2006. He qualified CSIR NET in December, 2012. He has 15 years of teaching experience at UG level. He is serving his duties on various college level committees as chairman and member.

Subharun Pal, an exemplar of scholastic perspicuity and methodological rigor, is simultaneously immersed in an advanced program of academic inquiry within the prestigious corridors of the Swiss School of Management (SSM) in Switzerland and the eminent European International University (EIU) in France. His intellectual fiefdom traverses an interdisciplinary confluence covering computer science engineering, innovative technological shifts, strategic operations management, integrative logistics, in-depth supply chain dynamics, applied finance, intricate commercial legal structures, and the foundational principles of education. Further buttressing his academic portfolio are nuanced engagements with premier pedagogical establishments, such as IIT Jammu, IIT Patna, IIM Calcutta, IIM Ranchi, Edith Cowan University Perth, CII-Institute of Logistics Chennai, National University of Juridical Sciences Kolkata, Karnataka State Open University Mysore, and Visvesvaraya Technological University Belgaum. His scholarly cachet is redoubled by accolades and endorsements from global organizational entities including The World Bank, KPMG, Cisco, Microsoft, Oracle, EC Council, Exemplar Global Inc., ISEL Global Canada, APMG UK, ISI Bangalore, NIIT, ILI New Delhi, SHRI Singapore, and TüV Süd Akademie.

Bhumi Publishing

ISBN 978-93-88901-91-8

9 789388 901918